Rewriting in Free Hypegraph Categories

Fabio Zanasi

University College London, United Kingdom

`f.zanasi@ucl.ac.uk`

We study rewriting for equational theories in the context of symmetric monoidal categories where there is a separable Frobenius monoid on each object. These categories, also called hypergraph categories, are increasingly relevant: Frobenius structures recently appeared in cross-disciplinary applications, including the study of quantum processes, dynamical systems and natural language processing. In this work we give a combinatorial characterisation of arrows of a free hypergraph category as cospans of labelled hypergraphs and establish a precise correspondence between rewriting modulo Frobenius structure on the one hand and double-pushout rewriting of hypergraphs on the other. This interpretation allows to use results on hypergraphs to ensure decidability of confluence for rewriting in a free hypergraph category. Our results generalise previous approaches where only categories generated by a single object (props) were considered.

1 Introduction

Symmetric monoidal categories (SMCs) are an increasingly popular mathematical framework for the formal analysis of network-style diagrammatic languages that are found across different disciplines. In an SMC arrows have two composition operations, intuitively corresponding to sequential ($c\,;d$) and parallel ($c\oplus d$) combination of compound systems, and there are symmetry arrows $\asymp$, intuitively representing tangles of wires. These constructs are traditionally rendered by the two-dimensional notation of *string diagrams*, which has the key advantage of absorbing most of the structural equalities prescribed by the definition of SMC. For instance, the two sides of the exchange law $(a_1\,;a_2)\oplus(b_1\,;b_2) = (a_1\oplus b_1)\,;(a_2\oplus b_2)$ are encoded by the same string diagram $\boxed{a_1}\boxed{a_2}\atop\boxed{b_1}\boxed{b_2}$. The graphical syntax emphasises *connectivity* and *resource-exchange* between components, which makes it particularly effective in the analysis of challenging computational models such as distributed systems (based on threads communication) and quantum processes (powered by a notion of non-separable—entangled—states).

Some applications demand SMCs with a richer structure. In this paper we focus on *hypergraph categories*, which are SMCs where each object x is equipped with a *separable Frobenius monoid*. That means, for each x there are operations as on the left, forming a commutative monoid and a commutative comonoid that interact according to the Frobenius law and the separability law, as on the right.

$$\tag{1}$$

The core intuition is that this extra structure allows dangling wires of a string diagram to fork, be discarded, be moved to the left/right side, resulting in a more flexible manipulation of the interfaces (variables, memory cells) of the represented system. The use of hypergraph categories as an algebraic approach to computation was pioneered by Walters and collaborators [9, 20], under the name of well-supported compact closed categories. Since then separable Frobenius monoids have appeared ubiquitously in diagrammatic calculi across diverse research threads. They notably feature in the ZX-calculus [10] (quantum theory), where each Frobenius structure has a precise physical meaning in terms

of quantum observables. Frobenius monoids also form the backbone of the calculus of stateless connectors [7], the calculus of signal flow diagrams [4, 5], Baez's network theory [1] and Pavlovic's monoidal computer [28]. More recently, a particular attention has been devoted to generic constructions of hypergraph categories through abstract notions of span, relation and their duals [35, 15, 26, 16].

Whereas separable Frobenius monoids constitute a common core for the aforementioned approaches, in each application string diagrams are further quotiented by domain-specific equations, instrumental in defining the appropriate notion of behavioural equality of systems. The perspective of this work is to acknowledge the conceptual difference between the symmetric monoidal and Frobenius structure on the one hand, which is a built-in part of any hypergraph category, and the domain-specific equations on the other hand. We shall study the latter as *rewriting rules*: if the left hand side of such an equation can be found in a larger string diagram, it can be deleted and replaced with its right hand side.

This is coherent with the everyday practice of users of diagrammatic calculi and is the starting point for implementing graphical reasoning in a proof assistant. There is a thorough mathematical theory of rewriting for monoidal categories, which regards rewrite rules as generator 2-cells (variously called computads [21] or polygraphs [8]) and the possible rewriting trajectories as composite two-cells. However, this abstract perspective does not provide immediate help when it comes to *implementing* rewriting. The main challenge is a concrete understanding of matching: in order to detect whether a string diagram contains the left-hand side of a rewriting rule, one needs to consider all its possible decompositions according to the structural equations. In an hypergraph category, this amounts to say that rewriting happens modulo the equations of separable Frobenius monoids. For instance, the rewriting rule

$$ -\boxed{c\ O_1}\ \overset{c}{\underset{d}{\boxed{O_1}}}\ \overset{c}{\underset{d}{\boxed{O_2}}}\ c- \quad \Rightarrow \quad -\boxed{c\ O_1}\ \overset{c}{\underset{d}{}}\overset{d}{\underset{d}{\bullet\boxed{O_2}}}\ c- $$

applies in the leftmost string diagram below, module the separable Frobenius structure on objects c, d.

$$ \tag{2} $$

We shall develop a combinatorial characterisation of this rewriting mechanism, in three steps:

- the first step (Section 3) is to give a combinatorial description of the string diagrams in free hypergraph categories. Our choice is a category of cospans of hypergraphs, which is shown to be isomorphic to the free hypergraph category generated by a signature. This data structure encapsulates all the equivalent representations of a string diagram modulo Frobenius into a single object, thus easing the complexity of matching. The idea of the isomorphism is that boxes in a string diagram are represented as hyperedges, and wires as nodes. The use of cospans $I_l \xrightarrow{f} G \xleftarrow{g} I_r$ is essential: the carrier G encodes the string diagram itself, whereas I_l and I_r are discrete hypergraphs (sets of nodes) that indicate through f and g which nodes of G are dangling wires on the left and on the right of the corresponding string diagram.

  As the name suggests, the close relationship between hypergraph categories and hypergraph structures was clear to previous authors [22], as well as the use of cospans to mimic interfaces [14, 31]. Our characterisation combines existing approaches in a way that best suits the application to rewriting. The main generalisation is characterising hypergraph categories that are freely generated by multi-sorted instead of single-sorted signatures.

- The second step (Section 4) is to exploit the combinatorial interpretation to realise string diagram rewriting as rewriting of hypergraphs. The fact that cospans of hypergraphs form an adhesive category [24] gives an off-the-shelf theory of double-pushout (DPO) rewriting. We show that rewriting

modulo Frobenius and DPO rewriting of hypergraphs are essentially the same thing. The problem of matching in an hypergraph category (2) is reduced to finding an hypergraph homomorphism.

- As last contribution (Section 5), we show that confluence for terminating rewriting systems in hypergraph categories is decidable, as it is reducible to a computable critical pair analysis. This well-known property of term rewriting becomes a non-trivial question when dealing with two-dimensional entities. For instance, in the aforementioned polygraph approach [8], where critical pairs are considered in the string diagrammatic syntax rather than in a graph model, even a finite set of rewriting rules may yield infinitely many critical pairs [27]. In the context of ordinary DPO graph rewriting decidability also fails [29] unless further conditions are imposed, such as requiring that all critical pairs satisfy a syntactic condition called coverability [30] or that they are joinable in a stricter sense [13]. We establish our decidability result within the framework of recent work [3] that studies confluence for DPO hypergraph rewriting *with interfaces*. Not only this variant enjoys decidability without further restrictions on critical pairs, but is precisely tailored for the interpretation of "syntactic" rewriting from hypergraph categories. We refer to [3] for a more extensive discussion of how the interface approach compares to others in the DPO rewriting literature.

Rewriting modulo Frobenius structure has been studied along the same lines in [2, 3]. These recent works by the author and collaborators serve as a roadmap for this paper: the aim here is to verify that such results generalise to multi-sorted algebraic theories, for which the freely generated category has a Frobenius structure on each sort. In light of [2, 3], the way this generalisation unfolds is not particularly surprising, as we are essentially able to lift the same proof techniques from a single to multiple sorts. However, we believe that the redaction of a reference paper for these results is timely. Firstly, it is justified by the renewed interest for hypergraph categories, witnessed by several recent applications, especially to circuit theory [15] and to natural language semantics [26, 19]: using the theory developed in [2, 3] is going to require the full generality of the multi-sorted case. Secondly, another justification comes from axiomatic approaches to various families of systems (concurrent [7], quantum [10], dynamical [4, 1]) in which the equational theory axiomatising system behaviour includes two or more Frobenius algebras. When it comes to rewriting, the approach introduced in [2] only allows to absorb *one* Frobenius structure in the combinatorial model. In this paper, we show how additional Frobenius structures can be also absorbed in the same manner[1], thus reducing the complexity of the aforementioned axiomatisations and simplifying the task of studying normal forms, confluence and termination.

