Rewriting modulo symmetric monoidal structure

Filippo Bonchi

CNRS, ENS Lyon

filippo.bonchi@gmail.com

Fabio Gadducci

U. Pisa

gadducci@di.unipi.it

Aleks Kissinger

Radboud U. Nijmegen

aleks@cs.ru.nl

Paweł Sobociński

U. Southampton

ps@ecs.soton.ac.uk

Fabio Zanasi

Radboud U. Nijmegen

fzanasi@cs.ru.nl

Abstract

String diagrams are a powerful and intuitive graphical syntax for terms of symmetric monoidal categories (SMCs). They find many applications in computer science and are becoming increasingly relevant in other fields such as physics and control theory.

An important role in many such approaches is played by equational theories of diagrams, typically oriented and applied as rewrite rules. This paper lays a comprehensive foundation for this form of rewriting. We interpret diagrams combinatorially as typed hypergraphs and establish the precise correspondence between diagram rewriting modulo the laws of SMCs on the one hand and double pushout (DPO) rewriting of hypergraphs, subject to a soundness condition called convexity, on the other. This result rests on a more general characterisation theorem in which we show that typed hypergraph DPO rewriting amounts to diagram rewriting modulo the laws of SMCs with a chosen special Frobenius structure.

We illustrate our approach with a proof of termination for the theory of non-commutative bimonoids.

1. Introduction

Symmetric monoidal categories (SMCs) are categories where arrows can be composed sequentially (;) and in parallel ($\oplus$). The interplay between these two kinds of composition is commonplace, and indeed SMCs have found many applications in computer science, physics and related fields. Focussing on computer science, they feature in concurrency theory, where they describe the concurrent nature of executions of Petri nets [30] as well as serving as their compositional algebra [8, 39], quantum information, where they model quantum circuits [10, 11], and in systems theory, where they provide a calculus of signal flow graphs [2, 4, 6].

In each case, the algebra of SMCs gives us a syntax to talk about domain-specific artefacts. However, the two composition operations in an SMC are related by functoriality, and symmetries are natural: this imposes a non-trivial structural equality relation on terms from the outset—something that in process algebra is referred to as *structural congruence*—that makes using ordinary tree-like syntax ineffectual. Functoriality means that, given terms A, B, C, D where A,B and C,D can be composed sequentially:

$$(A \oplus C) \, ; \, (B \oplus D) = (A \, ; \, B) \oplus (C \, ; \, D). \qquad (1)$$

As a consequence, this syntax is intrinsically 2-dimensional, and so diagrams—in this context often referred to as *string diagrams*—are a more efficient representation for arrows of SMCs. Indeed, both

sides of the equation above are represented diagrammatically as
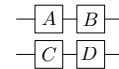
and so (1) is built into the representation, along with equational properties such as associativity of both composition operations.

The history of string diagrams begins with Feynman and Penrose, but they remained just a tool for private calculations, ultimately excluded from papers. This was likely due to a lack of foundational results that justified their use: the careful mathematician checked each step in a diagrammatic proof using standard term-based means. This changed with the 1991 paper [20] of Joyal and Street who formalised diagrams as topological structures and understood diagrammatic manipulation as homotopy. Their framework allowed them to show that the resulting diagrams-up-to-homotopy-equivalence served as a description for the arrows of free braided monoidal categories, of which SMCs are a special case. Subsequently, the use of diagrammatic notation exploded, see e.g. the survey [37]. The results of Joyal and Street mean that we have a formal description of the nature of 2-dimensional syntax, and so of the arrows of free braided monoidal categories.

Most applications, however, do not feature free categories but rather rely on the presence of additional equations: for example, algebraic structures such as bimonoids and Frobenius monoids are commonplace. Adding equations to a theory of string diagrams means that diagrammatic proofs include *rewriting*: if the left hand side of an equation can be found in a larger string diagram, it can be deleted and replaced with its right hand side.[1] From a mathematical point of view, one can formulate rewrite rules as generator 2-cells (this data structure is variously called a *computad* [40] or a *polygraph* [9]) and consider the resulting free 2-category, where the 2-cells witness the possible rewriting trajectories. This does not solve the problem of how to *implement* rewriting, and the approach of Joyal and Street does not offer an immediate solution either; we do not have an off-the-shelf rewriting theory for their diagrams.

One of the fundamental difficulties with working with terms modulo the laws of SMCs is finding matches. For example, consider the following rewrite rule
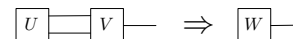
then, using naturality, we ought to be able to find a match in
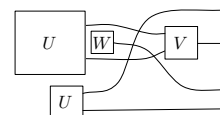
which, viewed as a term, does not contain the l.h.s. as a subterm.

---

[1] Diagram rewriting may represent *e.g.* a system whose topology dynamically changes during execution.

2016/4/29

Our approach is to think of string diagrams neither as terms nor as topological entities, but rather as combinatorial structures: i.e. as certain "open" (hyper-)graphs. Roughly speaking—since we focus on the symmetric case—the geometry of diagrams is discarded, and only connectivity information remains. We show that rewriting of diagrams is an instance of graph rewriting, in particular the double pushout (DPO) approach, a research topic that goes back to the 1970s [15]. The correspondence between rewriting modulo the laws of SMC and DPO is surprisingly tight, as we explain below.

To establish the connection between rewriting of string diagrams and graph rewriting, we extend and connect together two existing research threads. First, the algebraic nature of cospans of graphical structures was studied in e.g. [34], where the algebra of special Frobenius monoids was shown to play a crucial role. In a cospan $n \to G \leftarrow m$, the open nature of graphs is exhibited by the "interfaces" n and m which, like "dangling wires" in string diagrams, allow composition on the left and right.

Second, there has been work connecting computads in cospans [17, 35] and DPO graph rewriting: the key observation is that DPO rules, which are usually presented as spans of graphs $L \leftarrow j \to R$, correspond to rewrite rules in the cospan category as displayed on the right, where 0 is the initial object in the category of graphs.

$$
\begin{array}{ccc}
 & j & \\
\swarrow & & \searrow \\
L & \Rightarrow & R \\
\nwarrow & & \nearrow \\
 & 0 &
\end{array}
$$
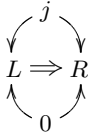
Our starting point is the extension of the result in [34] from graphs to hypergraphs. This is essential to tackle arbitrary symmetric monoidal signatures Σ: the category of cospans of hypergraphs with type Σ is isomorphic to the free *hypergraph category*, that is the SMC freely generated by Σ together with a chosen special Frobenius structure. Hypergraph categories have previously been called well-supported compact closed categories by Walters [41].

By connecting this result with the observations in [17, 35], we are able to show that DPO is just rewriting modulo the laws of SMCs "plus" Frobenius. More precisely, a DPO system on hypergraphs of type Σ amounts to a rewriting system on the free SMC generated by Σ together with a chosen special Frobenius structure. Intuitively speaking, the Frobenius equations imply a self-dual compact closed structure, which in turn allows us to bend rules around: an arbitrary rewriting rule of cospans, as below on the left, can always be transformed into the DPO rule on the right. The presence of a Frobenius structure means that this rule transformation is also sound in the corresponding SMC.

$$
\begin{array}{ccccccc}
 & j & & & & i+j & \\
\swarrow & & \searrow & & \swarrow & & \searrow \\
L & \Rightarrow & R & & L & \Longrightarrow & R \\
\nwarrow & & \nearrow & & \nwarrow & & \nearrow \\
 & i & & & & 0 &
\end{array}
$$

Next we tackle rewriting modulo the equations of SMCs in isolation. We can still consider DPO, but we must be more careful about rule application: since we may not have a compact closed structure on the algebraic side, it may be the case that a graph rewriting rule is unsound when considered as a rewrite of string diagrams. Here we introduce a restricted form of DPO rewriting, called convex DPO rewriting, and prove that it is sound and complete with respect to rewriting modulo symmetric monoidal structure.

Rewriting modulo SMC

Rewriting modulo SMC + chosen special Frobenius structure

DPO rewriting of hypergraphs

convex DPO rewriting of hypergraphs
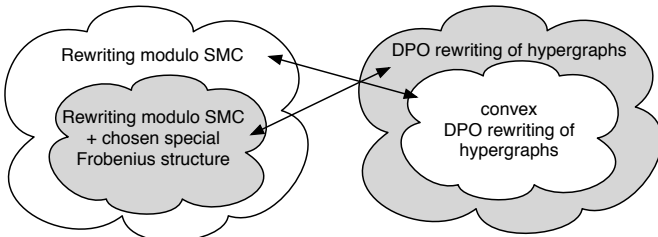
We conclude the paper with an illustrative example of our approach by proving termination for the theory of *non-commutative bimonoids* (a.k.a. bialgebras). As any non-commutative group yields a bimonoid whose multiplication is non-commutative, these play an important role in representation theory, and the case where neither the multiplication nor the comultiplication is commutative forms the basis of the study of quantum groups [22].

Related work. The correspondence between terms of algebraic structures with sequential and parallel composition (ultimately representing arrows of a free SMC) and flow diagrams (i.e., suitable hyper-graphs) was recognized early on in computer science, and studied at least since the work of Stefanescu (see the references in the survey [37]). After [20], and especially after the paper on traced monoidal categories by Joyal, Street and Verity [21], there was a flourishing of interest in string diagrams.

