Partially monoidal categories and the algebra of simultaneous substitutions

Samuel Balco Alexander Kurz

August 13, 2018

Abstract

The category of finite subsets of a countably infinite set of variables can be seen as the categorical semantics for the algebra of simultaneous substitutions of variables for variables. This category has a partially monoidal operation given by union of disjoint sets. We develop a calculus of string diagrams that represents this partially monoidal category up to isomorphism.

Contents

1 Introduction

We are interested in categories with partially monoidal structure. While we expect that this will have applications in various situations where resources can be only combined subject to certain constraints, the example we focus on in this paper is the 2-dimensional algebra of simultaneous substitutions.

Substitutions $[a \mapsto b]$ and $[b \mapsto c]$ can be composed sequentially, and we will have

$$[a \mapsto b] \,;\, [b \mapsto c] = [a \mapsto c].$$

Additionally, we want to build simultaneous substitutions

$$[a{\mapsto}b] \oplus [c{\mapsto}d] = [a{\mapsto}b, c{\mapsto}d].$$

We call $\oplus$ the tensor, or the monoidal or vertical or parallel composition. Semantically, this simultaneous substitution will correspond to the function

$$f : \{a, c\} \to \{b, d\}$$

satisfying $f(a) = b$ and $f(c) = d$ as suggested by the syntax $[a{\mapsto}b, c{\mapsto}d]$. Parallel composition of simultaneous substitutions is partial. For example,

$$[a{\mapsto}b] \oplus [a{\mapsto}c]$$

is undefined, since there is no function $\{a\} \to \{b, c\}$ that maps a simultaneously to both b and c.

What is the advantage of the 2-dimensional calculus over a 1-dimensional one? As we are setting out to represent the category of finite sets and functions, in a 1-dimensional calculus, operations $[a{\mapsto}b]$ will have to be indexed by a finite set S and will be of type $[a{\mapsto}b]_S : S]\uplus\{a\} \to S\uplus\{b\}$ for sets S with $a, b \notin S$. The first observation is that the ability of the 2-dimensional calculus to represent set-union by $\oplus$ makes indexing with subsets S unnecessary. More importantly, consider how to implement the swapping

$$[a{\mapsto}b] \oplus [b{\mapsto}a] : \{a, b\} \to \{a, b\} \tag{1}$$

in the 1-dimensional calculus via

$$\{a, b\} \longrightarrow \{c, b\} \longrightarrow \{c, a\} \longrightarrow \{a, b\}$$
$$[a{\mapsto}c]_{\{b\}} \,;\, [b{\mapsto}a]_{\{c\}} \,;\, [c{\mapsto}a]_{\{b\}}$$

While it is possible to write down the equations and rewrite rules for the 1-dimensional calculus, it does not appear as particularly natural. In particular, only in the 2-dimensional calculus, will the substitution $[a{\mapsto}c]_{\{b\}} \,;\, [b{\mapsto}a]_{\{c\}} \,;\, [c{\mapsto}a]_{\{b\}}$ have a simple normal form such as $[a{\mapsto}b] \oplus [b{\mapsto}a]$, which is unique up to commutativity of $\oplus$.

The two dimensional nature of simultaneous substitutions with sequential and parallel composition suggests to develop a calculus of string diagrams, see e.g. Selinger [11] for a survey of graphical languages.

The calculus we propose is given in Figure 2 and we prove it sound and complete w.r.t. the category of finite sets and functions. More precisely, we fix a set Σ of *names* $a, b, c, \ldots$ which is infinite and may be assumed to be countable. We also call Σ a set of *variables* to line up with the familiar terminology used for substitutions. The semantic category we are interested in then is the category $\mathfrak{F}$ which has finite subsets of Σ as objects and all functions as arrows and disjoint union as partial monoidal composition.

The soundness and completeness of Figure 2 can also be stated as follows: The category presented by Figure 2 is isomorphic to $\mathfrak{F}$.

It is worth emphasising that the partiality of the monoidal composition is needed because we want to represent $\mathfrak{F}$ up to isomorphism. Indeed, the skeleton of $\mathfrak{F}$, the category of natural numbers with all functions and addition as (total) monoidal operation, is equivalent to $\mathfrak{F}$. But from the point of view of syntax, these two categories are quite different: While we don't deal with variable binding in this paper, the skeleton of $\mathfrak{F}$ corresponds to using de

Bruijn indices in a λ-calculus whereas $\mathfrak{F}$ itself corresponds to using the more conventional variables.

To summarise the content, Section 2 introduces partially monoidal categories, Section 3 defines the syntax and semantics of our language of string diagrams and Section 4 and Section 5 show completeness of the axiomatisations of, respectively, bijections and functions. Section 6 gives a short account of the software we developed to support the mathematical reasoning of the paper.

Acknowledgments We gratefully acknowledge discussions with Roy Crole, Fredrik Dahlqvist, Samuel Mimram, Drew Moshier, David Pym, Tom Ridge, Pawel Sobocinski, Georg Struth, and Fabio Zanasi. Many of the diagrams in this paper were taken and adapted from Lafont [7].

2 Partially monoidal categories

Partial monoids play a role in many different areas of mathematics and computer science. One typical reason for partiality is the one also appearing in resource sensitive logics such as separation logic: If $f : H \to \mathbb{N}$ and $f' : H' \to \mathbb{N}$ are two partial functions from pieces H, H' of the memory, then they can be added if H and H' are disjoint.

In the literature there are slightly different notions of partial monoid, depending on the role of the neutral element. One can have no neutral element, one neutral element, or many neutral elements (one for each). For categories as partial monoids see Mac Lane [8, Ch.XII.5].

We write $\doteq$ to say that both sides are equal if either side is defined (hence, one side is defined if and only if the other side is).

Definition 2.1. A partial semigroup $(A, \otimes, D)$ consists a binary operation $\otimes$ defined on $D \subseteq A \times A$ such that for all $a, b, c \in A$ the following

$$(a \otimes b) \otimes c \doteq a \otimes (b \otimes c)$$

A partial monoid $(A, e, \otimes, D)$ has, moreover, a constant e for which

$$e \otimes a = a$$
$$a \otimes e = a$$

These structures are called commutative if $a \otimes b \doteq b \otimes a$.

As explained in the introduction, we are interested in partially monoidal categories. As our examples in this paper are strict, we can give the following simplified definition.

Definition 2.2. A (strict) partially monoidal category, or p-monoidal category, consists of

- a category $A = (A_0, A_1)$ with sets A_0 of objects and A_1 of arrows and
- partial monoids $(A_0, e, \otimes, D_0)$ and $(A_1, \mathrm{id}_e, \otimes, D_1)$
- such that $D = (D_0, D_1)$ is a subcategory of $A \times A$
- and $\otimes$ is a functor $D \to A$.

The category is called commutative p-monoidal if the two partial monoids are commutative. A strict monoidal functor is a functor F such that $F(e) = e$ and $F(a \otimes a') = F(a) \otimes F(a')$ whenever $a \otimes a'$ is defined.

Remark 2.3. In the examples of this paper, the third bullet point could be strengthened to say that D is a full subcategory, that is, two arrows can be composed by $\otimes$ whenever their domains and codomains can be composed. The fourth bullet point entails the interchange law

$$(f_1 \otimes f_2)\#(g_1 \otimes g_2) \doteq (f_1\#g_1) \otimes (f_2\#g_2) \tag{2}$$

whenever $(f_1, f_2) \in D$ and $(g_1, g_2) \in D$. We write $\doteq$ to emphasise that this interchange law is weaker than the one for 2-categories, which holds whenever one of the two sides is defined, see Mac Lane [8, Ch.XII.5]. Here, in the partially monoidal situation, the right-hand side may be defined without the left-hand side being defined. In particular, it will not always be possible to 'slice up' a string diagram in the familiar fashion, see the slashed red line in (5) or (4) for examples.

Similarly to Mac Lane [8, Ch.XII.5], we also give a one-sorted formulation of partially monoidal categories.

Proposition 2.4. The data of a partially monoidal category can also be described as a category $(C, s, t, \#)$ in the sense of (1-4) of [8, Ch.XII.5, p.297] equipped with a partial monoid $(C, \varepsilon, \otimes, D)$ where D restricts to a subcategory of $(C, s, t, \#)$ satisfying, moreover, whenever the left-hand sides are defined, the equations

$$s(c \otimes c') \doteq s(c) \otimes s(c')$$
$$t(c \otimes c') \doteq t(c) \otimes t(c')$$

and the interchange law (2).

Proof. Given the data of the proposition, we reconstruct the data from Definition 2.2 as follows. Let $D_1 = D$ be the domain of definition of $\otimes$. Define $A_0 = \{s(f) \mid f \in C\} = \{t(f) \mid f \in C\}$ and $A_1 = C$. All of $\varepsilon, s(\varepsilon), t(\varepsilon)$ are identities on A_0 w.r.t. $\otimes$, hence we can define $e = \varepsilon = s(\varepsilon) = t(\varepsilon)$ to obtain the partial monoid $(A_0, e, \otimes, D_0)$ with D_0 being the restriction of D_1 to A_0. It also follows that $\varepsilon = \mathrm{id}_e$, hence $(A_1, \mathrm{id}_e, \otimes, D_1)$ is the other monoid. And $\otimes$ is a functor since it preserves identities by definition and preserves composition due to the interchange law. □

A 2-category (C_0, C_1, C_2) almost becomes a p-monoidal category by taking the arrows C_1 as the objects A_0, but not quite, since a p-monoidal category needs to have a neutral element e. But there is a notion of partially monoidal category without unit that comprises 2-categories as a special case.

Example 2.5. We fix a countably infinite set Σ. The category $\mathfrak{F}$ of finite subsets of Σ with $\oplus$ the partially defined union of disjoint sets is a symmetric p-monoidal category. We denote by $\mathfrak{B}$ the subcategory of bijections.

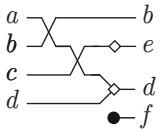
Remark 2.6. The category $\mathfrak{F}$ is equivalent as a category to the category $\mathbb{N}$ of natural numbers with functions. But they are not equivalent as strict partially monoidal categories. While there is a strict monoidal functor $|-| : \mathfrak{F} \to \mathbb{N}$ mapping a set to its cardinality, any inverse $F : \mathbb{N} \to \mathfrak{F}$ will only satisfy $F(n + m) \cong F(n) \oplus F(m)$ and thus not be *strict*. A related observation is that while $\mathfrak{F}$ is commutative, $\mathbb{N}$ is not: Even though $n + m$ equals $m + n$, the symmetry $n + m \to m + n$ is not the identity. This is another sense in which the p-monoidal category $\mathfrak{F}$ is easier to work with than the monoidal category $\mathbb{N}$.