Notation. In a category $\mathbf{C}$ with coproducts, $(h_1, h_2) \colon X + Y \to Z$ is the copairing of $h_1 \colon X \to Z$ and $h_2 \colon Y \to Z$, defined by universal property of $+$. Also, $f \,;\, g \colon a \to c$ is the composition of arrows $f \colon a \to b$, $g \colon b \to c$. We sometimes write $a \xrightarrow{f} b$ or $b \xleftarrow{f} a$ for $f \colon a \to b$, or also $\xrightarrow{f}$ and $\xleftarrow{f}$ if object names are immaterial for the context. We write $\oplus$ for the monoidal product in a monoidal category.

2   Props and Hypergraph Categories

We are going to study hypergraph categories freely generated by a signature of operations. The following is the notion appropriate to the monoidal context.

---

[1]The fact that multiple Frobenius structures are on the same object (like in the aforementioned theories) or on different objects (like in this work) of a category may be overcome with the addition of 'switch' operations from one object to the others, as we are going to show in a paper in preparation.

Definition 2.1. A *monoidal theory* is a tuple (Σ, C) of a *signature* Σ and a finite set C of *colours*. Elements of Σ are *operations* $o \colon w \to v$ with a certain *arity* w and *coarity* v, where $w, v \in C^\star$.

Generic theories are typically triples, allowing also for a set of equations on Σ-terms. We do not need that level of generality here: equations will be treated differently, as rewriting rules (unless they are structural, like the equations of symmetric monoidal categories or of hypergraph categories, see below).

Towards hypergraph categories, it is instrumental to describe first the free *symmetric* monoidal category generated by a theory (Σ, C), which is called a C-coloured prop [18] (**pro**duct and **p**ermutation category). This works in analogy with the single-sorted case $C = \{c\}$, in which monoidal theories act as presentations for $(\{c\}$-coloured) props [23].

Definition 2.2. Let C be a finite set. A C-*coloured prop* is a symmetric monoidal category (SMC) where the set of objects is $C^\star$ and the monoidal product $\oplus$ on objects is word concatenation. C-coloured props form a category $\mathbf{Prop}_C$ with morphisms the identity-on-objects symmetric monoidal functors.

Given a monoidal theory (Σ, C), one can freely construct a prop $\mathbf{P}_{\Sigma,C}$ with arrows the Σ-terms quotiented by the laws of symmetric monoidal categories. Σ-terms are freely obtained by combining operations in Σ, a *unit id*$\colon c \to c$ for each $c \in C$ and a *symmetry* $\sigma_{c,d} \colon cd \to dc$ for each $c, d \in C$, by sequential (;) and parallel ($\oplus$) composition. That means, given terms $a \colon w_1 \to w_2$, $b \colon w_2 \to w_3$, $a' \colon v_1 \to v_2$, one constructs new terms $a \,;\, b \colon w_1 \to w_3$ and $a \oplus a' \colon w_1 v_1 \to w_2 v_2$.

We shall adopt the graphical notation of string diagrams [32] for the arrows of $\mathbf{P}_{\Sigma,C}$. An arrow $a \colon w_1 \to w_2$ is pictured as $\xrightarrow{w_1}\boxed{a}\xrightarrow{w_2}$. Compositions via ; and $\oplus$ are drawn respectively as horizontal and vertical juxtaposition, that means, $a \,;\, b$ is drawn $\xrightarrow{w_1}\boxed{a}\xrightarrow{w_2}\boxed{b}\xrightarrow{w_3}$ and $a \oplus a'$ is drawn $\begin{smallmatrix}\xrightarrow{w_1}\boxed{a}\xrightarrow{w_2}\\\xrightarrow{v_1}\boxed{a'}\xrightarrow{v_2}\end{smallmatrix}$. There are specific diagrams for the symmetric monoidal structure, namely $\xrightarrow{\quad w \quad}$ for the identity $id_w \colon w \to w$ and $\begin{smallmatrix}u & w\\ w & u\end{smallmatrix}$ for the symmetry $\sigma_{w,u} \colon wu \to uw$, for $w, u \in C^\star$. These are definable from the basic identities and symmetries for colours in C using the pasting rules for ; and $\oplus$.

Example 2.3. The initial object in $\mathbf{Prop}_C$ is the C-coloured prop $\mathbf{Perm}_C$ whose arrows $w \to v$ are permutations of w into v (thus arrows exist only when the word v is an anagram of the word w). $\mathbf{Perm}_C$ is freely generated by the monoidal theory $(\emptyset, C)$.

Example 2.4. For $C' \subseteq C$, the C-coloured prop $\mathbf{Frob}_{C'}$ of *separable Frobenius C'-monoids* is freely generated by the monoidal theory $(\Sigma_{\mathbf{Frob}}^{C'}, C)$, where $\Sigma_{\mathbf{Frob}}^{C'} = \{\ \text{\small diagrams}\ |\ c \in C'\}$, and quotiented by equations, for each $c \in C'$, stating that the diagrams form a commutative monoid (3), that the diagrams form a commutative comonoid (4), and that these interact according to the Frobenius law and the separability law (5).

$$\tag{3}$$

$$\tag{4}$$

$$\tag{5}$$

As mentioned, when C is a singleton, C-coloured props are just called props, i.e. SMCs with objects the natural numbers where the monoidal product is addition on objects. For later use it is convenient to record the following result about the single-sorted case. It involves the prop $\mathsf{Csp}(\mathbf{FinSet})$ whose arrows $n_1 \to n_2$ are cospans $\overline{n_1} \xrightarrow{f} \overline{n_3} \xleftarrow{g} \overline{n_2}$ of functions between ordinals $\overline{n} := \{0, \ldots, n-1\}$.

Proposition 2.5 ([6, 23]). *There is an isomorphism of {c}-coloured props between* **Frob**$_{\{c\}}$ *and* Csp(**FinSet**). *It is defined by the following mapping on the* $\Sigma_{\mathbf{Frob}}^{\{c\}}$*-operations.*

As outlined in the introduction, we are interested in studying non-structural equations as rewriting rules. We now define the appropriate notion of rewriting for arrows in a prop. We call it "syntactic" to emphasise that matching happens when the left-hand side of a rule is a sub-term.

Definition 2.6 (Syntactic rewriting in a prop)**.** A *rewriting rule* in a C-coloured prop **C** is a pair of morphisms $l, r \colon v_1 \to v_2$ in **C**, for which we use the notation $(l, r) \colon v_1 \to v_2$. A *rewriting system* $\mathcal{R}$ is a finite set of rewriting rules. Given $a, b \colon w_1 \to w_2$ in **C**, we say that a rewrites into b via $\mathcal{R}$, notation $a \Rightarrow_{\mathcal{R}} b$, if there are a_1 and a_2 yielding the following decompositions in **C**, where $(l, r) \colon v_1 \to v_2$ is in $\mathcal{R}$.

We are going to study syntactic rewriting in free hypergraph categories, which we now introduce together with their properties.

Definition 2.7. An *hypergraph category* is an SMC **A** where each object $x \in \mathbf{A}$ has a separable Frobenius structure, i.e., maps forming a commutative monoid, a commutative comonoid and satisfying equations as in (1) for each $x \in \mathbf{A}$. Moreover, the Frobenius structure must be compatible with the monoidal product:

The free hypergraph category over (Σ, C), notation $\mathbf{H}_{\Sigma,C}$, is the free C-coloured prop on $(\Sigma \uplus \Sigma_{\mathbf{Frob}}^C, C)$ quotiented by equations (3), (4) and (5) for each $c \in C$.