A long tradition, pioneered by the work of Burroni on polygraphs [9], generalised term rewriting to higher dimensions, including the three-dimensional case of string diagram rewriting — see e.g. [31] for a survey. Here, the laws of SMCs are usually considered as explicit rewriting rules, resulting in rather elaborate rewriting systems whose analysis is often challenging (see e.g. [18, 27]). Abstract higher-dimensional rewriting is far more general than our approach, which focusses only on SMCs. However, the benefit of our graphical representation is that it has the laws of SMCs built-in, reducing considerably the number of rewriting rules and simplifying the analysis of the resulting rewriting systems.

In [13], the authors use cospans of *string graphs* (a.k.a. open graphs) to encode morphisms in a symmetric monoidal category and reason equationally via DPO rewriting. There is an evident encoding of the hypergraphs we use to string graphs, which yields an equivalence between the category of directed cospans from [13] and our subcategory of monogamous acyclic cospans in $\mathbf{FTerm}_\Sigma$ (characterised in Theorem 3.12). However, the notion of rewriting considered in loc. cit. is only sound if there is a trace on the SMC, whereas our notion of convex DPO rewriting guarantees soundness for any SMC. Another difference is that we directly work in an *adhesive category* [26], while the category of open graphs needs to inherit the good rewriting properties from an embedding into the adhesive category of typed graphs.

Structure of the paper. Section 2 provides background on SMCs and string diagrams. Section 3 introduces the combinatorial structures for interpreting diagrams and characterises them algebraically in terms of Frobenius monoids. Section 4 establishes the correspondence between DPO rewriting and rewriting in a free SMC with a chosen special Frobenius monoid. Section 5 develops a restriction of DPO rewriting, called *convex* DPO, that is adequate for rewriting in any free SMC. Section 6 is devoted to the proof of termination for the theory of non-commutative bimonoids. Omitted proofs are available in an extended version [?] of this paper.

2. Background

Notation. Composition of $f\colon a \to b$, $g\colon b \to c$ in a category $\mathbb{C}$ is written $f\,;\,g\colon a \to c$. For $\mathbb{C}$ symmetric monoidal (SMC), $\oplus$ is its monoidal product and $\sigma_{a,b}\colon a \oplus b \to b \oplus a$ is the symmetry for $a, b \in \mathbb{C}$. Given $\mathbb{C}$ with pushouts, its cospan bicategory has the objects of $\mathbb{C}$ as 0-cells, cospans of arrows of $\mathbb{C}$ as 1-cells and cospan morphisms as 2-cells; $\mathsf{Cospan}(\mathbb{C})$ is the *category* obtained by identifying the isomorphic 1-cells and forgetting the 2-cells.

SMTs and PROPs. A standard way of expressing the algebraic structure borne in SMCs is through the notion of *symmetric monoidal theory* (SMT). A one-sorted SMT is determined by (Σ, E) where Σ is the *signature*: a set of *generators* $o\colon n \to m$

$$(c\,;c')\,;d = c\,;(c'\,;d) \qquad id_n\,;c = c = c\,;id_m$$
$$(c \oplus c') \oplus d = c \oplus (c' \oplus d) \qquad id_0 \oplus c = c = c \oplus id_0$$

$$(c\,;c') \oplus (d\,;d') = (c \oplus d)\,;(c' \oplus d')$$
$$\sigma_{1,1}\,;\sigma_{1,1} = id_2$$
$$(c \oplus id_z)\,;\sigma_{m,z} = \sigma_{n,z}\,;(id_z \oplus c)$$

Figure 1. Laws of SMCs instantiated to a PROP **X**.

Figure 2. The equations E_F of special Frobenius monoids.

with *arity* n and *coarity* m where $m, n \in \mathbb{N}$. The set of Σ-terms is obtained by combining generators in Σ, the unit $id\colon 1 \to 1$ and the symmetry $\sigma_{1,1}\colon 2 \to 2$ with $;$ and $\oplus$. This is a purely formal process: given Σ-terms $t\colon k \to l$, $u\colon l \to m$, $v\colon m \to n$, we construct new Σ-terms $t\,;u\colon k \to m$ and $t \oplus v\colon k + m \to l + n$. The set E of *equations* contains pairs of Σ-terms of the form $(t, t'\colon i \to j)$; the only requirement is that t and t' have equal arities and coarities.

Just as ordinary (cartesian) algebraic theories have a categorical rendition as Lawvere categories [19], the corresponding (linear[2]) notion for SMTs is the one of PROP [28] (**pro**duct and **p**ermutation category). A PROP is a symmetric strict monoidal category with objects the natural numbers, where $\oplus$ on objects is addition. Morphisms between PROPs are identity-on-objects strict symmetric monoidal functors. PROPs and their morphisms form a category **PROP**. Any SMT (Σ, E) freely generates a PROP by letting the arrows $n \to m$ be the Σ-terms $n \to m$ modulo the laws of symmetric monoidal categories (in Figure 1) and the (smallest congruence containing the) equations $t = t'$ for any $(t, t') \in E$.

We write $\mathbf{S}_\Sigma$ to denote the PROP freely generated by $(\Sigma, \varnothing)$. There is a graphical representation of the arrows of $\mathbf{S}_\Sigma$ as string diagrams, which we now sketch, referring to [37] for the details. A Σ-term $n \to m$ is pictured as a box with n ports on the left and m ports on the right, which are ordered and referred to with top-down enumerations $1, \dots, n$ and $1, \dots, m$. Compositions via $;$ and $\oplus$ are drawn respectively as horizontal and vertical juxtaposition.
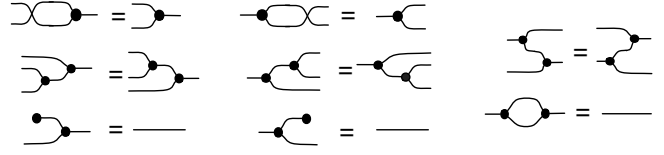
$$t\,;s \text{ is drawn } \boxed{t}\boxed{s} \qquad t \oplus s \text{ is drawn } \frac{\boxed{t}}{\boxed{s}} \tag{2}$$

There are specific diagrams for the Σ-terms responsible for the symmetries: these are $id_1\colon 1 \to 1$, represented as ⊣⊢, the symmetry $\sigma_{1,1}\colon 1 + 1 \to 1 + 1$, represented as ⋈, and the unit object for $\oplus$, that is, $id_0\colon 0 \to 0$, whose representation is an empty diagram ☐. Graphical representation for arbitrary identities id_n and symmetries $\sigma_{n,m}$ are generated according to the pasting rules in (2). Note that, of the equations displayed in Figure 1, the first five are implicit in the diagrammatic language.

It will be sometimes convenient to represent id_n with the shorthand diagram $\boxed{n}$ and, similarly, $t\colon n \to m$ with $\overset{n}{-}\boxed{t}\overset{m}{-}$.

Example 2.1.
(a) A basic example is the theory (Σ_M, E_M) of *commutative monoids*. The signature Σ_M contains two generators: *multiplication* — which we depict as the string diagram $\multimap\colon 2 \to 1$ — and *unit*, represented as $\bullet\!-\colon 0 \to 1$. Equations in E_M are given in the leftmost column of Figure 2: they are built from id_0, $\sigma_{1,1}$ and the generators, composed with $;$ and $\oplus$, and assert commutativity, associativity and unitality. We call **M** the PROP freely generated by the SMT (Σ_M, E_M). A useful observation is that we can give a concrete description of **M**: it is isomorphic to the PROP **F** with arrows $n \to m$ the functions $\{0, \dots, n-1\} \to \{0, \dots, m-1\}$, see e.g. [25] for details.

(b) An SMT that plays a key role in our exposition is the theory (Σ_F, E_F) of *special Frobenius monoids*. The signature Σ_F is:

$$\{\multimap\colon 2 \to 1, \ \bullet\!-\colon 0 \to 1, \ \triangleleft\colon 1 \to 2, \ -\!\bullet\colon 1 \to 0\}.$$

Equations E_F are in Figure 2. They include the theory of commutative monoids in the leftmost column. Dually, the equations in the middle column assert that $\triangleleft$ and $-\!\bullet$ form a cocommutative comonoid. Finally, the two rightmost equations describe the interaction between these two structures.

The PROP freely generated by (Σ_F, E_F) is called **Frob**. Just as **M**, **Frob** enjoys a concrete description: it is isomorphic to the PROP Cospan(**F**) of cospans in **F** [25]. The isomorphism $\xi\colon \mathbf{Frob} \to \mathsf{Cospan}(\mathbf{F})$ is the unique PROP morphism defined as follows on the generators in Σ_F.

$$\multimap \mapsto (2 \to 1 \leftarrow 1) \qquad \bullet\!- \mapsto (0 \to 1 \leftarrow 1)$$
$$\triangleleft \mapsto (1 \to 1 \leftarrow 2) \qquad -\!\bullet \mapsto (1 \to 1 \leftarrow 0)$$

(c) The theory of *non-commutative bimonoids* has signature

$$\{\multimap, \circ\!-, \triangleleft, -\!\circ\}$$

and the following equations:

$$\tag{3}$$

We call **NB** the resulting freely generated PROP. In Section 6 we show that the rewriting system obtained by orienting the equalities from left to right terminates. For this, it will be convenient to use μ, η, ν, ϵ respectively to refer to the generators.