We will see variations on this example in the next section. Semantically, we will have the opportunity to replace sets by words or multisets. Syntactically, we will represent $\mathfrak{B}$ and $\mathfrak{F}$ by a string diagrammatic calculus.

In our examples, the monoid operation is the partial union of disjoint sets. There are various ways in which one can turn this operation into a total operation, but that would introduce technicalities that would take us further away from the aim of this paper: Syntactic representations of $\mathfrak{B}$ and $\mathfrak{F}$ up to isomorphism that correspond closely to how we work with simultaneous substitutions in an informal way. As emphasised above, we are interested in a mechanism that reflects directly that $[a{\mapsto}b, a{\mapsto}c]$ is not a valid simultaneous substitution.

3 Syntax and Semantics

The syntax we will develop is that of string diagrams



(3)

with wires labelled from an infinite, countable alphabet Σ, the elements of which are written a, b, c etc. The semantics we are interested in is that of functions between finite sets. For example, the diagram above will correspond to the function

$$\{a,b,c,d\} \to \{b,d,e,f\}$$
$$a,d \mapsto d \quad b \mapsto b \quad c \mapsto e$$

There are three different ways to formalise this.

First, we treat wires as ordered and labelled with elements of Σ. Sequential composition respects the order of the wires. Parallel composition is partial because distinct wires should be labelled with distinct names. For example, $[a{\mapsto}b] \oplus [a{\mapsto}c]$ is not defined.

Second, we treat wires as ordered and number them explicitly. The label of a wire is then an occurrence of a, that is, a pair (i, a) where i is a number and a a name. Parallel composition can then be total and distinct wires will still have distinct labels as multiple occurrences of the same name are now distinguished by different indices i. But we need to be careful because we can now build diagrams such as $[a{\mapsto}b, a{\mapsto}c]$ that do not denote functions between subsets of Σ.

Third, we treat wires as unordered. Instead of thinking of ordered wires lined up in a linear fashion top to bottom, we now picture them as coming out of plane with no particular order between them. Sequential composition is still uniquely defined as each wire carries a unique label. Another way to look at it is that the rewrite rules of diagrams need to be understood modulo exchange of wires. Accordingly, proofs are one step further away from what would be implemented in a proof assistant but easier for human consumption and closer to geometric intuition.

Each of the three approaches can be understood as dealing in different ways with the simple fact that a set is a list modulo exchange and contraction. In the first and third approach, contraction is built into the data structure by restricting our categories to short diagrams. Consequently, parallel composition must be partial. Moreover, in the third approach, also

exchange is built in. In the second approach, parallel composition is total and it is possible to build diagrams that are not short and do not correspond to functions between sets of names. We show that the first and second approach generate the same equivalence relation on repetition-free diagrams. Alternatively, one could investigate adding explicit equations for contractions, which we do not do in this paper.

In the following we will discuss these three approaches in turn. We start with some notational preliminaries.

We fix an infinite, countable alphabet Σ, the elements of which are written $a, b, c, \ldots$ and are called letters, names, or variables. The number n is identified with the set

$$n = \{0, \ldots n-1\}$$

A word of length n is a function

$$\overrightarrow{X} : n \to \Sigma,$$

that is, an element of Σ^n. We may write a word $\overrightarrow{X}$ also as $(x_0, \ldots x_{n-1})$. If we want to emphasise that the range of a word is a set X of names, we write a word as

$$\overrightarrow{X} : n \to X,$$

or also as

$$\overrightarrow{X} : ||\overrightarrow{X}|| \to X,$$

where $n = ||\overrightarrow{X}||$ denotes the length of the word $\overrightarrow{X}$.

Since we are interested in representing functions between sets, we sometimes want to restrict attention to words that do not have multiple occurrences of the same letter. We call these words short.

Definition 3.1. A word $\overrightarrow{X} : ||\overrightarrow{X}|| \to X$ is called **short** if $\overrightarrow{X}$ is a bijection.

Writing $|-|$ for the cardinality of a set, if $\overrightarrow{X}$ is short then $||\overrightarrow{X}|| = |X|$ and we also write $\overrightarrow{X} : |X| \to X$.

Similarly, a **short diagram** is a diagram where distinct wires are labelled with distinct names. For example, Diagram (3) is short.

3.1 Ordered sets of wires

Semantically, we are considering here the category of ordered sets, that is, ordered subsets of names. More formally, we define the category of ordered sets as follows.

Definition 3.2. The category of (finite) ordered sets F is defined as the category that has short words over the alphabet Σ as objects and has as arrows $(f, g) : \overrightarrow{X} \to \overrightarrow{Y}$ commutative squares

$$\begin{array}{ccc} |X| & \xrightarrow{\quad f \quad} & |Y| \\ {\scriptstyle \overrightarrow{X}}\downarrow & & \downarrow{\scriptstyle \overrightarrow{Y}} \\ X & \xrightarrow[\quad g \quad]{} & Y \end{array}$$

We denote by B the subcategory where all g (hence f) are bijective.

While we are interested in the meaning of a diagram as a function between subsets of Σ, we start by interpreting them as arrows between words. The reason is that in a diagram the order of wires matters.

Remark 3.3. Since all functions in the commutative square making up an arrow (f, g) are bijective, the functions f and g determine each other.

Syntactically, diagrams are arrows in a syntactic category where objects are short words and arrows are built up from the basic diagrams and sequential and parallel composition:

Definition 3.4. The basic diagrams

$$\tau \text{ (twist)}, \ \delta \text{ (renaming)}, \ \mu \text{ (substitution) and } \eta \text{ (lollipop)}$$

are parameterised by distinct $a, b \in \Sigma$ and have the following interpretation as arrows $\Sigma^n \to \Sigma^m$

$$[\![\ \overset{a}{\underset{b}{\times}} \ ]\!] = \begin{array}{ccc} 2 & \xrightarrow{\substack{0 \mapsto 1 \\ 1 \mapsto 0}} & 2 \\ {\scriptstyle 0 \mapsto a \atop 1 \mapsto b}\downarrow & & \downarrow{\scriptstyle 0 \mapsto b \atop 1 \mapsto a} \\ \{a, b\} & \xrightarrow[id]{} & \{a, b\} \end{array} \qquad [\![\ a \diamond b \ ]\!] = \begin{array}{ccc} 1 & \xrightarrow{id} & 1 \\ {\scriptstyle 0 \mapsto a}\downarrow & & \downarrow{\scriptstyle 0 \mapsto b} \\ \{a\} & \xrightarrow[a \mapsto b]{} & \{b\} \end{array}$$

$$[\![\ \overset{a}{\underset{b}{\diamond}} \!\!- b \ ]\!] = \begin{array}{ccc} 2 & \xrightarrow{\substack{0 \mapsto 0 \\ 1 \mapsto 0}} & 1 \\ {\scriptstyle 0 \mapsto a \atop 1 \mapsto b}\downarrow & & \downarrow{\scriptstyle 0 \mapsto b} \\ \{a, b\} & \xrightarrow[\substack{a \mapsto b \\ b \mapsto b}]{} & \{b\} \end{array} \qquad [\![\ \bullet\!\!- a \ ]\!] = \begin{array}{ccc} \varnothing & \xrightarrow{\varnothing} & 1 \\ {\scriptstyle \varnothing}\downarrow & & \downarrow{\scriptstyle 0 \mapsto a} \\ \varnothing & \xrightarrow[\varnothing]{} & \{a\} \end{array}$$

Next, we generate a partially monoidal category from the above basic diagrams and sequential and parallel composition. Sequential composition of diagrams is given by sequential composition of functions:

$$\begin{array}{ccc} m & \xrightarrow{f} & n \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle j} \\ X & \xrightarrow[w]{} & Y \end{array} \ ; \ \begin{array}{ccc} n & \xrightarrow{g} & o \\ {\scriptstyle j}\downarrow & & \downarrow{\scriptstyle k} \\ Y & \xrightarrow[v]{} & Z \end{array} = \begin{array}{ccccc} m & \xrightarrow{f} & n & \xrightarrow{g} & o \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle j} & & \downarrow{\scriptstyle k} \\ X & \xrightarrow[w]{} & Y & \xrightarrow[v]{} & Z \end{array}$$

The notation above implies that the wires that are composed, denoted j, agree in number, order and labelling.

Parallel composition is partial, as it is only defined when $X \cap W = Y \cap V = \varnothing$:

$$\begin{array}{ccc} m & \xrightarrow{f} & n \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle j} \\ X & \xrightarrow[w]{} & Y \end{array} \ \oplus \ \begin{array}{ccc} o & \xrightarrow{g} & p \\ {\scriptstyle k}\downarrow & & \downarrow{\scriptstyle l} \\ W & \xrightarrow[v]{} & V \end{array} = \begin{array}{ccc} m + o & \xrightarrow{f \oplus g} & n + p \\ {\scriptstyle i \oplus k}\downarrow & & \downarrow{\scriptstyle j \oplus l} \\ X \uplus W & \xrightarrow[w \uplus v]{} & Y \uplus V \end{array}$$

where $\uplus = \cup$, due to the partiality constraint, and $\oplus$ is defined as:

$$f \oplus g : m + o \to n + p,$$
$$f \oplus g(j) = \begin{cases} f(j) & \text{for } 0 < j \leq m \\ g(j-m) + p & \text{for } m < j \leq m + o \end{cases}$$

and

$$i \oplus k : m + o \to X \uplus W,$$
$$i \oplus k(j) = \begin{cases} i(j) & \text{for } 0 < j \leq m \\ g(j-m) & \text{for } m < j \leq m + o \end{cases}$$

Definition 3.5. We write $\mathbb{F}$ for the partially monoidal category that has short words as objects and arrows freely generated from instances of τ, δ, μ, η. The partially monoidal category $\mathbb{B}$ is the subcategory freely generated from instances of τ and δ only.

Arrows are also called diagrams. Due to being partial, the parallel composition preserves shortness of words and, therefore, if $\phi : \overrightarrow{X} \to \overrightarrow{Y}$ is a diagram, then $\overrightarrow{X}$ and $\overrightarrow{Y}$ are short.