Observe that the free construction of $\mathbf{H}_{\Sigma,C}$ indeed creates a Frobenius structure for each object $w \in C^\star$ of the category, canonically defined in terms of the one on colours. For instance, for $w = c_1 \dots c_n$:

Example 2.8. Fix a set C of colours with just two elements, noted ● and ●, and a signature Σ consisting of two "colour switch" operations, ◆: ● → ● and ◆: ● → ●. We may construct the free hypergraph category $\mathbf{H}_{\Sigma,\{●,●\}}$ over $(\Sigma, \{●, ●\})$. Here is an example of a string diagram in this category, where we use the more suggestive convention of colouring wires instead of labelling them with objects ● and ●.

$$(7)$$

We claim that $\mathbf{H}_{\Sigma,\{●,●\}}$ is the same as the category of finite directed *bipartite graphs* (with interfaces). This will become clear in Example 3.5, after the characterisation provided by Corollary 3.3.

We now observe that the free hypergraph category $\mathbf{H}_{\Sigma,C}$ can be seen as a coproduct in $\mathbf{Prop}_C$. This will be useful in order to separate the component arising from Σ from the built-in Frobenius structure.

Proposition 2.9. *There is an isomorphism $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C \cong \mathbf{H}_{\Sigma,C}$ of C-coloured props. It extends to an isomorphism of hypergraph categories: thus $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ is the free hypergraph category on (Σ,C).*

Proof. The first part follows from how coproducts are computed in $\mathbf{Prop}_C$. As $\mathbf{P}_{\Sigma,C}$ is presented by (Σ,C) and $\mathbf{Frob}_C$ by $(\Sigma^C_{\mathbf{Frob}},C)$ quotiented by (3)-(5), then $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ is presented by $(\Sigma \uplus \Sigma^C_{\mathbf{Frob}},C)$ quotiented by (3)-(5): this is precisely the definition of $\mathbf{H}_{\Sigma,C}$. The second part holds because the isomorphism maps the Frobenius structure on $c \in C$ in $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ to the Frobenius structure on $c \in C$ in $\mathbf{H}_{\Sigma,C}$. $\square$

The free hypergraph category has a universal property (of a pushout) in $\mathbf{SymCat}$ too. Details are in the proof of Corollary 3.4, which uses this observation. We conclude by recalling that the Frobenius monoids in an hypergraph category defines a canonical compact closed structure. This also justifies the terminology "well-supported compact closed categories" originally used for hypergraph categories [9].

Proposition 2.10 ([9]). *Hypegraph categories are (self-dual) compact closed.*

Proof. It is useful to report how the compact closed structure is actually defined. For an object x of an hypergraph category $\mathbf{A}$, define $^x\!\subset$ as $\bullet\!\!\overset{x}{\bullet}\!\!\overset{x}{\underset{x}{\fbox{}}}$ and $\supset^x$ as $\overset{x}{\underset{x}{\fbox{}}}\!\!\bullet\!\overset{x}{\bullet}$. The Frobenius equation (5) implies the equation for compact closure:

$$\overset{x}{\underset{x}{\frown}}\!\!\overset{}{\underset{x}{\smile}}^x \;\;=\;\; \overset{x}{\underline{}}\;\;=\;\; \overset{x}{\underset{x}{\smile}}\!\!\overset{x}{\underset{x}{\frown}} \,.$$

The dual $\overset{x_2}{\underline{}}\fbox{a^*}\overset{x_1}{\underline{}}$ of a morphism $\overset{x_1}{\underline{}}\fbox{a}\overset{x_2}{\underline{}}$ is defined as $\overset{x_2}{\underset{x_1}{\frown}}\fbox{a}\overset{x_2}{\underset{x_1}{\smile}}$.

$\square$

3 The Combinatorial Interpretation

According to Definition 2.6, syntactic rewriting in free hypergraph categories happens modulo Frobenius structure. The goal of this section is to give a combinatorial description of the free hypergraph category, so that a more concrete account of the associated rewriting becomes available.

Fix a monoidal theory (Σ,C). We shall work with finite directed hypergraphs, whose hyperedges are labelled in Σ and nodes are labelled in C. We shall visualise hypergraphs as follows: $\copyright$ is a node labeled with $c \in C$ and $\vdots\!\!\mathrel{\succ}\!\!\fbox{O}\!\!\mathrel{\prec}\!\!\vdots$ is an hyperedge labeled with $o \in \Sigma$, with ordered tentacles attached to the left boundary linking to sources and the ones on the right linking to targets. An example is on the right, with $C = \{c_1, c_2\}$ and $\Sigma = \{o_1 : c_1 \to \epsilon, o_2 : c_1 c_2 \to c_1 c_1\}$.
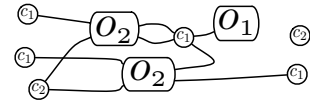
We now organise these structures into a category. First, consider the SMC $\mathbf{Hyp}$ with objects the finite directed (unlabelled) hypergraphs and homomorphisms between them. The monoidal theory (Σ,C) itself can be seen as an object of $\mathbf{Hyp}$. For instance, Σ and C as above yield the unlabelled hypergraph on the left (where we "call" o_2 the leftmost and o_1 the rightmost hyperedge, and c_2 the leftmost and c_1 the rightmost node). The desired labelling is given formally by working in the slice category $\mathbf{Hyp} \backslash (\Sigma,C)$, for which we shall use notation $\mathbf{Hyp}_{\Sigma,C}$. This definition ensures that a Σ-operation $o : w \to v$ labels an hyperedge only when the label of its input (respectively, output) nodes forms the word w (v).
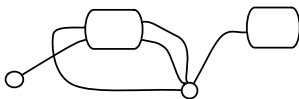
The next step is to add interfaces, formally modelled by cospans. Fix a set $\{x_i\}_{i\in\mathbb{N}}$ totally ordered by $\mathbb{N}$. Define $\mathbf{FTerm}_{\Sigma,C}$ as the restriction of the category of cospans in $\mathbf{Hyp}_{\Sigma,C}$ to objects the discrete hypergraphs (i.e. no hyperedges) whose set of nodes is isomorphic to an initial segment of $\{x_i\}_{i\in\mathbb{N}}$. This restriction ensures that $\mathbf{FTerm}_{\Sigma,C}$ is a C-coloured prop. Indeed, an object of $\mathbf{FTerm}_{\Sigma,C}$ can be identified with a natural number k together with a labelling function $\{0,\ldots,k-1\}\to C$, which is the same as a word in C^*. The notation $\mathbf{FTerm}_{\Sigma,C}$ stands for "Frobenius termgraphs", following the terminology introduced for the single-sorted case [2]. This name will be justified by the characterisation of Theorem 3.2 below.

We now define the two components of the functor that is going to interpret the syntactic definition of an hypergraph category as a combinatorial structure. The key to the approach is the second definition below, which essentially tells that the combinatorial model is able to absorb all the complexity of Frobenius structure simply in terms of nodes.

Definition 3.1. We define a C-coloured prop morphisms $\langle\!\langle \cdot \rangle\!\rangle\colon \colon \mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C \to \mathbf{FTerm}_{\Sigma,C}$ as the copairing of the following functors:

- $[\![\cdot]\!]\colon \mathbf{P}_{\Sigma,C} \to \mathbf{FTerm}_{\Sigma,C}$ is defined by the following mapping on operations $o\colon c_1 c_2 \ldots c_m \to b_1 b_2 \ldots b_m$ of Σ (where $c_i, b_i \in C$).
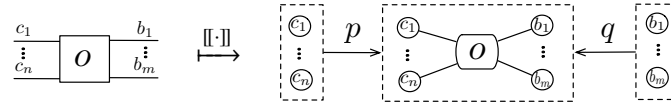
  The definition of hypergraph homomorphisms p and q is fixed by colour preservation.

- $[\cdot]\colon \mathbf{Frob}_C \to \mathbf{FTerm}_{\Sigma,C}$ is defined by the following mapping on the $\Sigma_{\mathbf{Frob}}^C$-operations.