Compact-closedness. SMTs containing a Frobenius monoid will play a special role in our work. We say that an SMT (Σ, E) is equipped with a *chosen special Frobenius structure* if $\Sigma = \Sigma' + \Sigma_F$ and $E = E' + E_F$ for some signature Σ' and set of equations E'. It is relevant for us that PROPs freely generated by such SMTs possess a self-dual compact closed structure [23]. This amounts to the existence, for each n, of "cups" $\boxed{\supset^n}\colon n + n \to 0$ and "caps" $\boxed{{}^n\subset}\colon 0 \to n + n$, subject to equations

$$\overset{n}{\underset{n}{\supset}}{}^n = \overset{n}{-} = {}^n\overset{n}{\underset{n}{\subset}}{}^n. \tag{4}$$

The equations of a Frobenius monoid imply (4) with cups defined according to the following pattern:

and similarly for caps, using $\boxed{\!\cdot\!}$ and $\boxed{\!\cdot\!}$. This data also yields a contravariant PROP morphism $\cdot^\star$ (*cf.* [36, Rmk 2.1]) defined as

$$\overline{n\boxed{c}m} \quad\mapsto\quad \overline{m\boxed{c^\star}n} \;:=\; \frac{m}{n\,\boxed{c}}\Big)^m .$$

Rewriting in a PROP. We now fix an arbitrary PROP $\mathbf{X}$. A rewriting *rule* is a pair $\langle l, r\rangle$ where $l, r\colon i \to j$ are arrows of $\mathbf{X}$ with the same source and target. We say that $i \to j$ is the rule's *type* and sometimes write $\langle l, r\rangle\colon (i, j)$. A *rewriting system* $\mathcal{R}$ is a set of rules. A rewriting step is defined for any two arrows $d, e\colon n \to m$ in $\mathbf{X}$ as $d \Rightarrow_{\mathcal{R}} e$ iff $\exists \langle l, r\rangle\colon (i, j) \in \mathcal{R}, c_1 : n \to k + i, c_2 : k + j \to n$ such that $d = c_1 \,;\, (id_k \oplus l) \,;\, c_2$ and $e = c_1 \,;\, (id_k \oplus r) \,;\, c_2$, i.e., diagrammatically:

$$\overline{n\boxed{d}m} \;=\; \overline{n\boxed{c_1}}\overset{k}{\underset{i\,\boxed{l}\,j}{}}\boxed{c_2}m$$

$$\overline{n\boxed{e}m} \;=\; \overline{n\boxed{c_1}}\overset{k}{\underset{i\,\boxed{r}\,j}{}}\boxed{c_2}m .$$

When the PROP under consideration $\mathbf{X}$ is $\mathbf{S}_\Sigma$ for some signature Σ, the notion of rewriting step can be reformulated as follows: in order to apply a rewrite rule $\langle l, r\rangle\colon (i, j)$ to a diagram d in $\mathbf{S}_\Sigma$ we need to find a match of l in d. This means finding a *context* C: a term in $\mathbf{S}_{\Sigma + \{\bullet : \, i \to j\}}$ with exactly one occurrence of $\bullet$, such that $d = C[l/\bullet]$. The rewrite then takes $d \Rightarrow_{\mathcal{R}} e$, where $e = C[r/\bullet]$.

Remark 2.2. Another common way of viewing a rewrite system on (the arrows) a monoidal category is to view a collection of rules $\langle l, r\rangle : (i, j)$ as a computad [40] or polygraph [9]. One can then consider the free 2-category on this data, in a similar way to how one obtains a free category on a directed graph.

With these definitions, diagrammatic reasoning can now be seen as a special case of rewriting. Given an arbitrary SMT (Σ, E) we can obtain a rewriting system $\mathcal{R}_E$ as

$$\mathcal{R}_E = \{\, \langle t, t'\rangle \mid (t, t') \in E \,\} \cup \{\, \langle t', t\rangle \mid (t, t') \in E \,\}.$$

Proposition 2.3. *Let c, d be two diagrams in $\mathbf{S}_\Sigma$. Then $c = d$ in the PROP freely generated by (Σ, E) iff $c \Rightarrow^*_{\mathcal{R}_E} d$.*

3. Frobenius termgraphs

In order to implement rewriting of string diagrams, we shift our focus away from viewing them as essentially a shorthand for terms, in favour of considering them as combinatorial hypergraph-like structures. In the following let $\mathbf{F}$ be the PROP of functions, introduced in Example 2.1.(a). We start by defining the category of hypergraphs.

Definition 3.1 (Hypergraphs)**.** The category of finite directed hypergraphs $\mathbf{Hyp}$ is the functor category $\mathbf{F^I}$ where $\mathbf{I}$ has as objects pairs of natural numbers $(k, l) \in \mathbb{N} \times \mathbb{N}$ together with one extra object $\star$. For each $k, l \in \mathbb{N}$, there are $k + l$ arrows from (k, l) to $\star$.

An object f of $\mathbf{F^I}$ is the hypergraph with set of *nodes* $f(\star)$ and, for each k, l, $f(k, l)$ the set of *hyperedges* with k (ordered) sources and l (ordered) targets. We shall visualise hypergraphs as follows: $\bullet$ is a node and $\boxed{::\!\boxed{}\!::}$ is an hyperedge, with ordered tentacles attached to the left boundary linking to sources and the ones on the right linking to targets. Here is an example.
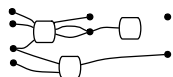
In order to serve as an interpretation for string diagrams, we need to extend such vanilla hypergraphs in two ways, leading to the notion of *Frobenius termgraph* (Def. 3.2). The first is to assign hyperedges to the generators in a given signature. Formally, this is achieved as follows. Any symmetric monoidal signature Σ can be considered as a hypergraph with a single node, in the obvious way. We can then express Σ-typed hypergraphs as the objects of the slice category $\mathbf{Hyp}_\Sigma \overset{\text{def}}{=} \mathbf{Hyp}/\Sigma$. The typing is represented pictorially by labeling hyperedges with generators in Σ.
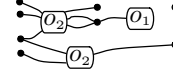
Henceforward we shall only work in the slice, thus referring to its objects directly as (Σ-)hypergraphs.

The second extension is to identify the "left and right dangling wires". A natural solution is to use *discrete cospans*, that is, the full subcategory of $\mathsf{Cospan}(\mathbf{Hyp}_\Sigma)$ (the *category* of cospans)[3] with objects discrete hypergraphs, i.e. those hypergraphs with no edges.

Such discrete cospans can encode terms, termgraphs [12, 38] (where variable sharing can occur), but also cyclic structures where, for example, an output of an operation can be fed back as an input. As we shall see (Theorem 3.3), there is a deep connection with the theory of special Frobenius monoids; for this reason, we shall refer to these discrete cospans as *Frobenius termgraphs*.

Definition 3.2 (Frobenius termgraphs)**.** The PROP $\mathbf{FTerm}_\Sigma$ of Σ-*Frobenius termgraphs* is the full subcategory of $\mathsf{Cospan}(\mathbf{Hyp}_\Sigma)$ with objects the discrete hypergraphs. Discrete hypergraphs are in 1-1 correspondence with the objects of $\mathbf{F}$, i.e. the natural numbers, whence $\mathbf{FTerm}_\Sigma$ is a PROP for the same reasons as $\mathsf{Cospan}(\mathbf{F})$.

The terminology *Frobenius termgraph* is justified by the following algebraic characterisation of $\mathbf{FTerm}_\Sigma$.

Theorem 3.3. $\mathbf{FTerm}_\Sigma \cong \mathbf{S}_\Sigma + \mathbf{Frob}$

Theorem 3.3 is crucial for our exposition as it serves as a bridge between algebraic and combinatorial structures. Indeed, it provides a presentation, by means of generators and equations, for the PROP $\mathbf{FTerm}_\Sigma$: the disjoint union of the SMTs of $\mathbf{S}_\Sigma$ and $\mathbf{Frob}$.

We remark that Theorem 3.3 generalises results already appearing in the literature (see e.g. [34, Proposition 3.2] and [17, Theorem 12], and the references therein) which however dealt only with operations of type $1 \to 1$. Here we consider operations with arbitrary (co)arities, thereby allowing us to deal with arbitrary SMTs.

Proof of Theorem 3.3. It suffices to verify that $\mathbf{FTerm}_\Sigma$ satisfies the universal property of coproducts in $\mathbf{PROP}$. We begin by defining morphisms

$$\lfloor\!\lfloor \cdot \rfloor\!\rfloor \colon \mathbf{S}_\Sigma \to \mathbf{FTerm}_\Sigma \text{ and } \psi : \mathbf{Frob} \to \mathbf{FTerm}_\Sigma.$$

Since $\mathbf{S}_\Sigma$ is the PROP freely generated by an SMT with no equations, it suffices to define $\lfloor\!\lfloor \cdot \rfloor\!\rfloor$ on the generators: for each $o\colon n \to m$ in Σ, we let $\lfloor\!\lfloor o \rfloor\!\rfloor$ be the following cospan of type $n \to m$.

(5)

For the definition of ψ, recall the isomorphism $\mathbf{Frob} \overset{\cong}{\to} \mathsf{Cospan}(\mathbf{F})$ of Example 2.1.(b). We let ψ be its composition with the trivial embedding $\mathsf{Cospan}(\mathbf{F}) \to \mathbf{FTerm}_\Sigma$ given by regarding a set as a discrete graph.

Now, given a PROP $\mathbf{X}$ and morphisms $\alpha : \mathbf{S}_\Sigma \to \mathbf{X}$, $\beta : \mathbf{Frob} \to \mathbf{X}$ we must show that there exists a unique morphism

---

[3] Since $\mathbf{F}$ has pushouts, so does $\mathbf{Hyp}$, where they are calculated pointwise.