On the semantic side, we recall that $\mathfrak{F}$ and $\mathfrak{B}$ denote the partially monoidal categories of (finite) functions and bijections, respectively. We have functors

$$\mathbb{F} \xrightarrow{[\![-]\!]} \mathsf{F} \xrightarrow{|-|} \mathfrak{F}$$

and

$$\mathbb{B} \xrightarrow{[\![-]\!]} \mathsf{B} \xrightarrow{|-|} \mathfrak{B}$$

where the forgetful functor $|-|$ maps an arrow (f, g) of words to the function g.

Proposition 3.6. The semantics extends to partially monoidal functors $|[\![-]\!]| : \mathbb{B} \to \mathfrak{B}$ and $|[\![-]\!]| : \mathbb{F} \to \mathfrak{F}$. In particular, if $(f, g) = [\![\phi : w \to v]\!]$, then g is a function from the set of letters of w to the set of letter of v. Moreover, if ϕ is in $\mathbb{B}$ then g is a bijection.

In Section 4 we are going to axiomatise the theory of bijections that describes which diagrams are identified by $|[\![-]\!]| : \mathbb{B} \to \mathfrak{B}$ and in Section 5 the theory of functions that describes which diagrams are identified by $|[\![-]\!]| : \mathbb{F} \to \mathfrak{F}$.

3.2 Ordered multisets of wires

The aim of this section is to investigate what happens if we make parallel composition total. One reason for doing this is that we will prove that even though the resulting rewriting system may take detours via meaningless diagrams, it is the case that every rewrite in the 'total system' between two short words corresponds to some rewrite in the 'partial system'.

For example, in this section we will allow to compose $[a \mapsto b] \oplus [a \mapsto c] = [a \mapsto b, a \mapsto c]$. Semantically, we make this correspond to a function $\{(0, a) \mapsto (0, b), (1, a) \mapsto (1, c)\}$ between not sets but occurrences of names. Accordingly, in the semantics we will use instead of words $(x_0, \ldots x_{n-1})$ words of pairs $((0, x_0), \ldots (n-1, x_{n-1}))$.

Technically, going to a total parallel composition corresponds to going from ordered sets of names to ordered sets of occurrences of names, from ordered sets to ordered multisets (ordered multisets are pomsets [10] where the order happens to be linear), and from short words to words.

Definition 3.7. The category of words with functions WF is defined as the category of words over the alphabet Σ with an arrow $(f, g) : \overrightarrow{X} \to \overrightarrow{Y}$ being a commutative square, where $m = ||\overrightarrow{X}||$ and $n = ||\overrightarrow{Y}||$,

$$\begin{array}{ccc} m & \xrightarrow{\quad f \quad} & n \\ {\scriptstyle \overrightarrow{X}}\downarrow & & \downarrow{\scriptstyle \overrightarrow{Y}} \\ X & \xrightarrow[\quad g \quad]{} & Y \end{array}$$

modulo an equivalence relation on arrows $(f, g) \simeq (f', g')$ if $f = f'$. We denote by WB the subcategory of arrows (f, g) where f is bijective.

The equivalence relation on arrows is justified by the observation that, on the image of $\overrightarrow{X}$, the arrow g is determined by f. (The reason we are only interested in the image of $\overrightarrow{X}$ is that this image determines the word uniquely.)

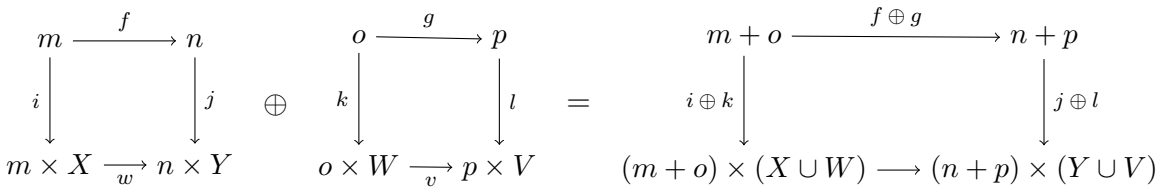
$$\left[\!\!\left[\begin{array}{c} a \quad b \\ \diagdown\!\diagup \\ b \quad a \end{array} \right]\!\!\right] = \begin{array}{ccc} & {\scriptstyle 0 \mapsto 1,\ 1 \mapsto 0} & \\ 2 & \longrightarrow & 2 \\ {\scriptstyle 0 \mapsto (0,a),\ 1 \mapsto (1,b)}\downarrow & & \downarrow{\scriptstyle 0 \mapsto (0,b),\ 1 \mapsto (1,a)} \\ 2 \times \{a,b\} & \xrightarrow[\substack{(0,a) \mapsto (1,a) \\ (1,b) \mapsto (0,b)}]{} & 2 \times \{a,b\} \end{array} \qquad \left[\!\!\left[\begin{array}{c} a \quad b \\ \diamond \end{array} \right]\!\!\right] = \begin{array}{ccc} 1 & \xrightarrow{\quad id \quad} & 1 \\ {\scriptstyle 0 \mapsto (0,a)}\downarrow & & \downarrow{\scriptstyle 0 \mapsto (0,b)} \\ 1 \times \{a\} & \xrightarrow[(0,a) \mapsto (0,b)]{} & 1 \times \{b\} \end{array}$$

$$\left[\!\!\left[\begin{array}{c} a \\ \diagdown\!\diagup\!\diamond\!\!- b \\ b \end{array} \right]\!\!\right] = \begin{array}{ccc} 2 & \longrightarrow & 1 \\ {\scriptstyle 0 \mapsto (0,a),\ 1 \mapsto (1,b)}\downarrow & & \downarrow{\scriptstyle 0 \mapsto (0,b)} \\ 2 \times \{a,b\} & \longrightarrow & 1 \times \{b\} \end{array} \qquad \left[\!\!\left[\begin{array}{c} \bullet\!\!- a \end{array} \right]\!\!\right] = \begin{array}{ccc} \varnothing & \xrightarrow{\quad \varnothing \quad} & 1 \\ {\scriptstyle \varnothing}\downarrow & & \downarrow{\scriptstyle 0 \mapsto (0,a)} \\ \varnothing & \xrightarrow[\quad \varnothing \quad]{} & 1 \times \{a\} \end{array}$$

Next, we generate a partially monoidal category from the above basic diagrams by closing under sequential and parallel composition. Sequential composition of diagrams is given by sequential composition of functions:

$$\begin{array}{ccc} m & \xrightarrow{\quad f \quad} & n \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle j} \\ m \times X & \xrightarrow[\quad w \quad]{} & n \times Y \end{array} \quad ; \quad \begin{array}{ccc} n & \xrightarrow{\quad g \quad} & o \\ {\scriptstyle j}\downarrow & & \downarrow{\scriptstyle k} \\ n \times Y & \xrightarrow[\quad v \quad]{} & o \times Z \end{array} \quad = \quad \begin{array}{ccccc} m & \xrightarrow{\quad f \quad} & n & \xrightarrow{\quad g \quad} & o \\ {\scriptstyle i}\downarrow & & \downarrow{\scriptstyle j} & & \downarrow{\scriptstyle k} \\ m \times X & \xrightarrow[\quad w \quad]{} & n \times Y & \xrightarrow[\quad v \quad]{} & o \times Z \end{array}$$

The parallel composition is now total:



Definition 3.8. We write $\overline{\mathbb{F}}$ for the monoidal category that has words as objects and has arrows that are freely generated from instances of τ, δ, μ, η. $\overline{\mathbb{B}}$ is the monoidal subcategory generated from τ and δ only.

Proposition 3.9. The semantics extends to monoidal functors $[\![-]\!] : \overline{\mathbb{B}} \to \mathsf{WB}$ and $[\![-]\!] : \overline{\mathbb{F}} \to \mathsf{WF}$.

The next proposition says that if a diagram in $\overline{\mathbb{F}}$ is short, then it induces, and is determined by, a unique function between sets of names (denoted g' in the proposition).

Proposition 3.10. Let $\phi : \overrightarrow{X} \to \overrightarrow{Y}$ be a diagram in $\overline{\mathbb{F}}$ and $(f, g) = [\![\phi]\!]$. If $\overrightarrow{X}$ and $\overrightarrow{Y}$ are short, then $g : ||\overrightarrow{X}|| \times X \to ||\overrightarrow{Y}|| \times Y$ is of the form $g = f \times g'$ for a unique function $g' : X \to Y$.

Proof. This follows since shortness means that $\overrightarrow{X}$ and $\overrightarrow{Y}$ are bijections. In detail, we have

$$\begin{array}{ccc} |X| & \xrightarrow{\quad f \quad} & |Y| \\ {\scriptstyle \langle \mathrm{id}, \overrightarrow{X} \rangle}\downarrow & & \downarrow{\scriptstyle \langle \mathrm{id}, \overrightarrow{Y} \rangle} \\ |X| \times X & \xrightarrow[\quad g \quad]{} & |Y| \times Y \end{array}$$

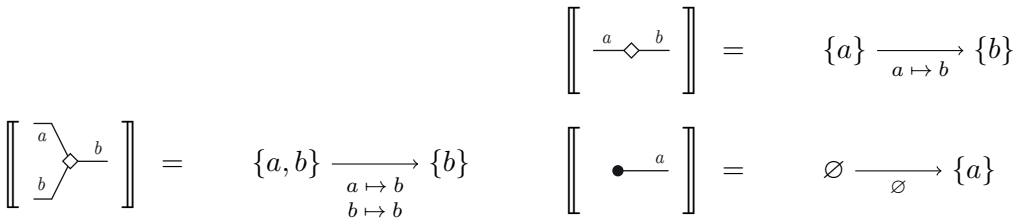
and define $g' = \overrightarrow{Y} \circ f \circ \overrightarrow{X}^{-1}$. The equation $(f \times g') \circ \langle \mathrm{id}, \overrightarrow{X} \rangle = \langle \mathrm{id}, \overrightarrow{Y} \rangle \circ f$ follows immediately as well as that any g' satisfying this equation is uniquely determined. □

3.3 Sets of wires

In this section, we change the notion of sequential composition so that it ignores the ordering of the wires. This is possible because, as in Section 3.1, every wire will carry a unique label. Thus, the domain and codomain of a diagram $\phi : X \to Y$ are sets of wires.

In the previous Sections 3.1 and 3.2, even though the generator $\tau =$

defines the identity function $\{a, b\} \to \{a, b\}$, we could not add the equation $\tau = \mathrm{id}$ as sequential composition had to respect the order of the wires and the labels. With the new sequential composition we could add this equation, but it seems easier to just drop τ from the generators and to take the domain and codomain of a diagram to be not words of labels but sets of labels. The semantics of δ (renaming), μ (substitution), and η (lollipop) can then be given directly in terms of functions.