(8)
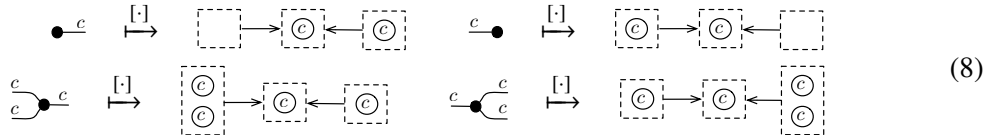
  Also here the definition of the hypergraph homomorphisms is predetermined.

We now have all the ingredients to state our characterisation theorem.

Theorem 3.2. $\langle\!\langle \cdot \rangle\!\rangle\colon \mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C \to \mathbf{FTerm}_{\Sigma,C}$ *is an isomorphism of C-coloured props.*

The proof of the theorem will be postponed to the end of the section. Let us first observe two interesting consequences.

Corollary 3.3. $\mathbf{FTerm}_{\Sigma,C}$ *is the free hypergraph category on (Σ, C), i.e. $\mathbf{H}_{\Sigma,C} \cong \mathbf{FTerm}_{\Sigma,C}$.*

Proof. $[\cdot]\colon \mathbf{Frob}_C \to \mathbf{FTerm}_{\Sigma,C}$ defines an hypergraph category structure on $\mathbf{FTerm}_{\Sigma,C}$ and the isomorphism of Theorem 3.2 extends to one of hypergraph categories. Then the result follows by Proposition 2.9 and Theorem 3.2. $\square$

The next corollary states that there is no 'information loss' in passing from the free symmetric monoidal category to the free hypergraph category on (Σ, C).

Corollary 3.4. $[\![\cdot]\!]\colon \mathbf{P}_{\Sigma,C} \to \mathbf{FTerm}_{\Sigma,C}$ *is faithful.*

*Proof.* We use that, just as for props [34, Prop. 2.8], coproducts of C-coloured props can be computed as certain pushouts in the category **SymCat** of small SMCs. In particular, $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ in $\mathbf{Prop}_C$ arises as

$$
\begin{array}{ccc}
\mathbf{Perm}_C & \xrightarrow{\quad !_2 \quad} & \mathbf{Frob}_C \\
{\scriptstyle !_1}\downarrow & & \downarrow{\scriptstyle [\cdot]} \\
\mathbf{P}_{\Sigma,C} & \xrightarrow[{[[\cdot]]}]{} & \mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C \cong \mathbf{FTerm}_{\Sigma,C}
\end{array}
\tag{9}
$$

in **SymCat**, where the maps $!_1$ and $!_2$ are given by initiality of $\mathbf{Perm}_C$ in $\mathbf{Prop}_C$ (see Example 2.3). Intuitively, in (9) $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ is built as the "disjoint union" of $\mathbf{P}_{\Sigma,C}$ and $\mathbf{Frob}_C$ where one identifies the set of objects $C^\star$ and the associated symmetric monoidal structure (the "contribution" of $\mathbf{Perm}_C$).

Now, in order to prove that $[[\cdot]]$ is faithful, we can use a result [25, Th. 3.3] about amalgamation in **Cat** (which transfers to **SymCat**). As all the functors in (9) are identity-on-objects and $!_1$, $!_2$ are faithful, it just requires to show that $!_1$ and $!_2$ satisfy the so-called 3-for-2 property: for $!_1$, this means that, given $h = f\,;g$ in $\mathbf{P}_{\Sigma,C}$, if any two of f, g, h are in the image of $!_1$, then so is the third. This trivially holds as every arrow of $\mathbf{Perm}_C$ is an isomorphism. The argument for $!_2$ is identical. $\qquad\square$

Example 3.5. We come back to the free hypergraph category $\mathbf{H}_{\Sigma,\{\bullet,\bullet\}}$ introduced in Example 3.5. By Corollary 3.3, $\mathbf{H}_{\Sigma,\{\bullet,\bullet\}} \cong \mathbf{FTerm}_{\Sigma,\{\bullet,\bullet\}}$. In hypergraphs of $\mathbf{FTerm}_{\Sigma,\{\bullet,\bullet\}}$, hyperedges correspond to switches ◀▶ or ◀▶, thus they are in fact edges (one input and one output node) and we may as well avoid drawing them in the graphical representation. Since they connect any two nodes only when these have a different colour, what we obtain are finite directed bipartite graphs. For instance, reprising (7):
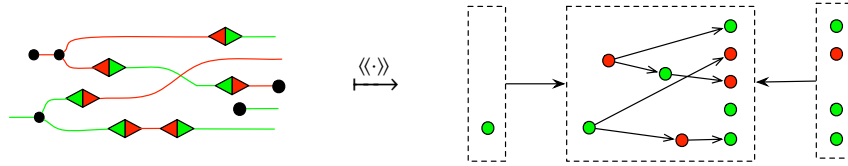
This example shows that Theorem 3.2 not only provides a combinatorial representation for algebraic structures, but conversely it is also instrumental in deriving an algebraic presentation for well-known graph-theoretic models.

We now give a proof of the characterisation theorem.

Proof of Theorem 3.2. As a preparatory step, we observe that $\mathbf{Frob}_C$ itself can be decomposed as a coproduct in $\mathbf{Prop}_C$, namely $\Sigma_{c\in C}\mathbf{Frob}_{\{c\}}$. Let us suppose for simplicity that $C = \{c, d\}$, so that $\mathbf{Frob}_C = \mathbf{Frob}_{\{c\}} + \mathbf{Frob}_{\{d\}}$. It will be apparent how the argument generalises.

The prop morphism $[\cdot]\colon \mathbf{Frob}_C \to \mathbf{FTerm}_{\Sigma,C}$ is analogously decomposable as the copairing $([\cdot]_c, [\cdot]_d)\colon \mathbf{Frob}_{\{c\}} + \mathbf{Frob}_{\{d\}} \to \mathbf{FTerm}_{\Sigma,C}$ of prop morphisms $[\cdot]_c\colon \mathbf{Frob}_{\{c\}} \to \mathbf{FTerm}_{\Sigma,C}$ and $[\cdot]_d\colon \mathbf{Frob}_{\{d\}} \to \mathbf{FTerm}_{\Sigma,C}$, defined by restricting the clauses (8) to the associated colour, either c or d.

We have thus reduced the statement to verify that $\mathbf{FTerm}_{\Sigma,C}$ satisfies the universal property of the coproduct $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_{\{c\}} + \mathbf{Frob}_{\{d\}}$ in $\mathbf{Prop}_C$.

Given α, β_c, β_d and a C-coloured prop **A** as in (10), we need to show the existence of a unique γ making the diagram commute. Now, because all morphisms in (10) are identity-on-objects, it suffices to show that any arrow of **FTerm**$_{\Sigma,C}$ can be decomposed in an essentially unique way into an expression where all the basic constituents lie in the image of $[[\cdot]]$, $[\cdot]_c$ or $[\cdot]_d$.

To this aim, fix a cospan $w \xrightarrow{f} G \xleftarrow{g} v$ in **FTerm**$_{\Sigma,C}$, where G has set of nodes N, set of hyperedges E and labelling functions $\psi \colon N \to C$ and $\chi \colon E \to \Sigma$. We pick an order $e_1, \ldots, e_j$ on the hyperedges in E and one $n_1, \ldots, n_k$ on the nodes in N. Let $\tilde{w} \xrightarrow{i} \tilde{E} \xleftarrow{o} \tilde{v}$ be the cospan defined as $\bigoplus_{1 \le i \le j} [[\chi(e_i)]]$. Intuitively, $\tilde{E}$ piles up all the hyperdges of G, but disconnected from each other. $\tilde{w}$ and $\tilde{m}$ are the word concatenations of all the inputs, respectively outputs of these hyperedges.