$\upsilon : \mathbf{FTerm}_\Sigma \to \mathbf{X}$ making the following diagram commute.

$$
\begin{array}{ccc}
\mathbf{S}_\Sigma & \xrightarrow{\lfloor\!\lfloor\cdot\rfloor\!\rfloor} \mathbf{FTerm}_\Sigma \xleftarrow{\psi} & \mathbf{Frob} \\
& \searrow^{\alpha} \quad \downarrow_{\upsilon} \quad \swarrow_{\beta} & \\
& \mathbf{X} &
\end{array}
\qquad (6)
$$

To do this, we decompose—in an essentially unique way—any Frobenius termgraph into an expression where all the basic constituents lie in the image of ϕ and ψ. This is possible because we are working with discrete cospans: all of the action happens on the nodes, that intuitively serve as coat-hangers for the edges.

Let $n \xrightarrow{f} G \xleftarrow{g} n$ be in $\mathbf{FTerm}_\Sigma$. Then $G = (N, E, \tau)$ where N is the ordinal of nodes, $E = \bigcup_{k,l \in \mathbb{N}} E_{k,l}$ is a family of (typed) hyperedges and $\tau : E \to \Sigma$ is the type morphism. Since n, m are discrete, we actually have a cospan of *functions* $n \xrightarrow{f} N \xleftarrow{g} m$.

Given a hyperedge $e \in E$, $\lfloor\!\lfloor \tau e \rfloor\!\rfloor$ is as illustrated in (5). Define

$$
\tilde{n} \xrightarrow{i} \tilde{E} \xleftarrow{o} \tilde{m} := \bigoplus_{e \in E} \lfloor\!\lfloor \tau e \rfloor\!\rfloor.
\qquad (7)
$$

Intuitively, $\tilde{E}$ is the collection of the hyperedges of G, but disconnected, and $\tilde{n}$ and $\tilde{m}$ are the finite ordinals of all the inputs and outputs, concatenated. There are the induced canonical maps $j : \tilde{n} \to N$ and $p : \tilde{m} \to N$ that send a "disconnected" input or output node to the corresponding node in N. Now consider the following composition of three arrows in $\mathbf{FTerm}_\Sigma$.

$$
\begin{array}{ccccccc}
& N & & N \oplus \tilde{E} & & N & \\
f\nearrow & \nwarrow^{[id\ j]} & \nearrow & \nwarrow & \nearrow^{[id\ p]} & \nwarrow g & \\
n & & N + \tilde{n} & {\scriptstyle id\oplus i} \quad {\scriptstyle id\oplus o} & N + \tilde{m} & & m
\end{array}
\qquad (8)
$$

It is straightforward to compute the resulting pushouts in $\mathbf{F}$ and conclude that we obtain $n \xrightarrow{f} G \xleftarrow{g} m$ as a result. Since the constituents of (8) are all in the image of ϕ and ψ, this means that we can define υ on G. This assignment is well-defined since the fact that $\mathbf{X}$ is symmetric monoidal means the order of the edges in $\tilde{E}$ is immaterial. Moreover, it is unique since the decomposition is unique up-to permutation of $\tilde{E}$. $\qquad\square$

We are interested in $\mathbf{FTerm}_\Sigma$ as a combinatorial universe for rewriting in $\mathbf{S}_\Sigma$. In the remainder of this section we thus focus on the coproduct injection $\lfloor\!\lfloor\cdot\rfloor\!\rfloor : \mathbf{S}_\Sigma \to \mathbf{FTerm}_\Sigma$. Our first observation is that $\lfloor\!\lfloor\cdot\rfloor\!\rfloor$ is *faithful*; the proof of the following relies on properties of coproducts in $\mathbf{PROP}$.

Proposition 3.4. $\lfloor\!\lfloor\cdot\rfloor\!\rfloor : \mathbf{S}_\Sigma \to \mathbf{FTerm}_\Sigma$ *is faithful.*

We conclude this section with a combinatorial characterisation of the image of $\lfloor\!\lfloor\cdot\rfloor\!\rfloor$. A preliminary series of definitions introduces the relevant hypergraph notions: *monogamicity* and *acyclicity*.

Definition 3.5 (Degree of a node). The *indegree* of a node v in hypergraph G is the number of pairs (h, i) where h is an hyperedge with v as its i-th target. Similarly, the *outdegree* of v is the number of pairs (h, j) where h is an hyperedge with v as its j-th source.

Definition 3.6 (Monogamicity). Given $m \xrightarrow{f} G \xleftarrow{g} n$ in $\mathbf{FTerm}_\Sigma$, let $\mathsf{in}(G)$ be the image of f and $\mathsf{out}(G)$ the image of g. We say that G is *monogamous* if f and g are mono and, for all nodes v of G,

$$
\text{the indegree of } v \text{ is } \begin{cases} 0 & \text{if } v \in \mathsf{in}(G) \\ 1 & \text{otherwise.} \end{cases}
$$

$$
\text{the outdegree of } v \text{ is } \begin{cases} 0 & \text{if } v \in \mathsf{out}(G) \\ 1 & \text{otherwise} \end{cases}
$$

Example 3.7. The following three cospans are *not* monogamous.

$$
1 \to \bullet\!\!-\!\!\square\!\!-\!\!\bullet \leftarrow 1 \quad 1 \to \bullet\!\!\genfrac{}{}{0pt}{}{\square}{\square}\!\!\bullet \leftarrow 1 \quad 1 \to \bullet\!\!-\!\!\square\!\!-\!\!\bullet \leftarrow 1.
$$

Remark 3.8. There is a compact characterisation of monogamicity that uses (8): the factorised cospan is monogamous precisely when

$$
[f\ p] : n + \tilde{m} \to V \text{ and } [g\ j] : m + \tilde{n} \to V
$$

are bijections.

The notion of *directed path* from a node v to v' in a directed graph generalises to (directed) hypergraphs in the obvious way.

Definition 3.9 (Acyclicity). A hypergraph G is *directed acyclic* if there exists no directed path from a node to itself. We also call a cospan $n \to G \leftarrow m$ directed acyclic if the property holds for G.

Definition 3.10 (Convex sub-hypergraph). A sub-hypergraph $H \subseteq G$ is *convex* if, for any nodes v, v' in H and any directed path p from v to v', every hyperedge in p must also be in H.

Lemma 3.11. *Let $m \to G \leftarrow n$ be a monogamous directed acyclic cospan and L a convex sub-hypergraph. Then there exists $k \in \mathbb{N}$ and a unique cospan $i \to L \leftarrow j$ such that G factors as:*

$$
(m \to C_1 \leftarrow i+k)\ ;\ \begin{pmatrix} k \xrightarrow{id} k \xleftarrow{id} k \\ \oplus \\ (i \to L \leftarrow j) \end{pmatrix}\ ;\ (j+k \to C_2 \leftarrow n) \quad (9)
$$

where all cospans in (9) are monogamous directed acyclic.

Theorem 3.12. $n \to G \leftarrow m$ *in $\mathbf{FTerm}_\Sigma$ is in the image of $\lfloor\!\lfloor\cdot\rfloor\!\rfloor$ if and only if $n \to G \leftarrow m$ is monogamous directed acyclic.*

Proof. The only if direction follows by induction on d such that $\lfloor\!\lfloor d \rfloor\!\rfloor = n \to G \leftarrow m$. For the converse direction, we can reason by induction on the number of hyperedges in G. If G does not contain any, then monogamicity and acyclicity imply that $n \to G$ and $m \to G$ are bijections, so that $n \to G \leftarrow m$ is in the image of an arrow only consisting of identities and symmetries. If G consists of a single hyperedge e, it has a Σ-type $\tau e \in \Sigma$. Because $n \to G \leftarrow m$ is monogamous directed acyclic, $\tau e : n \to m$ as an arrow of $\mathbf{S}_\Sigma$ and $\lfloor\!\lfloor o : n \to m \rfloor\!\rfloor = n \to G \leftarrow m$.

For the inductive step, pick any hyperedge e of G. By monogamicity and acyclicity, $\lfloor\!\lfloor \tau e \rfloor\!\rfloor$ is a convex sub-hypergraph of G. Hence, by Lemma 3.11, $n \to G \leftarrow m$ factors as (9), with L being $\lfloor\!\lfloor \tau e \rfloor\!\rfloor$. The lemma guarantees that all the above cospans are monogamous directed acyclic. Therefore, by the inductive hypothesis they are in the image of $\lfloor\!\lfloor\cdot\rfloor\!\rfloor$, and so the same holds for $n \to G \leftarrow m$. $\qquad\square$

4. DPO rewriting on Frobenius termgraphs

In this section, we exploit Theorem 3.3 to establish an equivalence between rewriting in a PROP and *DPO rewriting*. The definition below, due to Ehrig and König [14], extends the classical DPO approach [15] with the "interface" m.