Sequential composition of diagrams is described as above by linking wires with the same label. Parallel composition is stacking diagrams on top of each other and is partial as it has to respect the shortness constraints.

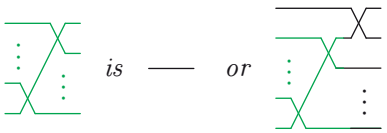
Definition 3.11. We write $\widetilde{\mathbb{F}}$ for the partially monoidal category freely generated from δ, μ, η and parallel and modified sequential composition as described above. We write $\widetilde{\mathbb{B}}$ for the partially monoidal category freely generated from δ only.

Proposition 3.12. The semantics extends to partially monoidal functors $[\![-]\!] : \widetilde{\mathbb{B}} \to \mathfrak{B}$ and $[\![-]\!] : \widetilde{\mathbb{F}} \to \mathfrak{F}$.

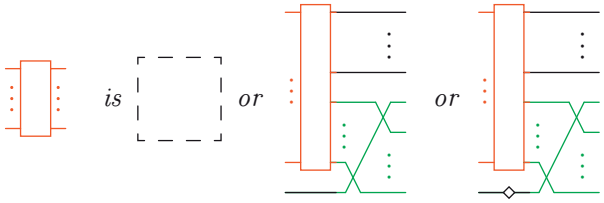
4 The Theory of Bijective Functions

In this section we will consider the category $\mathfrak{B}$ of bijections of finite subsets of some set Σ, with the generator $\delta_{ab} : \{a\}^1 \to \{b\}^1$ representing a bijection $\{a\} \rightarrowtail\!\!\!\!\!\twoheadrightarrow \{b\}$ and $\tau_{ab} : \{a, b\}^2 \to \{a, b\}^2$ representing the identity function $id_{\{a,b\}} : \{a, b\} \to \{a, b\}$.

Following Lafont [7], we introduce a notion of a canonical form for string diagrams formed from the generators τ and δ, defined in the previous section. We first inductively define the notion of *stairs*:



Next we define the *canonical form*:

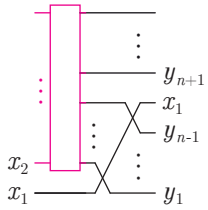


In both instances, we omit the names/labels on the wires for better readability.

Lemma 4.1. *Any bijective function $f : X \rightarrowtail\!\!\!\!\!\twoheadrightarrow Y$ together with an ordering on X and Y (given by $\overrightarrow{X} : |X| \rightarrowtail\!\!\!\!\!\twoheadrightarrow X$ and $\overrightarrow{Y} : |Y| \rightarrowtail\!\!\!\!\!\twoheadrightarrow Y$) is represented by a unique canonical form.*

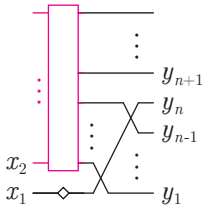
Proof. By induction on the size n of X and Y:

- If $n = 0$, then f is the identity function on the empty set and is represented by the empty string diagram.
- If $n \geq 1$, then given $\overrightarrow{X}$ and $\overrightarrow{Y}$, we have $x_1 = \overrightarrow{X}(1)$ and $y_n = f(x_1)$ (where $n = \overrightarrow{Y}^{-1}(f(x_1))$). Now, we have two cases, either $x_1 = y_n$, in which case we will have the diagram:



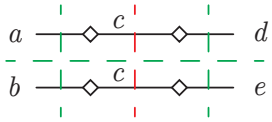
where the remaining part (in magenta) is given by the IH, by removing x_1 and y_n from f, $\overrightarrow{X}$ and $\overrightarrow{Y}$ and re-numbering the order functions.

In case $x_1 \neq y_n$, we get the following diagram (with the IH giving the remaining part, as in the previous case):



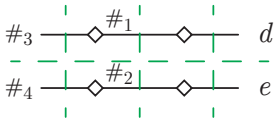
□

Before we show that the rewriting system of Figure 1 is terminating and rewrites to the canonical form, we have to take care of the fact that due to the partiality of $\oplus$ not all diagrams can be decomposed in the usual fashion:



(4)

Here, the vertical slicing of the diagram in the middle is not allowed, because the two sub-diagrams would violate the shortness constraint, since two c's would appear in the codomain and domain of the left and right sub-diagram, respectively. In order to avoid such conflicts, we define a notion of restricted substitution, wherein we replace repeated names with fresh ones, such that the new diagram can be sliced arbitrarily without restriction. For example, the substitution renames both c's into fresh variables:



As we will show, the rewriting system in Figure 1 axiomatises the theory of bijections. Notice that we elided the labels in some of the equations, which then need to be instantiated by appropriately labelling the wires as shown in Section 3.1.

Lemma 4.2. *Any diagram $\phi : \overrightarrow{X} \to \overrightarrow{Y}$ in which all internal names are fresh reduces to a canonical form $\widehat{\phi}$ by the rules of Figure 1.*

Proof. We prove this lemma by double induction on the number p of input/output ports, $p = |\overrightarrow{X}| = |\overrightarrow{Y}|$, together with the *size* s of ϕ, defined as the number of generators which make up the diagram.

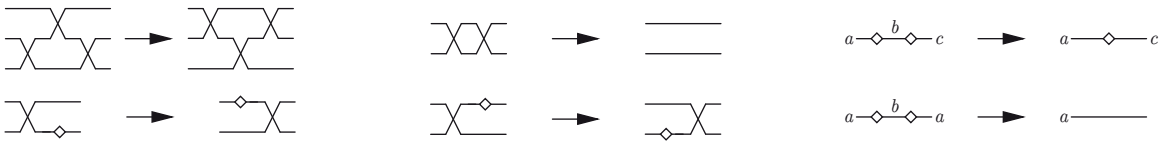
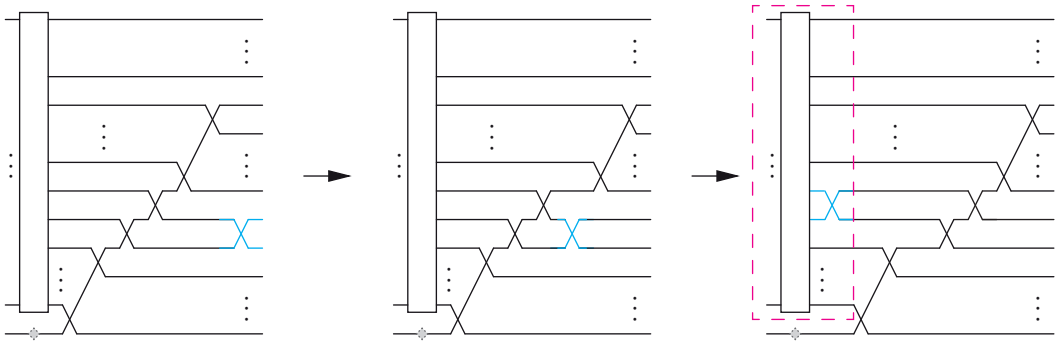


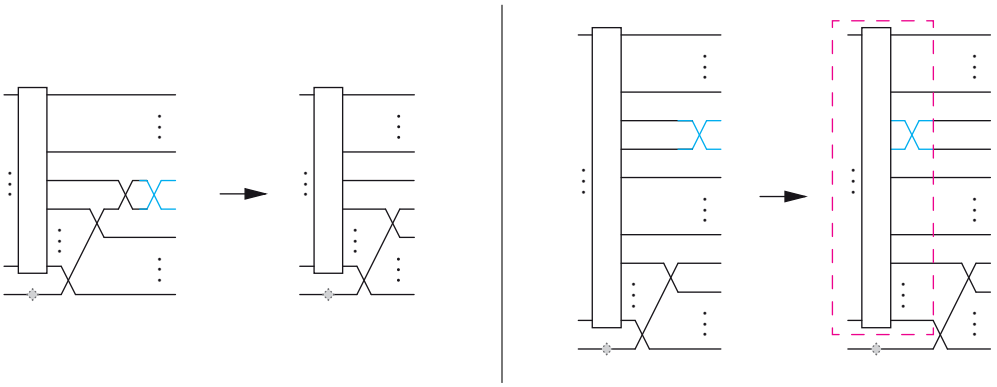
Figure 1: Rewrite rules of $\mathbb{B}$

- If $s = 0$, then $\phi = id_{\overrightarrow{X}} = id_{\overrightarrow{Y}}$, which is a canonical form.
- If $s \geq 1$ then we would like to separate the diagram into an elementary diagram ε and a diagram ψ of size $s - 1$, s.t. $\phi = \psi \,;\, \varepsilon$. However, we need to be careful as this might not always be possible.
  - If $\varepsilon = \tau$, the proof mirrors Lafont's in the usual way. There are 4 cases (the case for the canonical form with and without the diamond are exactly the same in these 4 cases and we highlight ε in cyan):

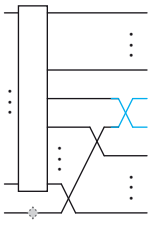
    In the first case, we apply the first rule and then apply the IH to a sub-diagram with $p - 1$ ports (outlined in magenta).



    In the second case, we apply the second rule and obtain a diagram in canonical form. In the third case, we use the IH for $p - 1$ again.

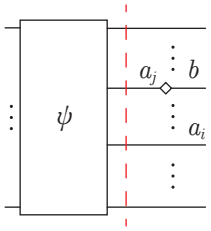


    The last case is a canonical form already.



– If $\varepsilon = \delta$, things are a bit more tricky as we cannot always decompose the diagram in the desired way.

Consider the following case:

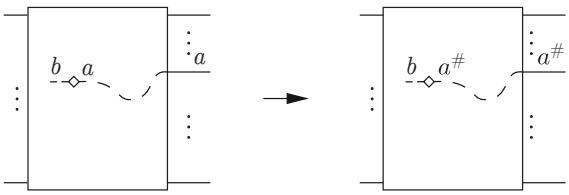


(5)

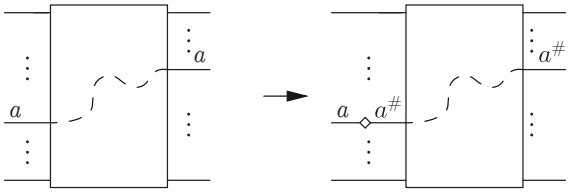
The diagram above is problematic, as we we cannot split it along the dashed line, since the smaller diagram ψ will no longer be short.