Similarly, we pile up all the (labeled) nodes in N, by forming the word $w_N \in C^\star$ as $\bigoplus_{1 \le i \le k} \psi(n_i)$. There are obvious functions from w, v, $\tilde{w}$ and $\tilde{v}$ to w_N, mapping labelled nodes to their occurrence in w_N. All this information is now gathered in the following composition of cospans [2]

$$(w \xrightarrow{f} w_N \xleftarrow{(id,j)} w_N \oplus \tilde{w}) \; ; \; (w_N \oplus \tilde{w} \xrightarrow{id \oplus i} w_N \oplus \tilde{E} \xleftarrow{id \oplus o} w_N \oplus \tilde{v}) \; ; \; (w_N \oplus \tilde{v} \xrightarrow{(id,p)} w_N \xleftarrow{g} v) \qquad (11)$$

Copairing maps (id, j) and (id, p) are well-defined as $\oplus$ is also a coproduct in **Hyp**$_{\Sigma,C}$. One can compute that the result of composing (11) (by pushout) is indeed isomorphic to $w \xrightarrow{f} g \xleftarrow{g} v$.

Towards a definition of γ, we need to check that every component of (11) is in the image of either $[[\cdot]]$ or $[\cdot] = ([\cdot]_c, [\cdot]_d)$. The middle cospan is clearly in the image of $[[\cdot]]$, as it is the monoidal product of the identity cospan $w_N \xrightarrow{id} w_N \xleftarrow{id} w_N$ with cospans in the image of some $o \in \Sigma$. Next, we want to check that the two outmost cospans are in the image of $([\cdot]_c, [\cdot]_d)$. To this aim, we show the following claim.

Claim. *Any arrow $u_1 \xrightarrow{h} u_3 \xleftarrow{q} u_2$ of* **FTerm**$_{\Sigma,C}$ *with u_1, u_2, u_3 discrete is in the image of $([\cdot]_c, [\cdot]_d)$.*

Proof. First, find permutations $\pi_1 \colon u_1 \to c^k d^z$, $\pi_2 \colon u_2 \to c^m d^n$ and $\pi_3 \colon u_3 \to c^l d^r$ factorising words u_1, u_2 and u_3 respectively as cs followed by ds. We can then define restrictions of h and of q to the c-segment or the d-segment of their domain: this gives functions $h_c \colon c^k \to c^l$, $h_d \colon d^z \to d^r$, $q_c \colon c^m \to c^l$ and $q_d \colon d^n \to d^r$. Observe that the codomain is restricted too, as h and q are colour-preserving maps. Putting these data together we can decompose $u_1 \xrightarrow{h} u_3 \xleftarrow{q} u_2$ as follows.

$$(u_1 \xrightarrow{id} w_1 \xleftarrow{\pi_1} c^k \oplus d^z) \; ; \; (c^k \oplus d^z \xrightarrow{h_c \oplus h_d} c^l \oplus d^r \xleftarrow{q_c \oplus q_d} c^m \oplus d^n) \; ; \; (c^m \oplus d^n \xrightarrow{\pi_2} u_2 \xleftarrow{id} u_2) \qquad (12)$$

It is now useful to recall Proposition 2.5. Observe that the bijection given therein between Frobenius structure and cospans in **FinSet** is defined by the same clauses (8) as $[\cdot] = ([\cdot]_c, [\cdot]_d)$, modulo the labelling of set elements all with c or with d. It follows that the cospan $c^k \xrightarrow{h_c} c^l \xleftarrow{q_c} c^m$ is in the image of $[\cdot]_c$ and the cospan $d^z \xrightarrow{h_d} d^r \xleftarrow{q_d} d^n$ is in the image of $[\cdot]_d$. Thus they are both in the image of $([\cdot]_c, [\cdot]_d)$. Concerning the two outermost cospans in (12), they are also in the image of $([\cdot]_c, [\cdot]_d)$, as this is a morphism of C-coloured prop and thus preserves and reflects the symmetry structure. Therefore, the whole of (12) is in the image of $([\cdot]_c, [\cdot]_d)$. $\square$

Back to the main proof, thanks to the claim we have shown that the two outmost cospans of (11) are in the image of $([\cdot]_c, [\cdot]_d)$. Therefore γ can be defined on $w \xrightarrow{f} G \xleftarrow{g} v$ by the values of $[[\cdot]]$ and $([\cdot]_c, [\cdot]_d)$

---

[2] We admit a certain degree of sloppiness in writing w_N both for an object of **FTerm**$_{\Sigma,C}$ and for the carrier of a cospan. For $w_N = \{n_1, \ldots, n_k\}$, these are isomorphic descriptions of the same data: in the first case it is treated as a word in $C^\star$, in the second as a set of nodes with a labelling function $\chi \colon \{n_1, \ldots, n_k\} \to C$.

on its decomposition as in (11). This is a correctly and uniquely defined assignment: in the construction of decompositions (11) and (12), the only variable parts are the different orderings that are picked for labelled nodes and for hyperedges in $\tilde{E}$, but these are immaterial since all the involved categories are symmetric monoidal. □

As a consequence of the claim enclosed in the proof of Theorem 3.2, it is worth noticing that the Frobenius structure identifies the hypergraphs with no hyperedges, i.e. the sets of C-labelled nodes.

Corollary 3.6. *There is an isomorphism of C-coloured props $\mathbf{Frob}_C \cong \mathbf{FTerm}_{\emptyset,C}$.*

As arrows of $\mathbf{FTerm}_{\emptyset,C}$ are the same thing as cospans in the slice category $\mathbf{FinSet} \setminus C$, this corollary can be seen as a multi-sorted analogue of the well-known result reported in Proposition 2.5.

Double-pushout rewriting with interfaces   We conclude this section by recalling double-pushout (DPO) rewriting [11], that we will use to compute in $\mathbf{FTerm}_{\Sigma,C}$. We will actually use a variation [17] of the standard definition: instead of just rewriting an hypergraph G, we shall rewrite an hypergraph homomorphism $G \leftarrow J$, standing for "G with interface J". The intuition is that this form of computation retains the information that J is how G "glues" to a bigger context. This is needed both to match the syntactic notion of rewriting (Definition 2.6) and for ensuring decidability of confluence for terminating system (Section 5). We formulate our definition at the level of *adhesive categories* [24]. This is the more abstract setting where DPO rewriting enjoys desirable properties (such as Church-Rosser) and where tools and algorithms for this form of rewriting are generally defined.

Definition 3.7 (DPOI rewriting)**.**   Fix an adhesive category $\mathbf{A}$. A *rule for double-pushout rewriting with interfaces (DPOI rule)* is a span $L \leftarrow J \rightarrow R$ in $\mathbf{A}$. A *DPOI rewriting system* $\mathcal{R}$ is a finite set of DPOI rules. Given morphisms $G \xleftarrow{r} I$ and $H \xleftarrow{p} I$ in $\mathbf{A}$, we say that G rewrites into H via $\mathcal{R}$ with interface I, notation $(G \xleftarrow{r} I) \rightsquigarrow_{\mathcal{R}} (H \xleftarrow{p} I)$, if there exists a DPOI rule $L \leftarrow J \rightarrow R$ in $\mathcal{R}$ and a cospan $J \rightarrow C \xleftarrow{q} I$ such that the following diagram commutes and the two squares are pushouts. We call $L \rightarrow G$ a *match* of L in G.

$$
\begin{array}{ccc}
L & \longleftarrow J \longrightarrow & R \\
\downarrow \;\ulcorner & \downarrow & \ulcorner\; \downarrow \\
G & \longleftarrow C \longrightarrow & H \\
& \underset{r}{\searrow} \; q\uparrow \; \underset{p}{\swarrow} & \\
& I &
\end{array}
\tag{13}
$$

If $\mathbf{A}$ has an initial object 0, one can relax rewriting to act on cospans of the form $0 \rightarrow G \xleftarrow{r} I$, seen as objects with interface $(G \xrightarrow{r} I)$. In this case we write $(0 \rightarrow G \xleftarrow{r} I) \rightsquigarrow_{\mathcal{R}} (0 \rightarrow H \xleftarrow{p} I)$ for a rewriting step.

The following makes DPOI rewriting possible in $\mathbf{A} = \mathbf{Hyp}_{\Sigma,C}$, see Ex. 4.2 below for an illustration.