Definition 4.1 (DPO). A *DPO rule* is a span $L \leftarrow j \to R$ in $\mathbf{Hyp}_\Sigma$ where j is discrete. A *DPO rewriting system* $\mathcal{R}$ is a set of DPO rules. Given $d, e : 0 \to m$ in $\mathbf{FTerm}_\Sigma$, namely $d = 0 \to G \xleftarrow{p} m$ and $e = 0 \to H \xleftarrow{q} m$, we write that $d \rightsquigarrow_\mathcal{R} e$ if there exists $L \leftarrow j \to R$ in $\mathcal{R}$ and cospan $j \to C \leftarrow m$ s.t. the diagram (in $\mathbf{Hyp}_\Sigma$) below commutes and the squares are pushouts.

$$
\begin{array}{ccccc}
L & \longleftarrow & j & \longrightarrow & R \\
\downarrow & \ulcorner & \downarrow & \urcorner & \downarrow \\
G & \longleftarrow & C & \longrightarrow & H \\
& \nwarrow_p & \uparrow_m & \nearrow_q & \\
& & m & &
\end{array}
\qquad (10)
$$

The arrow $L \to G$ is called a *match* of L in G. See Examples 4.8 and 4.10 for illustrations of DPO rewriting.

The first step in relating PROP and DPO rewriting in $\mathbf{Hyp}_\Sigma$ is observing that the latter coincides with *ground* PROP rewriting.

Definition 4.2. A *ground* rewriting system on a PROP $\mathbf{X}$ is a collection of rules $\langle l, r \rangle : (i, j)$ with $i = 0$.

Lemma 4.3. *Let $\mathcal{R}$ be a ground rewriting system on a PROP $\mathbf{X}$ and $d, e : 0 \to m$ be in $\mathbf{X}$. Then $d \Rightarrow_\mathcal{R} e$ if and only if there are $\langle l, r \rangle : (0, j) \in \mathcal{R}$ and $c : j \to m$ such that*

$$d = l \,;\, c \quad and \quad e = r \,;\, c. \tag{11}$$

Proof. Suppose that $d \Rightarrow_\mathcal{R} e$. By definition d is as shown below left. It is equal to the diagram below right by the equations of SMCs. We define the context $c : j \to m$ as the dashed diagram.


Similarly, we have $e = r \,;\, c$. The other direction is immediate. $\square$

Observe that there is a 1-1 correspondence between ground systems and DPO systems in $\mathbf{FTerm}_\Sigma$. Indeed, any DPO rule $L \leftarrow j \to R$ uniquely induces, by initiality of 0 in $\mathbf{Hyp}_\Sigma$, a ground rule $\langle 0 \to L \leftarrow j, 0 \to R \leftarrow j \rangle$, and vice versa. Moreover, this correspondence lifts to the rewrite relations.

Theorem 4.4. *Let $\mathcal{R}$ be a ground rewriting system in $\mathbf{FTerm}_\Sigma$. Then, $d \Rightarrow_\mathcal{R} e$ iff $d \rightsquigarrow_\mathcal{R} e$.*

Proof. The conditions in (10) are the same as those on (11): it is enough to take c in (11) as $j \to C \leftarrow m$ in (10) and recall that composition in $\mathbf{FTerm}_\Sigma$ is defined by pushout. $\square$

This correspondence can be further extended to arbitrary—i.e. not necessarily ground—rewriting systems. Indeed, in PROPs freely generated by an SMT equipped with a chosen Frobenius structure, for any rewriting system there exists an equivalent ground one. The idea is to bend a rule $\langle l, r \rangle$ of type $n \to m$ into a ground rule of type $0 \to n + m$ by exploiting the compact closed structure introduced in Section 2. To make this intuition formal, for any morphism $d : n \to m$, we define $\ulcorner d \urcorner : 0 \to n + m$ as
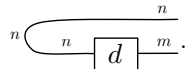
Then, for a system $\mathcal{R}$, we define the ground rewrite system $\ulcorner \mathcal{R} \urcorner$ as

$$\{ \langle \ulcorner l \urcorner, \ulcorner r \urcorner \rangle \mid \langle l, r \rangle \in \mathcal{R} \}$$

which, as stated in the following, is equivalent to $\mathcal{R}$.

Lemma 4.5. *Let $\mathcal{R}$ be a rewriting system on a PROP freely generated by an SMT with a chosen special Frobenius structure. Then*

$$d \Rightarrow_\mathcal{R} e \quad iff \quad \ulcorner d \urcorner \Rightarrow_{\ulcorner \mathcal{R} \urcorner} \ulcorner e \urcorner.$$

Proof. For the left-to-right direction, let $\langle l, r \rangle : i \to j$ be the applied rule. That means, for some c_1, c_2 and k,

$$\tag{12}$$

Now, using the compact-closed structure and the laws of SMCs we can transform $\ulcorner d \urcorner$ as

$$\tag{13}$$

There is an analogous computation for $\ulcorner e \urcorner$ exploiting (12).

$$\tag{14}$$

To conclude, observe that the outcome of (13) rewrites into the outcome of (14) with the rule $\langle \ulcorner l \urcorner, \ulcorner r \urcorner \rangle : i \to j$.

For the converse direction, suppose that $\ulcorner d \urcorner \Rightarrow_{\ulcorner \mathcal{R} \urcorner} \ulcorner e \urcorner$ via $\langle \ulcorner l \urcorner, \ulcorner r \urcorner \rangle : i \to j$. By Lemma 4.3 this means that, for some c,

$$\tag{15}$$

$$\tag{16}$$

Now, by (15) we can suitably shape d as follows using the compact-closed and symmetric monoidal structure.

$$\tag{17}$$

We act analogously on e exploiting (16).

$$\tag{18}$$

The result of (17) is in the shape $d_1 \,;\, (\mathrm{id} \oplus r) \,;\, d_2$ and that of (18) is of the form $d_1 \,;\, (\mathrm{id} \oplus r) \,;\, d_2$, whence $d \Rightarrow_\mathcal{R} e$ via rule $\langle l, r \rangle$. $\square$

We now combine the insights provided by Theorem 4.4 and Lemma 4.5 into the following theorem, which gives a tight correspondence between rewriting in $\mathbf{S}_\Sigma + \mathbf{Frob}$ and DPO rewriting in $\mathbf{FTerm}_\Sigma$. First, given a rewriting system $\mathcal{R}$ on $\mathbf{S}_\Sigma + \mathbf{Frob}$, we define the system $\Phi(\mathcal{R})$ on $\mathbf{FTerm}_\Sigma$ as $\{ \langle \Phi(l), \Phi(r) \rangle \mid \langle l, r \rangle \in \mathcal{R} \}$ where $\Phi : \mathbf{S}_\Sigma + \mathbf{Frob} \to \mathbf{FTerm}_\Sigma$ is the iso of Theorem 3.3.

Theorem 4.6. *Let $\mathcal{R}$ be any rewriting system on $\mathbf{S}_\Sigma + \mathbf{Frob}$. Then,*

$$d \Rightarrow_\mathcal{R} e \quad iff \quad \Phi(\ulcorner d \urcorner) \rightsquigarrow_{\Phi(\ulcorner \mathcal{R} \urcorner)} \Phi(\ulcorner e \urcorner).$$

Proof. Immediate from Lemma 4.5 and Theorems 3.3, 4.4. $\square$

A wide corpus of theorems, algorithm and tools has been developed in the last decade for DPO rewriting on *adhesive* categories [26]. These can be reused to deal for rewriting SMTs equipped with a Frobenius structure because of Theorem 4.6 and the following.

Proposition 4.7. $\mathbf{Hyp}_\Sigma$ *is adhesive.*

Proof. The PROP **F** of functions is adhesive for the same reasons that **Set** is. Adhesivity of $\mathbf{Hyp}_\Sigma$ then follows since it is a functor category over **F**, see [26, Prop. 3.5]. $\qquad\square$

The theory of adhesive categories says that pushout complements for a DPO rewriting as in (10) are uniquely defined when the arrow $j \to L$ is mono. However, this situation is not enforced by our approach, as shown by the following example.

Example 4.8. Consider $\Sigma = \{e_1 : 0 \to 1, e_2 : 1 \to 0, e_3 : 1 \to 1\}$ and a rewriting system $\mathcal{R}$ with the rule $\langle \text{———}, \, \boxed{e_3}\text{—}\rangle : (1,1)$. In $\mathbf{FTerm}_\Sigma$, it is captured by the DPO rule

$$\overset{0,1}{\bullet} \quad \longleftarrow \quad \overset{\bullet\,0}{\bullet\,1} \quad \longrightarrow \quad \overset{0}{\bullet}\boxed{e_3}\overset{1}{\bullet}$$

The left leg of the span is not mono, and therefore pushout complements are not necessarily unique for the application of this rule, as testified by the following two DPO rewriting steps.

$$(19)$$

$$(20)$$

The different outcome is due to the fact g maps 0 to the leftmost and 1 to the rightmost node, whereas g' swaps the assignments. Note that, as guaranteed by Theorem 4.6, both rewriting steps can be mimicked at the syntactic level in $\mathbf{S}_\Sigma + \mathbf{Frob}$. In the second case, one needs to use the compact closed structure.

$$\boxed{e_1}\text{—}\boxed{e_2} \qquad \Rightarrow_{\mathcal{R}} \qquad \boxed{e_1}\text{—}\boxed{e_3}\text{—}\boxed{e_2}$$

$$\boxed{e_1}\text{—}\boxed{e_2} = \boxed{e_1}\!\!\rightthreetimes\!\!\boxed{e_2} = \boxed{e_1}\!\!\times\!\!\boxed{e_2} \quad \Rightarrow_{\mathcal{R}} \quad \boxed{e_3}\big/\big\backslash\boxed{e_1}\boxed{e_2}$$

Example 4.9. A number of theories contain special Frobenius structures, see e.g. [3, 5, 10]. Our results mean that each theory can be seen as a smaller collection of DPO rewriting rules (cf. Proposition 2.3) on the associated hypergraphs, since a chosen Frobenius structure is implicitly handled by the combinatorial representation.