In order to proceed, we define a notion of restricted substitution on a diagram, where we replace one specific occurrence of a label with a fresh name, starting from the output port going backwards. There will be two cases:

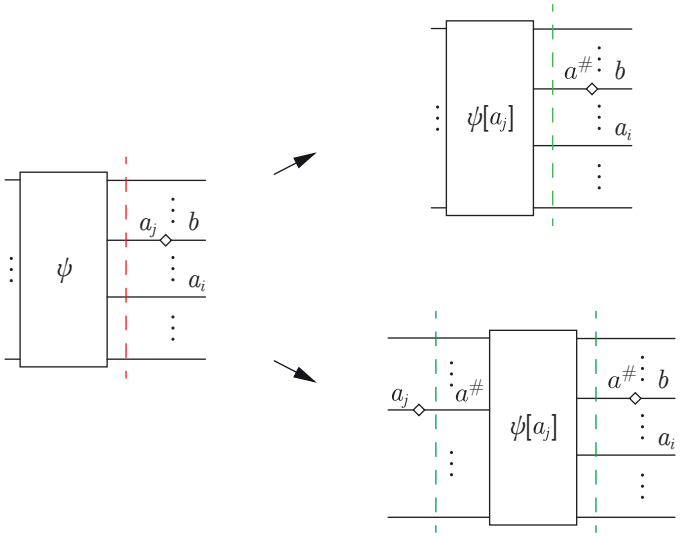
In the first case, the operation traces the label a backwards, replacing it with a fresh label $a^{\#}$, until it reaches a diamond:



In the other case, the label is traced all the way back to the input port, in which case we get the following diagram:



Using this operation we apply it to the problematic diagram and due to the definition of the operation, obtain two cases:

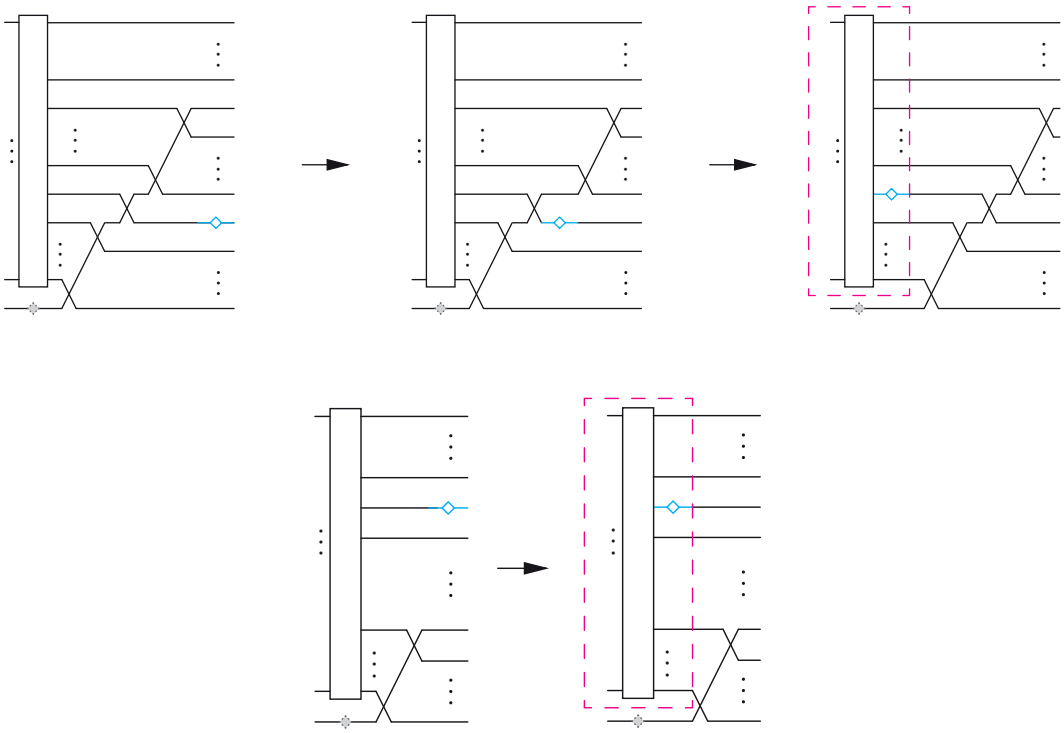


This now allows us to separate the original diagram ϕ into $\psi[a_j]\,;\,(\overrightarrow{id}\oplus\delta_{a^\#b}\oplus\overrightarrow{id})$ or $(\overrightarrow{id}\oplus\delta_{a_ja^\#}\oplus\overrightarrow{id})\,;\,\psi[a_j]\,;\,(\overrightarrow{id}\oplus\delta_{a^\#b}\oplus\overrightarrow{id})$.

Since, by definition, $a^\#$ is a fresh variable, not appearing anywhere in the diagram, this decomposition is defined and since $size(\psi[a_j]) = size(\psi) = s-1$, we can apply the IH to $\psi[a_j]$ and obtain $\widehat{\psi[a_j]}$, which is in canonical form.

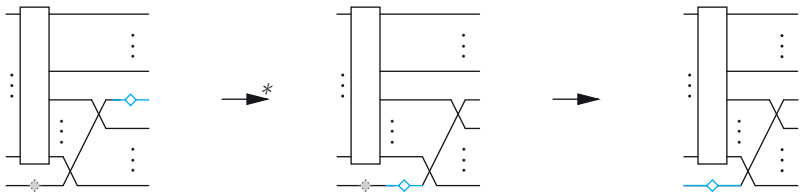
We analyze the following three cases of the subdiagram $\widehat{\psi[a_j]}\,;\,(\overrightarrow{id}\oplus\delta_{a^\#b}\oplus\overrightarrow{id})$, ignoring the second case of the substitution above (for now).

For the first case, we simply slide the diamond past the twist (4th rule) and apply IH to a subdiagram with $p-1$ ports (likewise for the second case):

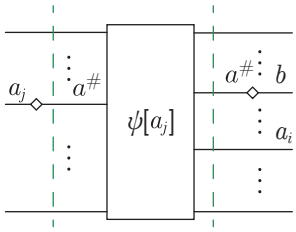


The last case involves multiple application of the last rule, sliding the diamond past all the twists of the *stairs* (this derived rule is easily proven by induction

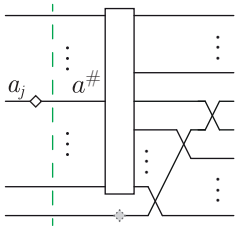
on the height of the *stairs*). Here, we have two further cases. Either the normal form has a diamond at the bottom most port, in which case we apply rule 3 and obtain a diagram in canonical form, otherwise, the last port is an identity and we already have a canonical form.



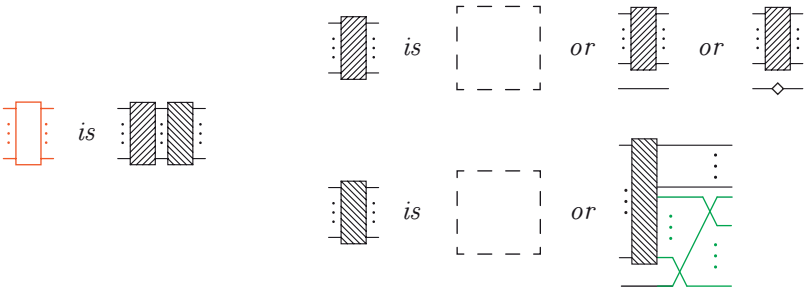
Finally, we get back to the other case of the diagram substitution, namely:



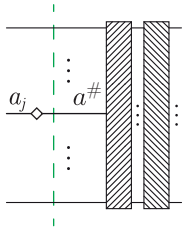
We apply the same reasoning as for the first substitution case and obtain the following diagram (which is almost in canonical form):



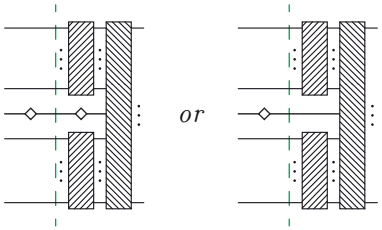
In order to show that the diagram above is/can be turned into canonical form, we will modify the definition of canonical form slightly. One can easily see that the following definition of canonical form is equivalent to the one introduced earlier:



Thus, the previous diagram can be rewritten as:



Which, according to the definition above is the same as:



For the first case, we apply the 3rd rule and obtain canonical form and in the second case, we already have canonical form. □

Recall from Section 3.1 that the partially monoidal category $\mathbb{B}$ is 'free over twist and diamond' and that there is a functor

$$[\![-]\!] : \mathbb{B} \to \mathsf{B}$$

into the category of short words with bijections. We have shown that the kernel of this functors is axiomatised by the equations of Figure 1:

Theorem 4.3. *The partially monoidal category $\mathbb{B}$ modulo the equations from Figure 1 is isomorphic to the partially monoidal category B of short words with bijections.*

Proof. A quick inspection verifies that the left and right-hand side of all rewrites in Figure 1 are mapped to the same bijection between short words in B. This shows the soundness of the equations. For completeness, we need to show that if two diagrams ϕ, ψ are identified by $[\![-]\!]$, then they can be proved equal using the equations. By Lemma 4.1, we know that $[\![\phi]\!] = [\![\psi]\!]$ has a unique canonical form and by Lemma 4.2 we know that both ϕ and ψ rewrite to this canonical form. Hence ϕ and ψ are equal according to the equations. □

The next theorem shows that when using equational reasoning, we can work with a total parallel composition. While not all diagrams showing up in such a proof correspond to functions between sets of names, they do so if their domain and codomain are short words.

Recall $[\![-]\!] : \overline{\mathbb{B}} \to \mathsf{WB}$ from Proposition 3.9.