Proposition 3.8. $\mathbf{Hyp}_{\Sigma,C}$ *is an adhesive category.*

Proof. The category $\mathbf{Hyp}$ of finite directed (unlabeled) hypergraphs is a presheaf category and thus adhesive [24, 2]. The statement then follows because $\mathbf{Hyp}_{\Sigma,C}$ is defined as the slice category $\mathbf{Hyp} \setminus (\Sigma, C)$ and adhesive categories are closed under slice [24]. □

4   DPOI Implementation of Rewriting Modulo Frobenius

The category $\mathbf{Hyp}_{\Sigma,C}$ has an initial object 0: the hypergraph with neither nodes nor hyperedges. Therefore, as mentioned in Definition 3.7, we can equivalently think of hypergraphs with interface $(G \rightarrow I)$ as cospans $0 \rightarrow G \leftarrow I$, and meaningfully define DPOI rewriting on the morphisms with source 0 in $\mathbf{FTerm}_{\Sigma,C}$.

On the other hand, our semantics $\langle\!\langle\cdot\rangle\!\rangle$ maps diagrams of $\mathbf{P}_\Sigma + \mathbf{Frob}_C$ to cospans with any source. Thus, in order to interpret syntactic rewriting, we need an intermediate step where we "fold" the two interfaces w_1, w_2 of a string diagram $a\colon w_1 \to w_2$ into one $w_1 w_2$. This is formally defined, with the help of the compact closed structure on $\mathbf{P}_\Sigma + \mathbf{Frob}_C$ (Proposition 2.10), by an operation $\ulcorner\cdot\urcorner$:

We are now ready to formulate the correspondence theorem between syntactic rewriting in $\mathbf{P}_\Sigma + \mathbf{Frob}_C$ and DPOI rewriting in $\mathbf{FTerm}_\Sigma$.

Theorem 4.1. *Let (l, r) be any rewriting rule on $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$. Then,*

$$a \Rightarrow_{(l,r)} b \quad\text{ iff }\quad \langle\!\langle\ulcorner a\urcorner\rangle\!\rangle \leadsto_{\langle\!\langle\langle\ulcorner l\urcorner,\ulcorner r\urcorner\rangle\rangle\!\rangle\rangle} \langle\!\langle\ulcorner b\urcorner\rangle\!\rangle\,.$$

Proof. On the direction from left to right, suppose that $a \Rightarrow_{(l,r)} b$. Thus, by definition,

                                                                                          (14)

Using the compact closed structure of $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ we can put $\ulcorner a\urcorner$ in the following shape

The dashed line decomposes the rightmost diagram into $\ulcorner l\urcorner\colon 0 \to v_1 v_2$ followed by a diagram of type $v_1 v_2 \to w_1 w_2$, which we name $\tilde{a}$. With analogous reasoning,

            meaning that $\ulcorner a\urcorner = \ulcorner l\urcorner\,;\tilde{a}$ and $\ulcorner b\urcorner = \ulcorner r\urcorner\,;\tilde{a}.$ (15)

Next, we introduce cospans giving semantics to the various diagrams:

$$\langle\!\langle\ulcorner l\urcorner\rangle\!\rangle = 0 \to L \leftarrow v_1 v_2 \qquad \langle\!\langle\tilde{a}\rangle\!\rangle = v_1 v_2 \to C \leftarrow w_1 w_2 \qquad \langle\!\langle\ulcorner r\urcorner\rangle\!\rangle = 0 \to R \leftarrow v_1 v_2$$
$$\langle\!\langle\ulcorner a\urcorner\rangle\!\rangle = 0 \to G \leftarrow w_1 w_2 \qquad \langle\!\langle\ulcorner b\urcorner\rangle\!\rangle = 0 \to H \leftarrow w_1 w_2.$$
(16)

Equation (15) tells that the cospan giving semantics to $\ulcorner a\urcorner$ (respectively, $\ulcorner b\urcorner$) is the composite of cospans giving semantics to $\ulcorner l\urcorner$ (respectively, $\ulcorner r\urcorner$) and $\tilde{a}$. As composition of cospans is by pushout, we obtain a double-pushout diagram as in (13) with $J = v_1 v_2$ and $I = w_1 w_2$, meaning that $\langle\!\langle\ulcorner a\urcorner\rangle\!\rangle \leadsto_{\langle\!\langle\langle\ulcorner l\urcorner,\ulcorner r\urcorner\rangle\rangle\!\rangle\rangle} \langle\!\langle\ulcorner b\urcorner\rangle\!\rangle$. We now conclude the proof by showing the right to left direction of the statement. Suppose that $\langle\!\langle\ulcorner a\urcorner\rangle\!\rangle \leadsto_{\langle\!\langle\langle\ulcorner l\urcorner,\ulcorner r\urcorner\rangle\rangle\!\rangle\rangle} \langle\!\langle\ulcorner b\urcorner\rangle\!\rangle$. Naming cospans $\langle\!\langle\ulcorner a\urcorner\rangle\!\rangle$, $\langle\!\langle\ulcorner b\urcorner\rangle\!\rangle$, $\langle\!\langle\ulcorner l\urcorner\rangle\!\rangle$ and $\langle\!\langle\ulcorner r\urcorner\rangle\!\rangle$ as in (16), this implies by definition the existence of a pushout complement C yielding a DPOI diagram as (13) with $J = v_1 v_2$ and $I = w_1 w_2$. Now, pick $\hat{a}\colon v_1 v_2 \to w_1 w_2$ such that $\langle\!\langle\hat{a}\rangle\!\rangle = v_1 v_2 \to C \leftarrow w_1 w_2$, which exists by fullness of $\langle\!\langle\cdot\rangle\!\rangle$. Because composition in $\mathbf{FTerm}_{\Sigma,C}$ is by pushout, the existence of such a DPOI diagram yields

$$\langle\!\langle\ulcorner a\urcorner\rangle\!\rangle = (0 \to G \leftarrow w_1 w_2) = (0 \to L \leftarrow v_1 v_2)\,; (v_1 v_2 \to C \leftarrow w_1 w_2) = \langle\!\langle\ulcorner l\urcorner\rangle\!\rangle\,; \langle\!\langle\hat{a}\rangle\!\rangle$$
$$\langle\!\langle\ulcorner b\urcorner\rangle\!\rangle = (0 \to H \leftarrow w_1 w_2) = (0 \to R \leftarrow v_1 v_2)\,; (v_1 v_2 \to C \leftarrow w_1 w_2) = \langle\!\langle\ulcorner r\urcorner\rangle\!\rangle\,; \langle\!\langle\hat{a}\rangle\!\rangle.$$
(17)

Because $\langle\langle\cdot\rangle\rangle$ is faithful, (17) yields decompositions $\ulcorner a \urcorner = \ulcorner l \urcorner ; \hat{a}$ and $\ulcorner b \urcorner = \ulcorner r \urcorner ; \hat{a}$ also on the syntactic side. This allows for a rewriting step $a \Rightarrow_{(l,r)} b$ as below, where the dashed lines show how the syntactic matching (*cf.* the shape (6)) is performed.