For example, consider the theory of interacting bimonoids from [5]. In that paper, there are two special Frobenius structure, one "black" (where the monoid and comonoid structures are drawn as in Fig. 2) and one "white" where the the generators are drawn as those in Example 2.1.(c). Let us consider the the black special Frobenius structure as the chosen one. Now, seen as hypergraphs, instead of eight generators, it suffices to consider two hyperedges that serve as the multiplication/comultiplication and unit/counit:

The two indicative steps below illustrate the translation from SMT equations to DPO rules.
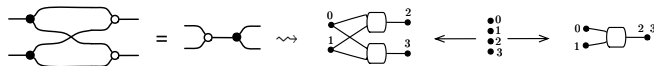
Example 4.10. In a similar fashion we would like to study the rewriting theory of the SMTs introduced in Example 2.1, in particular the case of non-commutative bimonoids (Ex.2.1.(c)). First, we orient the equations (3) from left to right, creating a rewriting system $\mathcal{R}_{\mathbf{NB}}$. We then obtain the ground rewriting system $\ulcorner\mathcal{R}_{\mathbf{NB}}\urcorner$ by bending each rule in $\mathcal{R}_{\mathbf{NB}}$ and, finally, via the translation $\Phi : \mathbf{S}_\Sigma + \mathbf{Frob} \overset{\cong}{\Rightarrow} \mathbf{FTerm}_\Sigma$ we obtain the DPO system $\Phi(\ulcorner\mathcal{R}_{\mathbf{NB}}\urcorner)$ — see Figure 3. In Section 6, we will show termination of $\Phi(\ulcorner\mathcal{R}_{\mathbf{NB}}\urcorner)$. For this to work, however, we first need a rewriting procedure that is sound and complete for rewriting in SMTs such as **NB** which do not have additional special Frobenius structure required by Theorem 4.6. This is the content of the next section.

5. The general symmetric monoidal case

In the general case, where there is no chosen special Frobenius structure, we can still use $\mathbf{FTerm}_\Sigma$ for rewriting, but we need to careful about how we find *legal* contexts. For instance, the context in (20) relies on the extra Frobenius structure being present.

The remit of this section is to tailor a suitable restriction of DPO rewriting that adheres to rewriting for an SMT as closely as ordinary DPO rewriting corresponds to rewriting in an SMT with a chosen Frobenius structure (Theorem 4.6).

Before the formal developments, let us sketch the intuition. Given a rewriting system $\mathcal{R}$, we are interested in the shape of contexts for $\ulcorner\mathcal{R}\urcorner$-rewriting in $\mathbf{S}_\Sigma + \mathbf{Frob} \cong \mathbf{FTerm}_\Sigma$ that "look like" legal contexts for $\mathcal{R}$-rewriting in $\mathbf{S}_\Sigma$. By Lemma 4.5, a generic context in $\mathbf{S}_\Sigma + \mathbf{Frob}$ for a rule $\langle\ulcorner l\urcorner, \ulcorner r\urcorner\rangle$ has shape
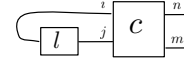
Now, this context corresponds to a legal one in $\mathbf{S}_\Sigma$ only when we can find c_1, c_2 in $\mathbf{S}_\Sigma$ such that

$$(21)$$

Indeed, in that case, we are in presence of the bent version of a legal context in $\mathbf{S}_\Sigma$ for rule $\langle l, r\rangle$ (cf. (13)):
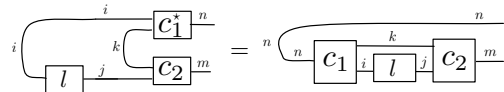
Our task is to characterise legal contexts like (21). We proceed in two steps. First, we introduce the notion of *boundary complement* (Definition 5.1), which rests on monogamicity (Definition 3.6). It clarifies the role that interfaces i, j, m, n play in the legal context (21): j, n serve as inputs and i, m serve as outputs of its un-bent version. Contrary to arbitrary contexts in $\mathbf{FTerm}_\Sigma$ (*cf.* Example 4.8), boundary complements are uniquely defined up-to iso when they exist (Proposition 5.2). The second notion is *convex matching* (Definition 5.4), which enforces that the only communication between c_1 and c_2 happens along the interface k, *cf.* (21).

A convex matching implies the existence of a unique boundary complement as context: we call this situation a *convex* a DPO rewriting step (Definition 5.5). The main result is the adequacy of convex DPO rewriting for rewriting in an SMT (Theorem 5.6).

Definition 5.1 (Boundary complement). For monogamous cospans $i \xrightarrow{a_1} L \xleftarrow{a_2} j$ and $n \xrightarrow{b_1} G \xleftarrow{b_2} m$, and a mono $f : L \to G$, a

pushout complement as (†) below

$$
\begin{array}{ccc}
L & \xleftarrow{\; a:=[a_1,a_2]\;} & i+j \\[2pt]
{\scriptstyle f}\big\downarrow & \quad(\dagger)\quad\; \ulcorner & \big\downarrow{\scriptstyle c:=[c_1,c_2]} \\[2pt]
G & \xleftarrow{\;\;g\;\;} & L^{\perp} \\
\end{array}
$$

is called a *boundary complement* if c is mono and there exists $d_1\colon n \to L^{\perp}$ and $d_2\colon m \to L^{\perp}$, making the above triangle commute, such that

$$
j + n \xrightarrow{\;[c_2,d_1]\;} L^{\perp} \xleftarrow{\;[c_1,d_2]\;} m + i \tag{22}
$$

is a monogamous cospan.

Proposition 5.2. *Boundary complements in $\mathbf{Hyp}_\Sigma$ are unique, when they exist.*

As mentioned, the notion of boundary complement does not tell the whole story about which matches in $\mathbf{FTerm}_\Sigma$ are legal in $\mathbf{S}_\Sigma$. The missing requirement is that $c_1^\star$ and c_2 in (21) can only communicate through interface k, which connects outputs of c_1 to inputs of c_2. The following counterexample shows that, even in presence of a boundary complement, linking outputs of c_2 to inputs of c_1 may still yield an illegal rewriting step in $\mathbf{S}_\Sigma$.

Example 5.3. Fix $\Sigma = \{e_1\colon 1 \to 2, e_2\colon 2 \to 1, e_3\colon 1 \to 1, e_4\colon 1 \to 1\}$ and consider the following rewriting rule in $\mathbf{S}_\Sigma$.

$$
\langle \;\boxed{e_1}\!\!\boxed{e_2}\; , \; \boxed{\genfrac{}{}{0pt}{}{e_4}{e_4}}\;\rangle \colon (2,2) \tag{23}
$$

Left and right side are interpreted in $\mathbf{FTerm}_\Sigma$ as cospans

$$
2 \to \;\boxed{e_1}\,\boxed{e_2}\; \leftarrow 2 \qquad 2 \to \;\boxed{\genfrac{}{}{0pt}{}{e_4}{e_4}}\; \leftarrow 2.
$$

We introduce another diagram $c\colon 1 \to 1$ in $\mathbf{S}_\Sigma$ and its interpretation in $\mathbf{FTerm}_\Sigma$:

$$
\boxed{e_1}\boxed{e_3}\boxed{e_2} \;\xmapsto{\;\lfloor\!\lfloor\cdot\rfloor\!\rfloor\;}\; 1 \to \boxed{e_1}\boxed{e_3}\boxed{e_2} \leftarrow 1.
$$

Now, the left-hand side of the rule (23) cannot match in c. However, their interpretation yields a DPO rewriting step in $\mathbf{FTerm}_\Sigma$ as below, where f maps grey nodes to grey nodes.

Observe that the leftmost pushout above *is* a boundary complement: the input-output partition is correct. Still, mimicking this rewriting step in $\mathbf{S}_\Sigma$ fails because the context does not fit the legal shape (21). This is most evident by showing how matching in $\lfloor\!\lfloor c\rfloor\!\rfloor$ appears under the isomorphism $\mathbf{FTerm}_\Sigma \cong \mathbf{S}_\Sigma + \mathbf{Frob}$:

The diagrams l, c_1 and c_2 on the right side are defined in terms of the dotted squares on the right. The difference with the prescription of (21) is apparent: the topmost output interface of c_2 connects to the bottommost input interface of c_1. For this reason, we cannot un-bend our context to obtain a legal matching in $\mathbf{S}_\Sigma$.

In order to rule out situations (23) and complete the characterisation of legal contexts like (21), we recover the notion of convexity (Def. 3.10) to introduce *convex* matchings. The intuition is that a convex matching cannot leave "holes" as f does in Example 5.3.

Definition 5.4 (Convex matching). We call $m\colon L \to G$ in $\mathbf{Hyp}_\Sigma$ a *convex matching* if it is mono and its image is convex.

Lemma 3.11 and Theorem 3.12 are testimony to the importance of the convex matchings: given such a match, it is always possible to recover the appropriate context in $\mathbf{S}_\Sigma$. We now combine the notions of boundary complement and of convex matching to tailor a family of DPO rewriting steps which only yield legal $\mathbf{S}_\Sigma$-rewriting.