Theorem 4.4. *Let ϕ, ψ be two short diagrams in $\overline{\mathbb{B}}$ such that $\phi = \psi$ in the equational theory of $\overline{\mathbb{B}}$ plus the equations of Figure 1. Then $\phi = \psi$ in the equational theory of $\mathbb{B}$ plus the equations of Figure 1.*

Proof. There are more equations in $\overline{\mathbb{B}}$ than in $\mathbb{B}$ because the interchange law in $\mathbb{B}$ is restricted by the partiality of parallel composition, see (4). Nevertheless, if $\phi = \psi$ in the equational theory of $\overline{\mathbb{B}}$ then $[\![\phi]\!] = [\![\psi]\!]$ in WB because the equations of Figure 1 are sound wrt to $[\![-]\!] : \overline{\mathbb{B}} \to \mathsf{WB}$. But since ϕ and ψ are short, $[\![\phi]\!] = [\![\psi]\!]$ in WB implies $[\![\phi]\!] = [\![\psi]\!]$ in B. Now the result follows from the completeness part of Theorem 4.3. □

Next, we come to the question of representing the category of $\mathfrak{B}$ up to isomorphism. Intuitively, the presentation $\mathbb{B}$ plus the equations of Figure 1 present $\mathfrak{B}$ up to isomorphism once we add equations between objects identifying all words that only differ in the order of their letters. This can be made precise by adapting the notion of *presentation modulo* of Curien and Mimram [1] to partially monoidal categories:

The category presented by $\mathbb{B}$ plus the equations of Figure 1 plus equational generators [1, Def.7] identifying words that only differ in the order of their letters is isomorphic to the category $\mathfrak{B}$ of finite sets of names.

The details will have to be deferred to future work.

If we are willing to work with sets of wires instead of words of wires we obtain the following. Recall Proposition 3.12.

Theorem 4.5. *The partially monoidal category $\widetilde{\mathbb{B}}$ modulo equations*

is isomorphic to the partially monoidal category $\mathfrak{B}$ of finite sets of names with bijections.

Why are the equations of Theorem 5.7 so much simpler than the equations of Figure 1? Geometrically, if wires are sets, then wires do not line up in one dimension but can be pictured as coming out of a 2-dimensional plane as in Fig 3.1 of [6]. Similarly to how the geometry of planar string diagrams trivialises the laws of monoidal categories, going from ordered wires to sets of wires trivialises the equations of Figure 1 that involve twisting of wires. The price to pay is that the diagrams do not live in two dimensional space anymore.

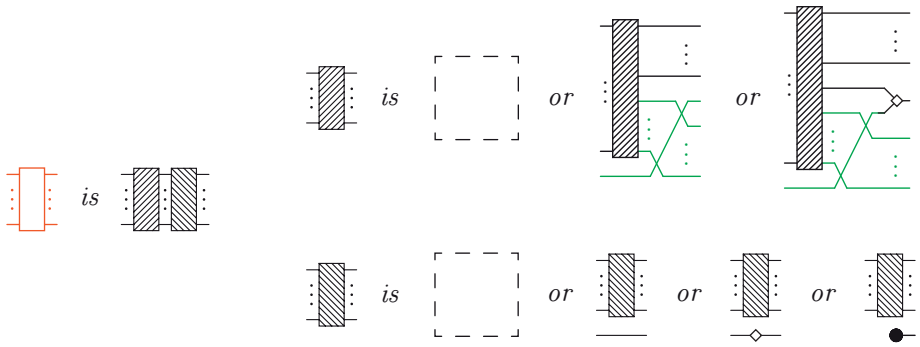
Remark 4.6. Ignoring for a moment the monoidal structure (parallel composition), Theorem 5.7 can also be viewed in the light of Theorem 38 of [1]. We start from a presentation of the category WB (words with bijections) given by δs and equations

We then add (parallel compositions of) τs and μs as what [1] call equational generators and add some more rewrite rules that make sure that each equivalence class (which is now a set of names) has a unique normal form. Then Theorem 38 ensures us that quotienting WB by the equations gives a category that is isomorphic to the subcategory of WB given by the normal forms defined by the equational generators. This can be seen as a modularity result: The part consisting of δ is separated from the part consisting of τ, μ as the latter are now only used in defining the subcategory of normal forms of a category described solely in terms of δs. Theorem 68 of [1] extends this modularity also to monoidal categories. But this cannot be directly applied to our situation as the subcategory of normal forms is not closed under the monoidal composition, which is only partially defined as a composition on normal forms.

5 The Theory of Functions

We extend the results from the previous section from bijections to functions. In other words, going back to Definition 3.5, we extend $\mathbb{B}$ with the generators μ and η, see Definition 3.4.

Again, we define a canonical form for these diagrams:



The rules of this rewrite system are given in Figure 2.

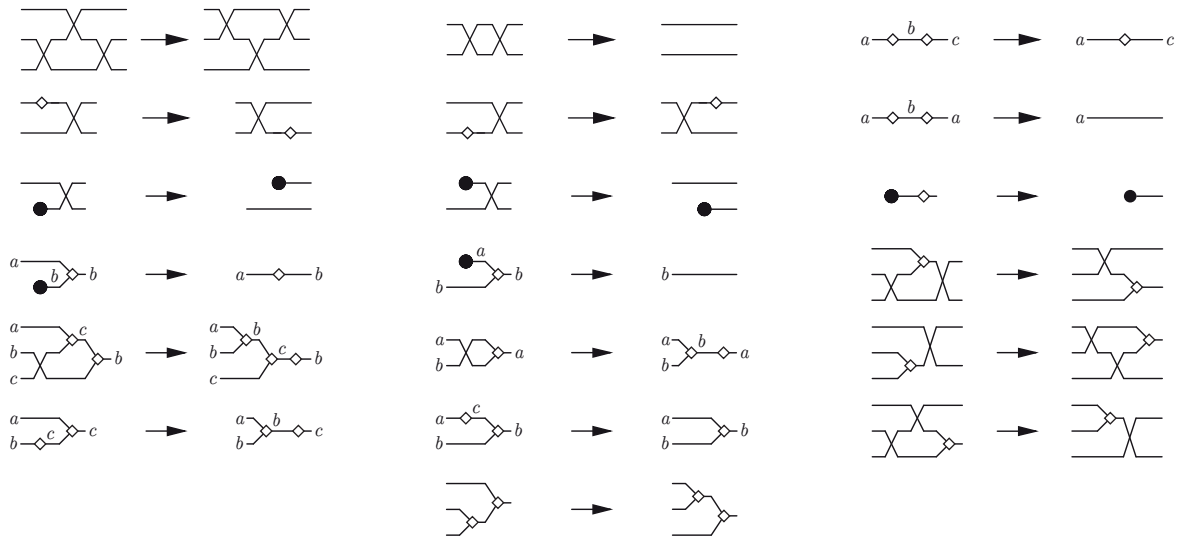
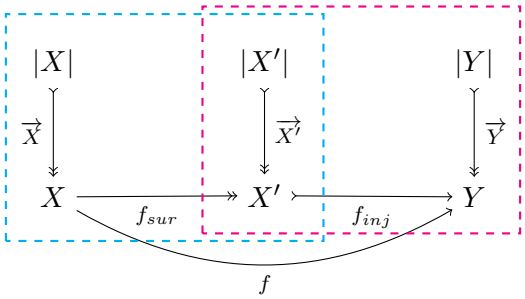


Figure 2: Rewrite rules of $\mathbb{F}$

Lemma 5.1. *Any function $f : \overrightarrow{X} \to \overrightarrow{Y}$ is represented by a unique canonical form in $\mathbb{F}$.*

Proof. We begin with the observation that any function $g : X \to Y$ can be factored as a surjection, followed by an injection in a straightforward way. Thus we can do the following factorization for our function f:



We will first focus on the left part of the picture, namely the surjection. We define f_{sur} in the following way:

$$f_{sur}(x) = max^{\overrightarrow{X}}(2^f(\{f(x)\}))$$

where $2^f : 2^Y \to 2^X$ is the pre-image of f and $max^{\overrightarrow{X}} : 2^X \to X$ is defined as:

$$max^{\overrightarrow{X}}(Z) = \overrightarrow{X}(max(\overrightarrow{X}^{-1}[Z]))$$

($\overrightarrow{X}^{-1}[Z] : 2^X \to 2^{|X|}$ is the inverse function of $\overrightarrow{X}$ lifted to sets)

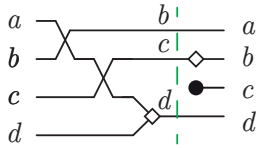
Intuitively, f_{sur} acts as the identity on everything, but the elements, which are identified in the image of f. For those, we take the pre-image of f and choose a canonical/maximal element, which is given to us by the $max^{\overrightarrow{X}}$ function. This function simply takes the set of elements that are to be identified and choses the largest one, according to the given ordering $\overrightarrow{X}$.

Next we need to give the ordering function $\overrightarrow{X'}$. Since this must be a bijection, we will instead give the definition of the inverse function $\overrightarrow{X'}^{-1}$:

$$\overrightarrow{X'}^{-1}(x) = \mathit{shift}(\overrightarrow{Y}^{-1}(f(x)), \overrightarrow{Y}^{-1}(f(x)) + 1)$$

$$\begin{aligned}
&\mathit{shift} : |Y| \times |Y| \to |X|,\\
&\mathit{shift}(g, 0) = g\\
&\mathit{shift}(g, c) = \begin{cases} \mathit{shift}(g, c-1) & \text{if } \overrightarrow{Y}(c-1) \in f[X] \\ \mathit{shift}(g-1, c-1) & \text{otherwise} \end{cases}
\end{aligned}$$

The ordering $\overrightarrow{X'}$ is defined from the ordering $\overrightarrow{Y}$, by essentially composing $\overrightarrow{Y}$ with f and then filtering out the elements in the domain of $\overrightarrow{Y}$, which do not appear in the image of f. It is much easier to see this pictorially:



In the diagrammatic representation of a function f above, f_{sur} is defined as

$$f_{sur}(a) = d \quad f_{sur}(b) = b \quad f_{sur}(c) = c \quad f_{sur}(d) = d$$

and can be read off from the sub-diagram to the left of the dashed line. The ordering $\overrightarrow{X'}$ is:

$$\overrightarrow{X'}(0) = b \quad \overrightarrow{X'}(1) = c \quad \overrightarrow{X'}(2) = d$$

and thus

$$\overrightarrow{X'}^{-1}(b) = 0 \quad \overrightarrow{X'}^{-1}(c) = 1 \quad \overrightarrow{X'}^{-1}(d) = 2$$

We can verify that our definition of $\overrightarrow{X'}^{-1}$ above is correct, by checking that $\overrightarrow{X'}^{-1}(d)$ is really 2.