$\square$

Example 4.2. The syntactic rewriting step (2) takes place in $\mathbf{P}_{\Sigma,\{c,d\}} + \mathbf{Frob}_{\{c,d\}}$ where $\Sigma = \{o_1 \colon c \to cd, o_2 \colon dd \to c\}$. It maps via Theorem 4.1 to the following DPOI rewriting step in $\mathbf{FTerm}_{\Sigma,\{c,d\}}$. We use numbers $1, 2, 3$ to disambiguate the "folding" of the rule interfaces.
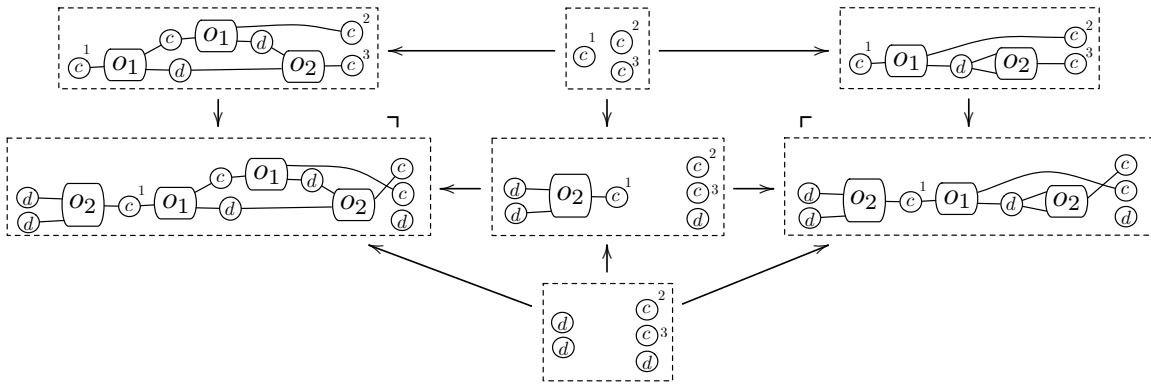
5 Decidability of Confluence

This section verifies that the form of rewriting crystallised by Theorem 4.1 enjoys the *Knuth-Bendix property*. Echoing the case of term rewriting, we use this terminology to mean that the confluence problem is reducible to critical pair analysis and both are decidable for terminating rewriting systems.

To this aim, we instantiate to our setting the results of [3]. There the author and collaborators showed that DPOI rewriting enjoys the aforementioned Knuth-Bendix property. Interfaces play a crucial role here, as Plump showed that for DPO rewriting (without interfaces) confluence is undecidable [29].

Definition 5.1 (DPOI Critical Pair). Fix an adhesive category and DPOI rules $L_1 \leftarrow K_1 \to R_1$ and $L_2 \leftarrow K_2 \to R_2$. Consider the following two derivations with common source S.

(18)

We say that $(H_1 \leftarrow J) \leftsquigarrow (S \leftarrow J) \rightsquigarrow (H_2 \leftarrow J)$ is a *critical pair* if (i) $[f_1, f_2] \colon L_1 + L_2 \to S$ is epi and (ii) (†) is a pullback. It is *joinable* if there exists $(W \leftarrow J)$ such that $(H_1 \leftarrow J) \rightsquigarrow^\star (W \leftarrow J) \, {}^\star\!\!\leftsquigarrow (H_2 \leftarrow J)$.

Intuitively, condition (i) ensures that a critical pair S is not bigger than $L_1 + L_2$ and condition (ii) says that J is the largest interface that allows both rewriting steps (J is the "intersection" of C_1 and C_2).

Proposition 5.2 ([3]). *Suppose that* **C** *satisfies the following assumptions: (1) it has an epi-mono factorisation system; (2) it has binary coproducts, pushouts and pullbacks; (3) it is adhesive (4) with all the pushouts stable under pullbacks. Then DPOI rewriting in* **C** *has the Knuth-Bendix property for computable rewriting systems.*

Remark 5.3. In the statement of Proposition 5.2, computability refers to the conditions that (i) pullbacks are computable, (ii) for every pair of DPOI rules $L_1 \leftarrow K_1 \rightarrow R_1$ and $L_2 \leftarrow K_2 \rightarrow R_2$ the set of quotients of $L_1 + L_2$ is finite and computable, and (iii) for all $G \leftarrow J$ one can compute every $H \leftarrow I$ such that $(G \leftarrow I) \rightsquigarrow (H \leftarrow I)$. In particular, (i)-(ii) ensure that the set of critical pairs is finite and computable, and (iii) ensures that any rewriting step is also computable— see [3] for the full technical details. Caveats on computability are intended to single out those structures where it is reasonable to apply the DPOI mechanism, as opposed to e.g. systems with infinitely many critical pairs.

Remark 5.4. The one of Definition 5.1 is sometimes called a "pre-critical" pair, as no condition ensures that L_1 and L_2 actually overlap in S. This distinction can be formulated abstractly for DPO in adhesive categories when rules are (left- or right-) linear, see [12] for an overview. However, Proposition 5.2 works with arbitrary (non-linear) rules, *cf.* [3], whence the more general definition. Non-overlapping pairs can be singled out in our category of interest, **FTerm**$_{\Sigma,C}$, and discarded for confluence testing as they are always joinable— also, their set is finite, because any DPOI systems is computable in **FTerm**$_{\Sigma,C}$.

Theorem 5.5. *DPOI rewriting in* **FTerm**$_{\Sigma,C}$ *has the Knuth-Bendix property.*

Proof. Given that DPOI rewriting in **FTerm**$_{\Sigma,C}$ is defined in terms of DPOI rewriting in **Hyp**$_{\Sigma,C}$, it suffices to check the statement in **Hyp**$_{\Sigma,C}$. We use Proposition 5.2: assumptions (1)-(4) hold in any presheaf category and are closed under slice. Therefore, as **Hyp**$_{\Sigma,C}$ is defined as the slice of a presheaf category (*cf.* proof of Proposition 3.8), it satisfies (1)-(4). It remains to check that any DPOI rewriting system in **Hyp**$_{\Sigma,C}$ is computable: the relevant observations are that in **Hyp**$_{\Sigma,C}$ (i) pushouts and pushout complements are effectively computable and (ii) for any two rules there are only finitely many hypergraphs that may witness a critical pair as defined in Definition 5.1. $\square$

We would now wish to transfer Theorem 5.5 to syntactic rewriting in $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$. This requires some extra care. In order to determine if a rewriting system $\mathcal{R}$ on $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ is confluent, by Theorem 5.5 and 4.1 it is enough that all the critical pairs in the DPOI system $\langle\!\langle \ulcorner \mathcal{R} \urcorner \rangle\!\rangle$ are joinable. However, for full decidability we also need to make sure that the converse holds: if one critical pair in $\langle\!\langle \ulcorner \mathcal{R} \urcorner \rangle\!\rangle$ is not joinable, then $\mathcal{R}$ should not be confluent. To ensure this, we need to verify that all the critical pairs of $\langle\!\langle \ulcorner \mathcal{R} \urcorner \rangle\!\rangle$ lay in the image of $\langle\!\langle \ulcorner \cdot \urcorner \rangle\!\rangle$. This amounts to check that they all have discrete interfaces.

Lemma 5.6. *Consider a critical pair in* **Hyp**$_{\Sigma,C}$ *as in* (18). *If both K_1 and K_2 are discrete hypergraphs, so is the interface J.*

Proof. For $i = 1, 2$, since K_i is discrete, the hyperedges of C_i are exactly those of G_i that are not in $f_i(L_i)$. Since $[f_1, f_2]: L_1 + L_2 \rightarrow S$ is epi, all the hyperedges of G are either in $f_1(L_1)$ or $f_2(L_2)$. Therefore, J cannot contain any hyperedge. $\square$

By definition of $\langle\!\langle \cdot \rangle\!\rangle$, for every rule $L \leftarrow K \rightarrow R$ in $\langle\!\langle \ulcorner \mathcal{R} \urcorner \rangle\!\rangle$, K is discrete. Therefore we can finally transfer the decidability result to the context of $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$.

Corollary 5.7. *Syntactic rewriting in* $\mathbf{P}_{\Sigma,C} + \mathbf{Frob}_C$ *has the Knuth-Bendix property.*

Proof. Since DPOI rewriting in **FTerm**$_{\Sigma,C}$ has the Knuth-Bendix property (Theorem 5.5) and the two forms of rewriting coincide (Theorem 4.1), as discussed above it suffices to check that for a given critical pair in **FTerm**$_{\Sigma,C}$, say witnessed by $(S \leftarrow J)$, there exists a in $\mathbf{P}_\Sigma + \mathbf{Frob}_C$, such that $\langle\!\langle a \rangle\!\rangle = 0 \rightarrow S \leftarrow J$. As $\langle\!\langle \cdot \rangle\!\rangle$ is full on **FTerm**$_{\Sigma,C}$, it suffices that J is discrete, which is true by Lemma 5.6. $\square$

6 Conclusions

We described a sound and complete interpretation of string diagram rewriting in hypergraph categories as double-pushout rewriting of hypergraphs, and showed that it enjoys decidability of confluence for terminating rewriting systems. A chief advantage of this approach is that the challenge posed by performing matching modulo Frobenius equations disappears in the combinatorial model. This becomes important when studying axiomatisations with multiple Frobenius monoids: these can now be all seen as structural equations and baked into the combinatorial model, thus confining questions of confluence and termination to the non-Frobenius axioms. This application of our theory, which we plan to explore in future work, was the main reason to generalise the framework of [2, 3], which is only able to absorb a single Frobenius structure. Another promising direction is the algebraic study of bipartite graphs (Example 3.5), which may be relevant for analysing diagrammatic languages, like biological metabolic networks [33], based on these structures.