Definition 5.5 (Convex DPO rewriting step). Let $\mathcal{R}$ be a DPO rewriting system. A *convex* DPO rewriting step

$$
0\;\nearrow\!\!\searrow\!\! \begin{array}{c} D \xleftarrow{[q_1,q_2]} \\ n+m \end{array} \quad\Rightarrow_{\mathcal{R}}\quad 0\;\nearrow\!\!\searrow\!\! \begin{array}{c} E \xleftarrow{[p_1,p_2]} \\ n+m \end{array}
$$

happens when there is a rule $L \xleftarrow{[a_1,a_2]} i+j \xrightarrow{[b_1,b_2]} R$ in $\mathcal{R}$ such that the following diagram in $\mathbf{Hyp}_\Sigma$ commutes

$$
\begin{array}{ccccc}
L & \xleftarrow{[a_1,a_2]} & i+j & \xrightarrow{[b_1,b_2]} & R \\
{\scriptstyle f}\big\downarrow & \ulcorner & \big\downarrow & \urcorner & \big\downarrow \\
D & \longleftarrow & C & \longrightarrow & E \\
& {\scriptstyle [q_1,q_2]}\searrow & \big\downarrow & \swarrow{\scriptstyle [p_1,p_2]} & \\
& & n+m & &
\end{array} \tag{24}
$$

the two squares are pushouts, and the following conditions hold:

- $f\colon L \to G$ is a convex matching;
- in the leftmost pushout $i+j \to C$ is a boundary complement.

Note that the relation $\Rightarrow_{\mathcal{R}}$ is contained in $\leadsto_{\mathcal{R}}$ (Definition 4.1), the difference being that the leftmost pushout has to rest on a convex matching and a boundary complement.

We have now all the ingredients to prove the adequacy of convex DPO rewriting with respect to rewriting in $\mathbf{S}_\Sigma$.

Theorem 5.6. *Let $\mathcal{R}$ by any rewriting system on $\mathbf{S}_\Sigma$. Then,*

$$
d \Rightarrow_{\mathcal{R}} e \quad \Leftrightarrow \quad \Phi(\ulcorner d \urcorner) \Rightarrow_{\Phi(\ulcorner\mathcal{R}\urcorner)} \Phi(\ulcorner e \urcorner).
$$

Proof. For the only if direction, by Theorem 4.6 $d \Rightarrow_{\mathcal{R}} e$ implies $\Phi(\ulcorner d \urcorner) \leadsto_{\Phi(\ulcorner\mathcal{R}\urcorner)} \Phi(\ulcorner e \urcorner)$. It is not difficult to check that the argument actually constructs a *convex* DPO rewriting step, thus yielding the desired statement.

We now turn to the converse direction. Let

$$
\Phi(\ulcorner d \urcorner) =: \;0\;\nearrow\!\!\searrow\!\! \begin{array}{c} D \xleftarrow{[q_1,q_2]} \\ n+m \end{array} \qquad \Phi(\ulcorner e \urcorner) =: \;0\;\nearrow\!\!\searrow\!\! \begin{array}{c} E \xleftarrow{[p_1,p_2]} \\ n+m \end{array}.
$$

Our assumption gives us a diagram as in (24), with application of a rule $\langle\Phi(\ulcorner l \urcorner), \Phi(\ulcorner r \urcorner)\rangle$ in $\Phi(\ulcorner\mathcal{R}\urcorner)$. We now want to show that $d \Rightarrow_{\mathcal{R}} e$ with rule $\langle l,r\rangle$, say of type (i,j). Now, because $n \xrightarrow{q_1} D \xleftarrow{q_2} m = \lfloor\!\lfloor d\rfloor\!\rfloor$, it is monogamous directed acyclic by Theorem 3.12. Since the matching $f\colon L \to D$ in (24) is convex, Lemma 3.11 yields a decomposition of $n \xrightarrow{q_1} D \xleftarrow{q_2} m$ in terms

of monogamous directed acyclic cospans:

$$(n \to C_1 \leftarrow i+k) \; ; \; \begin{pmatrix} k \xrightarrow{id} k \xleftarrow{id} k \\ \oplus \\ (i \to L \leftarrow j) \end{pmatrix} \; ; \; (j+k \to C_2 \leftarrow m).$$

Applying again Theorem 3.12 we obtain c_1, c_2 in $\mathbf{S}_\Sigma$ such that

$$\lfloor\lfloor c_1 \rfloor\rfloor = n \to C_1 \leftarrow i + k \qquad \lfloor\lfloor c_2 \rfloor\rfloor = j + k \to C_2 \leftarrow m.$$

By functoriality of $\lfloor\lfloor \cdot \rfloor\rfloor$, $\lfloor\lfloor d \rfloor\rfloor = \lfloor\lfloor c_1 \; ; \; (id \oplus l) \; ; \; c_2 \rfloor\rfloor$ and, since $\lfloor\lfloor \cdot \rfloor\rfloor$ is a faithful prop morphism, $d = c_1 \; ; \; (id \oplus l) \; ; \; c_2$. Thus we can apply the rule $\langle l, r \rangle$ on e, which yields $e = c_1 \; ; \; (id \oplus r) \; ; \; c_2$ such that $d \Rightarrow_{\mathcal{R}} e$. We can conclude that $\lfloor\lfloor e \rfloor\rfloor = n \xrightarrow{p_1} E \xleftarrow{p_2} m$ because boundary complements are unique (Proposition 5.2). $\square$

6. Case study: non-commutative bimonoids

Thanks to Theorem 5.6, we can now return to Example 4.10 and study the DPO rewriting theory of non-commutative bimonoids (Example 2.1.(c)), shown in Figure 3. As a proof of concept, we focus on proving *termination* for this system. First, we construct a metric based on two kinds of paths and a weight.

- A *U-path* is a path p from an input or an η-hyperedge to an output or an ϵ-hyperedge.

- An *M-path* is a path from a μ-hyperedge to a ν-hyperedge.

A μ-tree with root x is a maximal tree of μ-hyperedges with output x. Similarly, a ν-tree with root x is a maximal tree of ν-hyperedges with input x.

For a μ-hyperedge h, let the L-weight $\ell(h)$ be the size of the μ-tree whose root is the first input of μ. Similarly, for a ν-hyperedge, let $\ell(h)$ be the size of the ν-tree whose root is the first output of h. Let $\ell(h) = 0$ otherwise and:

$$G \preceq_L H \iff \sum_{h \in E_G} \ell(h) \leq \sum_{h \in E_H} \ell(h).$$

Next, define orders $\preceq_U, \preceq_M, \preceq_\mu, \preceq_\nu$ based on counting the number of U-paths, M-paths, μ-hyperedges, and ν-hyperedges, respectively. Armed with the five orders, we define the following lexicographic ordering that combines them as its components:

$$\preceq := \text{lex}(\preceq_U, \preceq_M, \preceq_\mu, \preceq_\nu, \preceq_L) \qquad (25)$$

Clearly, $\preceq$ is well-founded, thus we can conclude as follows.

Theorem 6.1. *NB is strictly decreasing in $\preceq$, thus it terminates.*

Proof. We argue rule-by-rule, showing that each is strictly decreasing in one of the orders from (25), and non-increasing in every order that is prior in the lexicographic ordering.

Since every rule $\mathbf{NB}_j$ has a unique path from every input to every output on the LHS and RHS, applications of these rules have no effect on paths which start and finish outside of their image. Hence, for each rule, we only need to consider paths which start or terminate in the image of the LHS.

$\mathbf{NB}_1$ has no effect on η or ϵ hyperedges, hence no effect on $\preceq_U$. No M-path can terminate in $\mathbf{NB}_1$ and any M-path originating on $\mathbf{NB}_1$ must exit through the unique output. Since there are precisely two μ-hyperedges in the LHS and RHS, there is a 1-to-1 correspondence between M-paths before and after applying the rule. $\mathbf{NB}_1$ also leaves the number of μ and ν hyperedges fixed, so we only need to show it strictly decreases $\preceq_L$. Applying the rule has no effect on the L-weight of any μ-hyperedges outside of the image of the LHS. Suppose there are μ-trees of size a, b, c connected to inputs $0, 1, 2$ of the LHS, respectively. Then, the L-weight of the two μ-hyperedges on the LHS are a and $a + b + 1$, whereas on the LHS

they are a and b. Hence $\preceq_L$ is strictly decreased. $\mathbf{NB}_2$ follows via a symmetric argument.

Since $\mathbf{NB}_3$–$\mathbf{NB}_6$ and $\mathbf{NB}_{10}$ remove η- and ϵ-hyperedges from the hypergraph, they will strictly decrease the number of U-paths.

For $\mathbf{NB}_7$, no U-path can terminate in the LHS, and any U-path starting in LHS must exit through one of the two outputs. Hence it corresponds to a unique U-path exiting the RHS. M-paths are unaffected, as is the number of μ-hyperedges. However, the number of ν-hyperedges is strictly decreased, so $\mathbf{NB}_7$ strictly decreases $\preceq_\nu$. The argument for $\mathbf{NB}_8$ is again symmetric.

$\mathbf{NB}_9$ has no η or ϵ-hyperedges in its LHS or RHS, so it leaves the number of U-paths fixed. Consider an M-path which enters the LHS from the left. It enters either from input 0 or input 1, hence it corresponds to a unique M-path entering the RHS. We can argue similarly for M-paths exiting on the right. Hence, the only M-path left to consider is the one from the μ-hyperedge to the ν-hyperedge in the LHS, which is eliminated. Thus $\mathbf{NB}_9$ strictly reduces $\preceq_M$. $\square$

Hence, the combinatorial presentation gives a short and simple proof of termination. We conjecture that one can construct a similarly simple proof of confluence, via analysis of critical pairs. There is an existing notion of critical pair in the DPO literature, of which $\mathbf{NB}$ rewrite system has 24 (all of which are 'joinable'). There are, however, subtleties arising in critical pair analysis for DPO rewrite systems [32] and general rewriting for SMCs [27]. We leave development of a comprehensive theoretical framework for critical pair analysis on convex DPO rewrite systems as future work.