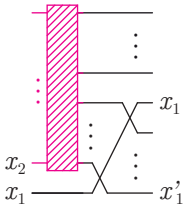
$$
\begin{aligned}
\overrightarrow{X'}^{-1}(d) &= \mathit{shift}(\overrightarrow{Y}^{-1}(f(d)+1), \overrightarrow{Y}^{-1}(f(d))) \\
&= \mathit{shift}(\overrightarrow{Y}^{-1}(d), \overrightarrow{Y}^{-1}(d)+1) \\
&= \mathit{shift}(3,4) \\
&= \mathit{shift}(3,3) && \overrightarrow{Y}(3) \in f[X]\ (\overrightarrow{Y}(3) = d) \\
&= \mathit{shift}(2,2) && \overrightarrow{Y}(2) \notin f[X]\ (\overrightarrow{Y}(2) = c) \\
&= \mathit{shift}(2,1) && \overrightarrow{Y}(1) \in f[X]\ (\overrightarrow{Y}(1) = b) \\
&= \mathit{shift}(2,0) && \overrightarrow{Y}(0) \in f[X]\ (\overrightarrow{Y}(1) = a) \\
&= 2
\end{aligned}
$$

Finally, the right side of the picture, namely the definition of $f_{inj} : X' \to Y$ is simply:

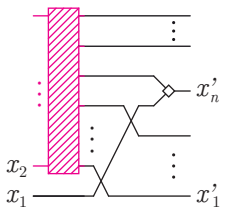
$$f_{inj}(x) = f(x)$$

We will now match the surjective decomposition of f to the left side of the canonical form form and the injective decomposition will correspond to the right canonical diagram.

- Surjection: We proceed by induction on the size n of X:
  - If $n = 0$, then f is the identity function on the empty set and is represented by the empty string diagram.
  - If $n \geq 1$, then given $\overrightarrow{X}$ and $\overrightarrow{X'}$, we have $x_1 = \overrightarrow{X}(1)$ and $x'_n = f_{sur}(x_1)$ (where $n = \overrightarrow{X'}^{-1}(f_{sur}(x_1))$). We have two cases. Either x_1 is mapped to x_1 and no other value is identified with $x_!$ (i.e. $|2^f(\{f(x_1)\})| = 1$), in which case we will have the diagram:

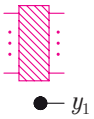
    otherwise, we have $|2^f(\{f(x_1)\})| > 1$ and x_1 is identified with some x'_n, in which case we have:

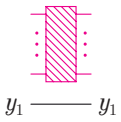
    where the rest of the diagram is given by the IH, by re-numbering $\overrightarrow{X}, \overrightarrow{X'}$ and removing x_1 from the domain of f_{sur}.

- Injection: Again, we proceed by induction, now on the size n of Y:
  - If $n = 0$, then f is the identity function on the empty set and is represented by the empty string diagram.

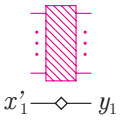
– If $n \geq 1$, then we have $y_1 = \overrightarrow{Y}(1)$. In case we have $y_1 \notin f[X]$, we get the diagram:



Otherwise we have $2^{f_{inj}}(\{y_1\}) = \{x'_1\}$ where we either have $x'_1 = y_1$, in which case we get:



otherwise:



The rest of the diagram in all three cases is given by the IH, by re-numbering $\overrightarrow{Y}$ and removing x'_1 from the domain of f_{inj}.

□

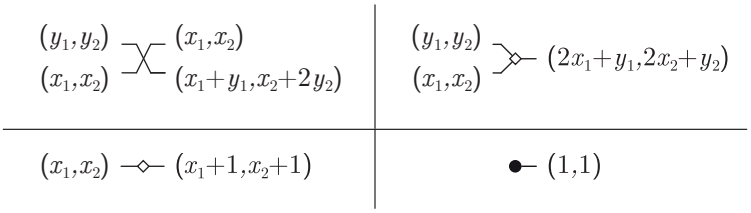
Lemma 5.2. *The rewrite system in Figure 2 is terminating.*

Proof. In order to prove termination, we will use an argument of polynomial interpretation, similar to the one found in Lafont [7].

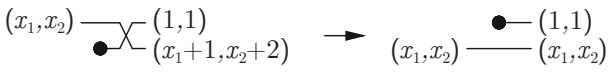
For all diagrams $\sigma : \overrightarrow{X} \to \overrightarrow{Y}$ we will define a strictly monotonic map $[\sigma] : (N^+ \times N^+)^{|X|} \to (N^+ \times N^+)^{|Y|}$, where N^+ is the set of strictly positive integers and $(N^+ \times N^+)^n$ comes with a lexicographic product order:

$$((x_{11}, x_{12}), \ldots, (x_{n1}, x_{n2})) \leq ((y_{11}, y_{12}), \ldots, (y_{n1}, y_{n2})) \text{ whenever}$$
$$(x_{11}, x_{12}) \leq (y_{11}, y_{12}), \ldots, (x_{n1}, x_{n2}) \leq (y_{n1}, y_{n2})$$

where $(x_1, x_2) \leq (y_1, y_2) \stackrel{\text{def}}{=} x_1 \leq y_1 \vee (x_1 = y_1 \wedge x_1 \leq y_2)$
We give a pair of interpretation functions for each of the generators:



This interpretation is compatible with the parallel and sequential composition, and it thus suffices to check that all the rewrite rules, when interpreted, strictly decrease in at least one coordinate. In the example below, this condition is satisfied, since $(x_1+1, x_2+2) > (x_1, x_2)$:



We will omit the rest of the rules. The full proof was formalized and checked using an SMT solver and is discussed in further detail in Section 6. □

Lemma 5.3. *The canonical form, defined at the beginning of this section is a normal form for the rewriting sytem, presented in Figure 2.*

Proof. To see why the canonical form is a normal form, we analyze the rules of the system and argue that the canonical form must be the normal form because it contains no redexes.

Looking at the canonical form, we can see that it is split into the a left and a right canonical form, where the left diagram contains only twists and cups and the right side only contains diamonds and lollipops. We can thus eliminate all the rules which have a lollipop/diamond before a cup or a twist, such as

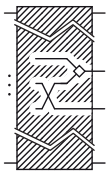
since the diagram on the left of such rule cannot appear neither in the left nor in the right canonical diagram.

For rules, such as

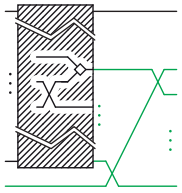
the left hand sides clearly cannot appear in the left canonical form, and by inspecting the right canonical form, we can see that they also cannot appear there, as the canonical right diagram can only ever have one generator at any given port/level.

A similar argument can be made for the rest of the rules, which involve analyzing the left normal form diagram. Take for example

We need to show that the left hand side of the rule could never appear in a canonical form. Due to the shape of the rule we must necessarily have the following diagram:



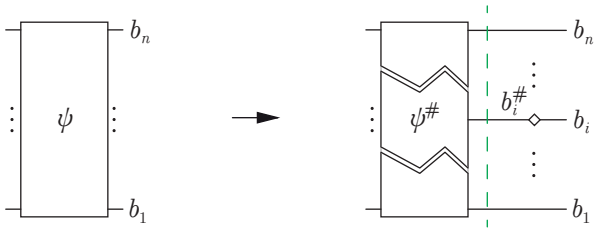
It is then easy to see that there is no way to attach the second twist, such that the resulting diagram is a canonical one, since the only way to attach a twist this diagram is to use stairs, which will lead to



The other rules follow in a similar fashion. □

Lemma 5.4. *The rewrite system in Figure 2 is locally confluent and reduces to the canonical form.*

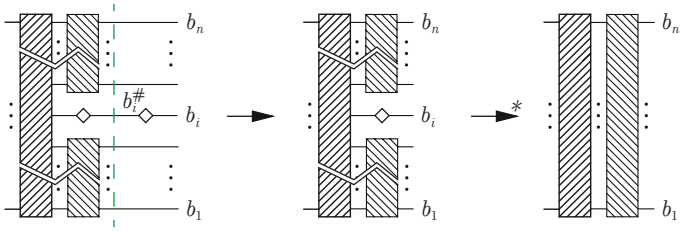
Proof. To avoid the difficulty of the partiality of the parallel composition, we will use a similar trick as in Lemma 4.2, wherein we substitute multiple occurrences of the same variable with fresh ones, s.t. we get the following:



Since the diagram $\psi^\#$ (before the dashed line) only contains one occurrence of each name, all compositions are defined (informally, this is because any sub-diagram will have a short word on both the input and output ports). It now suffices to show that $\psi^\#$ reduces to normal form, by proving local confluence.

To show local confluence, ∼100 critical pairs have to be checked.[1] These are omitted for brevity and can be found on github[2].

Having shown that the normal form is indeed the canonical form in Lemma 5.3, the reduced diagram $\widehat{\psi^\#}$ is of the form:



In order to get the final canonical from, we simply need to apply the 3rd or 6th rule to collapse the two diamonds for all the diamonds introduced by the substitution operation. □

The remainder of this section copies almost verbatim the corresponding part in Section 4, replacing the appropriate occurrences of Bs by Fs.

Recall from Section 3.1 that the partially monoidal category $\mathbb{F}$ is 'free over twist, diamond, cup and lollipop' and that there is a functor

$$[\![-]\!] : \mathbb{F} \to \mathsf{F}$$

into the category of short words with functions. We have shown that the kernel of this functors is axiomatised by the equations of Figure 2:

Theorem 5.5. *The partially monoidal category $\mathbb{F}$ modulo the equations of Figure 2 is isomorphic to the partially monoidal category F of short words with functions.*

[1] As discussed in Section 6, we are currently not confident we have found all the critical peaks and for future work, we plan on generating them automatically.

[2] See https://goodlyrottenapple.github.io/string-diagrams-functions/confluence.html

Proof. A quick inspection verifies that the left and right-hand side of all rewrites in Figure 2 are mapped to the same function between short words in F. This shows the soundness of the equations. For completeness, we need to show that if two diagrams ϕ, ψ are identified by $[\![-]\!]$, then they can be proved equal using the equations. By Lemma 5.1, we know that $[\![\phi]\!] = [\![\psi]\!]$ has a unique canonical form and by Lemmas 5.2 and 5.4 we know that both ϕ and ψ rewrite to this canonical form. Hence ϕ and ψ are equal according to the equations. □

The next theorem shows that when using equational reasoning, we can work with a total parallel composition. While not all diagrams showing up in such a proof correspond to functions between sets of names, they do so if their domain and codomain are short words.

Recall $[\![-]\!] : \overline{\mathbb{F}} \to \mathsf{WF}$ from Proposition 3.9.