References

[1] John Baez & Jason Erbele (2015): *Categories In Control. Theory and Application of Categories* 30, pp. 836–881.

[2] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobociński & Fabio Zanasi (2016): *Rewriting modulo symmetric monoidal structure*. In: *LiCS 2016*, pp. 710–719, doi:10.1145/2933575.2935316.

[3] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobociński & Fabio Zanasi (2017): *Confluence of graph rewriting with interfaces*. In: *ESOP 2016*, pp. 141–169, doi:10.1007/978-3-662-54434-1.

[4] Filippo Bonchi, Pawel Sobocinski & Fabio Zanasi (2014): *A Categorical Semantics of Signal Flow Graphs*. In: *CONCUR 2014*, *LNCS* 8704, Springer, pp. 435–450, doi:10.1007/978-3-662-44584-6.

[5] Filippo Bonchi, Pawel Sobocinski & Fabio Zanasi (2017): *The Calculus of Signal Flow Diagrams I: Linear relations on streams*. *Inf. Comput.* 252, pp. 2–29, doi:10.1016/j.ic.2016.03.002.

[6] Roberto Bruni & Fabio Gadducci (2001): *Some algebraic laws for spans*. *ENTCS* 44, pp. 175–193, doi:10.1016/S1571-0661(04)80937-X.

[7] Roberto Bruni, Ivan Lanese & Ugo Montanari (2006): *A basic algebra of stateless connectors*. *Theoretical Computer Science* 366(1–2), pp. 98–120, doi:10.1016/j.tcs.2006.07.005.

[8] Albert Burroni (1993): *Higher dimensional word problems with applications to equational logic*. *Theoretical Computer Science* 115(1), pp. 43–62, doi:10.1016/0304-3975(93)90054.

[9] Aurelio Carboni & R. F. C. Walters (1987): *Cartesian Bicategories I. Journal of Pure and Applied Algebra* 49(1-2), pp. 11–32, doi:10.1016/0022-4049(87)90121.

[10] Bob Coecke & Ross Duncan (2008): *Interacting Quantum Observables*. In: *ICALP 2008*, *LNCS* 5216, Springer, pp. 298–310, doi:10.1007/978-3-540-70583-3.

[11] A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel & M. Loewe (1997): *Algebraic Approaches to Graph Transformation, Part I: Basic Concepts and Double Pushout Approach*. In: *Handbook of Graph Grammars*, University of Pisa, pp. 163–246.

[12] Andrea Corradini (2016): *On the definition of parallel independence in the algebraic approaches to graph transformation*. In: *STAF 2016*, *LNCS* 9946, Springer, doi:10.1007/978-3-319-50230-4.

[13] Hartmut Ehrig, Annegret Habel, Julia Padberg & Ulrike Prange (2004): *Adhesive High-Level Replacement Categories and Systems*. In: *ICGT 2004*, *LNCS* 2987, Springer, pp. 144–160, doi:10.1007/978-3-540-30203-2.

[14] Hartmut Ehrig & Barbara König (2004): *Deriving Bisimulation Congruences in the DPO Approach to Graph Rewriting*. In: *FoSSaCS 2004*, *LNCS* 2987, Springer, pp. 151–166, doi:10.1007/978-3-540-24727-2.

[15] Brendan Fong (2016): *The Algebra of Open and Interconnected Systems*. Ph.D. thesis, University of Oxford.

[16] Brendan Fong & Fabio Zanasi (2017): *A Universal construction for (co)relations*. In: *Proceedings of CALCO'17*.

[17] Fabio Gadducci & Reiko Heckel (1997): *An inductive view of graph transformation*. In: *WADT 1997*, *LNCS* 1376, Springer, pp. 223–237, doi:10.1007/3-540-64299-4.

[18] Philip Hackney & Marcy Robertson (2015): *On the Category of Props*. Applied Categorical Structures 23(4), pp. 543–573, doi:10.1007/s10485-014-9369-4.

[19] Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, Stephen Pulman & Bob Coecke (2014): *Reasoning about Meaning in Natural Language with Compact Closed Categories and Frobenius Algebras*. CoRR abs/1401.5980. Available at http://arxiv.org/abs/1401.5980.

[20] Piergiulio Katis, Nicoletta Sabadini & Robert Frank Carslaw Walters (1997): *Span(Graph): a categorical algebra of transition systems*. In: *Proceedings of AMAST '97*, *LNCS* 1349, Springer, pp. 322–336, doi:10.1007/BFb0000479.

[21] G. M. Kelly & Ross Street (1974): *Review of the elements of 2-categories*. In: *Category Seminar (Proc. Sem., Sydney, 1972/1973)*, Springer, pp. 75–103. Lecture Notes in Math., Vol. 420, doi:10.1016/0022-4049(72)90019-9.

[22] Aleks Kissinger (2014): *Finite matrices are complete for (dagger-)hypergraph categories*. CoRR abs/1406.5942. Available at http://arxiv.org/abs/1406.5942.

[23] Stephen Lack (2004): *Composing PROPs*. Theory and Application of Categories 13(9), pp. 147–163.

[24] Stephen Lack & Paweł Sobociński (2005): *Adhesive and quasiadhesive categories*. Theoretical Informatics and Applications 39(3), pp. 511–546, doi:10.1051/ita:2005028.

[25] John MacDonald & Laura Scull (2009): *Amalgamations of categories*. Can Math B 52, pp. 273–284, doi:10.4153/CMB-2009-030-5.

[26] Dan Marsden & Fabrizio Genovese (2017): *Custom hypergraph categories via generalized relations*. arXiv abs/1703.01204. Available at http://arxiv.org/abs/1703.01204.

[27] Samuel Mimram (2010): *Computing Critical Pairs in 2-Dimensional Rewriting Systems*. In: *RTA 2010*, *LIPIcs* 6, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 227–242, doi:10.4230/LIPIcs.RTA.2010.227.

[28] Dusko Pavlovic (2013): *Monoidal computer I: Basic computability by string diagrams*. Information and Computation 226, pp. 94–116, doi:10.1016/j.ic.2013.03.007.

[29] Detlef Plump (1993): *Hypergraph Rewriting: Critical Pairs and Undecidability of Confluence*. In: *Term Graph Rewriting: Theory and Practice*, Wiley, pp. 201–213.

[30] Detlef Plump (2010): *Checking Graph-Transformation Systems for Confluence*. In: *Manipulation of Graphs, Algebras and Pictures*, *ECEASST* 26, EASST.

[31] Robert Rosebrugh, Nicoletta Sabadini & R. F. C. Walters (2005): *Generic Commutative Separable Algebras and Cospans of Graphs*. Theory and Application of Categories 17(6), pp. 164–177.

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

[33] Veeramani, Balaji, & Joel S Bader (2010): *Predicting Functional Associations from Metabolism Using Bi-Partite Network Algorithms*. BMC Systems Biology 4, doi:10.1186/1752-0509-4-95.

[34] Fabio Zanasi (2015): *Interacting Hopf Algebras: the theory of linear systems*. Ph.D. thesis, Ecole Normale Supérieure de Lyon.

[35] Fabio Zanasi (2016): *The Algebra of Partial Equivalence Relations*. In: *Mathematical Foundations of Program Semantics (MFPS)*, 325, pp. 313–333, doi:10.1016/j.entcs.2016.09.046.