7. Conclusions

Our approach rests on the bridge between algebraic and combinatorial descriptions of string diagrams, in the form of the isomorphism $\mathbf{S}_\Sigma + \mathbf{Frob} \cong \mathbf{FTerm}_\Sigma$. We used it to spell out two fundamental ways in which rewriting in symmetric monoidal categories and DPO hypergraph rewriting are compatible.

The first is that we can use DPO to rewrite SMCs generated by theories containing a special Frobenius structure. Increasingly theories of that kind are appearing in diverse research threads: they play a role in the compositional study of Petri nets [8, 33, 39], in the algebra of concurrent connectors [7], in signal flow diagrams [2, 4, 6, 16] and in the ZX-calculus [10, 11], just to mention a few examples. Our approach enables the use of the technology that has been developed for DPO in the last decades in those areas.

The second approach – convex DPO rewriting – allows to rewrite in categories presented by arbitrary symmetric monoidal theories. Our results lay the theoretical foundations for the development of comprehensive tool support for reasoning about rewriting in SMCs. Emerging tools such as Globular [1] do not factor in the symmetric structure, whereas Quantomatic [24] relies on additional structure, requiring for example that the ambient category be traced. Our theory overcomes these limitations. Furthermore, under the encoding of hypergraphs as string graphs (the underlying theory used by Quantomatic), it is now possible to extend Quantomatic to support rewriting in a generic SMC. We leave the algorithmic and implementation details as future work.

Just as in term rewriting, commutative operators pose a interesting problem: naively encoding commutativity in a rule immediately yields a non-terminating system. We leave this issue as future work, but preliminary results suggest that the commutativity of some operators can be effectively encoded in the graphical structure.

From a theoretical viewpoint, we intend to build on this work to investigate the relationship between distributive laws of PROPs [25, 42] and rewriting. Compatibility conditions expressed by a distributive law come with an orientation, making natural to think of them as rewriting rules. Just as in our approach, these conditions are de-

Figure 3. DPO Rewriting system $\Phi(\ulcorner\mathcal{R}_{\mathbf{NB}}\urcorner)$ — *cf.* Example 4.10.
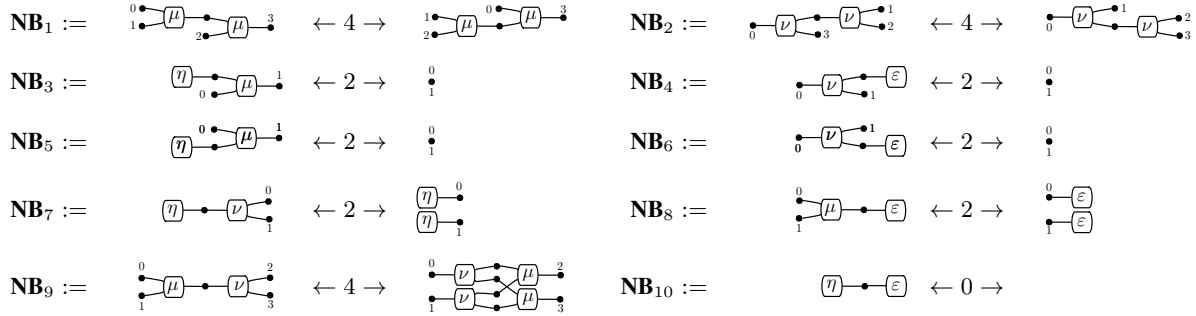
fined modulo the laws of SMCs: it thus appears promising to explore the correspondence between the two perspectives.

References

[1] Globular. http://globular.science/. Accessed: 2016-01-18.

[2] J. Baez and J. Erbele. Categories in control. *CoRR/1405.6881*, 2014.

[3] J. C. Baez and B. Fong. A compositional framework for passive linear circuits. *CoRR*, abs/1504.05625, 2015.

[4] F. Bonchi, P. Sobocinski, and F. Zanasi. A categorical semantics of signal flow graphs. In *CONCUR 2014*, pages 435–450, 2014.

[5] F. Bonchi, P. Sobociński, and F. Zanasi. Interacting Hopf algebras. *CoRR*, abs/1403.7048, 2014.

[6] F. Bonchi, P. Sobociński, and F. Zanasi. Full abstraction for signal flow graphs. In *POPL 2015*, pages 515–526, 2015.

[7] R. Bruni, I. Lanese, and U. Montanari. A basic algebra of stateless connectors. *Theor Comput Sci*, 366:98–120, 2006.

[8] R. Bruni, H. C. Melgratti, and U. Montanari. A connector algebra for P/T nets interactions. In *CONCUR '11*, pages 312–326, 2011.

[9] A. Burroni. Higher dimensional word problems with applications to equational logic. *Theor Comput Sci*, 115:43–62, 1993.

[10] B. Coecke and R. Duncan. Interacting quantum observables. In *ICALP'08*, pages 298–310, 2008.

[11] B. Coecke, R. Duncan, A. Kissinger, and Q. Wang. Strong complementarity and non-locality in categorical quantum mechanics. In *LiCS'12*, pages 245–254, 2012.

[12] A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via GS-monoidal categories. *Appl Cat S*, 7(4):299–331, 1999.

[13] L. Dixon and A. Kissinger. Open-graphs and monoidal theories. *Math Str Comput Sci*, 23:308–359, 4 2013.

[14] H. Ehrig and B. König. Deriving bisimulation congruences in the DPO approach to graph rewriting. In *FoSSaCS '04*, LNCS, 2004.

[15] H. Ehrig and H.-J. Kreowski. Parallelism of manipulations in multidimensional information structures. In *MFCS '76*, LNCS, 1976.

[16] B. Fong, P. Rapisarda, and P. Sobociński. A categorical approach to open & interconnected dynamical systems. In *LiCS'16*, 2016.

[17] F. Gadducci and R. Heckel. An inductive view of graph transformation. In *WADT 1997*, volume 1376. Springer, 1997.

[18] Y. Guiraud. Termination orders for 3-dimensional rewriting. *CoRR*, abs/math/0612083, 2006.

[19] M. Hyland and J. Power. Lawvere theories and monads. *ENTCS*, 172:437–458, 2007.

[20] A. Joyal and R. Street. The geometry of tensor calculus, I. *Adv Math*, 88:55–112, 1991.

[21] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Math Proc Cambridge Phil Soc*, 119:447–468, 4 1996.

[22] C. Kassel. *Quantum Groups*. Springer, 2012.

[23] G. M. Kelly and M. L. Laplaza. Coherence for compact closed categories. *J Pure Appl Algebra*, 19:193–213, 1980.

[24] A. Kissinger and V. Zamdzhiev. Quantomatic: A proof assistant for diagrammatic reasoning. *CoRR*, abs/1503.01034, 2015.

[25] S. Lack. Composing PROPs. *Theor App Cats*, 13(9):147–163, 2004.

[26] S. Lack and P. Sobociński. Adhesive and quasiadhesive categories. *RAIRO-Theor. Inf. Appl.*, 39(3):511–546, 2005.

[27] Y. Lafont. Towards an algebraic theory of boolean circuits. *J Pure Appl Alg*, 184:257–310, 2003.

[28] S. Mac Lane. Categorical algebra. *B Am Math Soc*, 71:40–106, 1965.

[29] J. MacDonald and L. Scull. Amalgamations of categories. *Can Math B*, 52:273–284, 2009.

[30] J. Meseguer and U. Montanari. Petri nets are monoids. *Inform Comput*, 88:105–155, 1990.

[31] S. Mimram. Towards 3-dimensional rewriting theory. *Log Meth Comput Sci*, 10(2), 2014.

[32] D. Plump. Hypergraph rewriting: Critical pairs and undecidability of confluence. In *Term Graph Rewriting*, pages 201–213. Wiley, 1993.

[33] J. Rathke, P. Sobociński, and O. Stephens. Compositional reachability in Petri nets. In *Reachability Problems '14*, 2014.

[34] R. Rosebrugh, N. Sabadini, and R. F. C. Walters. Generic commutative separable algebras and cospans of graphs. *Th Appl Categories*, 17(6):164–177, 2005.

[35] V. Sassone and P. Sobociński. Reactive systems over cospans. In *LiCS '05*, pages 311–320. IEEE Press, 2005.

[36] P. Selinger. Dagger compact closed categories and completely positive maps. *ENTCS*, 170:139–163, 2007.

[37] P. Selinger. A survey of graphical languages for monoidal categories. *Springer Lecture Notes in Physics*, 13(813):289–355, 2011.

[38] M. R. Sleep, M. J. Plasmeijer, and M. C. J. D. van Eekelen, editors. *Term Graph Rewriting: Theory and Practice*. Wiley, 1993.

[39] P. Sobociński and O. Stephens. A programming language for spatial distribution of net systems. In *Petri Nets '14*, 2014.

[40] R. Street. Limits indexed by category-valued 2-functors. *J Pure Appl Algebra*, 8(2):149 – 181, 1976.

[41] R. F. C. Walters. The tensor product of matrices. Lecture, International Conference on Category Theory, Louvain-la-Neuve, 1987.

[42] F. Zanasi. *Interacting Hopf Algebras: the theory of linear systems*. PhD thesis, Ecole Normale Supérieure de Lyon, 2015.