Theorem 5.6. *Let ϕ, ψ be two short diagrams in $\overline{\mathbb{F}}$ such that $\phi = \psi$ in the equational theory of $\overline{\mathbb{F}}$ plus the equations of Figure 2. Then $\phi = \psi$ in the equational theory of $\mathbb{F}$ plus the equations of Figure 2.*

Proof. There are more equations in $\overline{\mathbb{F}}$ than in $\mathbb{F}$ because the interchange law in $\mathbb{F}$ is restricted by the partiality of parallel composition. Nevertheless, if $\phi = \psi$ in the equational theory of $\overline{\mathbb{F}}$ then $[\![\phi]\!] = [\![\psi]\!]$ in WF because the equations of Figure 2 are sound wrt to $[\![-]\!] : \overline{\mathbb{F}} \to \mathsf{WF}$. But since ϕ and ψ are short, $[\![\phi]\!] = [\![\psi]\!]$ in WF implies $[\![\phi]\!] = [\![\psi]\!]$ in F. Now the result follows from the completeness part of Theorem 5.5. □

Next, we come to the question of representing the category of $\mathfrak{F}$ up to isomorphism. Intuitively, the presentation $\mathbb{F}$ plus the equations of Figure 2 present $\mathfrak{F}$ up to isomorphism once we add equations between objects identifying all words that only differ in the order of their letters. This can be made precise by adapting the notion of *presentation modulo* of Curien and Mimram [1] to partially monoidal categories:

The category presented by $\mathbb{F}$ plus the equations of Figure 2 plus equational generators [1, Def.7] identifying words that only differ in the order of their letters is isomorphic to the category $\mathfrak{F}$ of finite sets of names.

The details will have to be deferred to future work.

If we are willing to work with sets of wires instead of words of wires we obtain the following. Recall Proposition 3.12.

Remark 5.7. The partially monoidal category $\widetilde{\mathbb{F}}$ modulo equations of Figure 3 is isomorphic to the partially monoidal category $\mathfrak{F}$ of finite sets of names with functions. — The proof is not at the same level of detail as the previous ones and the corresponding normal form is less algorithmic as we rewrite here up to an equivalence relation given by the two $\equiv$-rules. Otherwise, the proof follows the same general ideas. Every map can be factored into a retract followed by a bijection followed by an inclusion. The rules in the first row of Figure 3 make sure that all lollipops can be moved to the right which gives the inclusion. The rules in the second row allow to move to the middele all diamonds not in a cup. The rules in the third row allows to eliminate all redundant diamonds in the middle. The rules in the last row capture the partions induced by a retraction.

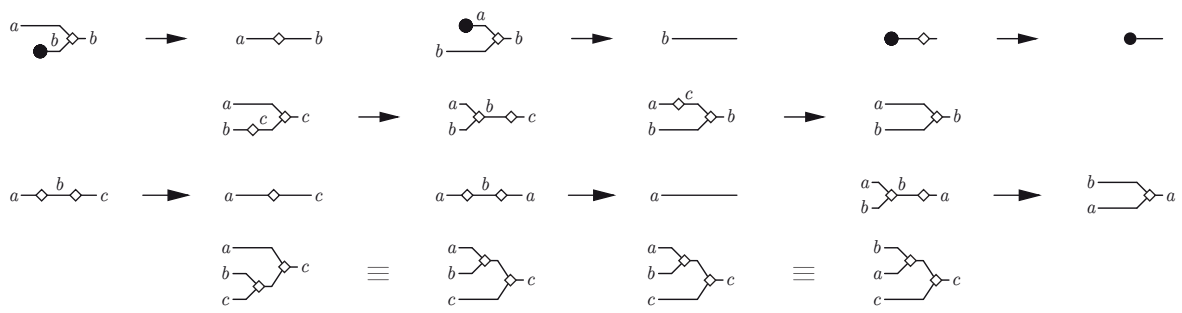


Figure 3: Rewrite rules of $\widetilde{\mathbb{F}}$

6 Software Tools

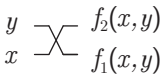
The proofs of termination and confluence presented in Section 5 were given in reduced detail as the specifics are rather technical. In order to alleviate the burden, we developed some software tools that help building these proofs and are presented briefly in this section.

6.1 Termination Proof

The termination proof, Lemma 5.2, uses a polynomial interpretation argument adapted from Lafont's original proof [7]. Whilst trying to modify the original proof (by adding the lollipop generator and the associated rules) it quickly became tedious to check if all 19 rules preserved the order. Moreover, when playing around with different possible rules, one needs a way to quickly check whether the new rules still terminate or not. Since the polynomial interpretation involves only simple arithmetic, we decided to automate the proof checking by delegating this work to an SMT solver.

As a result, we developed a Python script, which encodes the generators and all the rules as first order logic formulas and are then given to the Z3 SMT solver.

All the generators are translated into abstract functions, for example,



becomes

```
f1 = Function('f1', IntSort(), IntSort(), IntSort())
f1_def = ForAll([x, y], f1(x, y) == x+y)
f2 = Function('f2', IntSort(), IntSort(), IntSort())
f2_def = ForAll([x, y], f2(x, y) == x)
```

The rules and the monotonicity condition

are then encoded as

```
And(f1(g(),x) >= x, f2(g(),x) >= g(), Or(f1(g(),x) > x, f2(g(),x) > g()))
```

which corresponds to $f_1(g(), x) \geq x \wedge f_2(g(), x) \geq g() \wedge (f_1(g(), x) > x \vee f_2(g(), x) > g())$, encoding the condition that at least one argument is strictly decreasing, and none are increasing.

The implementation helped us to experiment with different functions for $f_1, f_2, g, \ldots$ and was instrumental in finding the solution employed in the termination proof.

6.2 Confluence proof

In order to check local confluence of our system, we decided to implement the rewrite system in Haskell. This allowed us to generate the confluence proofs automatically, by simply giving the tool the critical peak diagrams we wanted to check. For future work, we plan to also generate the critical peaks automatically and provide an argument that we have checked all the critical peaks for the given system, since, at the moment, we aren't sure whether we have indeed listed all the critical peaks.

Both the Python termination script and the implementation of the rewriting system can be found at https://github.com/goodlyrottenapple/string-diagrams-functions.

7 Conclusion

We presented a diagrammatic, two-dimensional calculus for simultaneous substitutions by presenting the category of finite sets up to isomorphism.

One motivation for this work was that the category of finite subsets of a countably infinite set Σ can be seen as a categorical semantics for simultaneous substitutions. To achieve our aims two immediate directions of further research are the following.

First, we need to extend the calculus so that one can substitute not only variables but also terms. We do expect this to be straightforward, following standard technology from categorical logic.

Second, there is still a certain gap between our calculus and more informal mathematical reasoning. In particular, the distinction between renaming and substitution is natural when working w.r.t. a specified context of a finite set of free variables (and if working with string diagrams this context is given explicitly by the wires). But in mathematical practice one does not specify explicitly the set of free variables and one may denote both δ and μ simply by $a \mapsto b$ (or $[b/a]$ in another common notation). There are two possibilities here. One consists of relaxing the shortness constraints in our calculus and to replace μ by a "cup" where all wires carry the same label. Another one, similar to nominal sets [4], is to give meaning to string diagrams as finitely supported functions $\Sigma \to \Sigma$ and to take as basic operations transposition, μ and δ.

The previous paragraph suggested to study string diagrammatic calculi for certain nominal algebras. Apart from this idea of graphical calculi representing certain nominal algebras, there is also a possibility of a nominal theory of string diagrams that are labelled with names: Some of our proofs very much have the flavour of string diagrams in nominal sets and there may well be an underlying theory waiting to be discovered.

Variations of $\mathfrak{B}$ and $\mathfrak{F}$ such as the categories of injections, surjections, partial functions, partial injections, and relations can also be characterised by theories similar to the one in Figure 2. The details still need to be worked out.

One starting point for this paper was the display calculus for first-order logic recently presented in Tzimoulis [12]. In this work, finite sets of variables play a crucial role as they

allow us to use Lawvere's observation that quantifiers are adjoints and hence satisfy the so-called display postulates, which in turn places us inside the framework of multi-type display calculi [3]. This work raised the question of an appropriate calculus not only for predicates and quantifiers but also for terms and substitutions. While we believe that a string diagrammatic calculus for terms and simultaneous substitutions is natural, the ways in which it can be integrated with a calculus of predicates and quantifiers needs further investigation.

On the category theoretic side, for a more thorough development, we should extend the results of Curien and Mimram [1] to partially monoidal categories.

Another starting point for the paper were discussions in the BI and separation logic literature about the importance of partiality of monoidal operations in the context of modelling resources, see e.g. [9, 5, 2]. This work typically builds on partial monoids, but in situations where resources come with their own notion of maps, replacing partial monoids by partially monoidal categories could be of interest.

Finally, we indicated in Section 6 that software tools provided essential help. It would be interesting to learn more about what has been developed in this direction by the string diagram community. Developing and sharing tools could play an important role for future progress.

References

[1] Pierre-Louis Curien, Samuel Mimram: Coherent Presentations of Monoidal Categories. Logical Methods in Computer Science 13(3) (2017)

[2] Brijesh Dongol, Victor B. F. Gomes, Georg Struth: A Program Construction and Verification Tool for Separation Logic. MPC 2015

[3] Sabine Frittella, Giuseppe Greco, Alexander Kurz, Alessandra Palmigiano, Vlasta Sikimic: Multi-type display calculus for dynamic epistemic logic. J. Log. Comput. 26(6) (2016)

[4] Murdoch Gabbay, Andrew M. Pitts: A New Approach to Abstract Syntax with Variable Binding. Formal Asp. Comput. 13(3-5) (2002)

[5] Didier Galmiche, Daniel Méry, David J. Pym: The semantics of BI and resource tableaux. Mathematical Structures in Computer Science 15(6) (2005)

[6] André Joyal, Ross Street: The Geometry of Tensor Calculus, I. Advances in Mathematics 88 (1991)

[7] Yves Lafont: Towards an Algebraic Theory of Boolean Circuits, Journal of Pure and Applied Algebra 184 (2-3) (2003)

[8] Saunders Mac Lane: Categories for the Working Mathematician, Second Edition, Springer (1971)

[9] Peter W. O'Hearn, David J. Pym: The logic of bunched implications. Bulletin of Symbolic Logic 5(2) (1999)

[10] Vaughan R. Pratt: Modeling concurrency with partial orders. International Journal of Parallel Programming 15(1) (1986)

[11] Peter Selinger: A survey of graphical languages for monoidal categories. Springer Lecture Notes in Physics 813 (2011)

[12] Apostolos Tzimoulis: Algebraic and Proof-Theoretic Foundations of the Logics for Social Behaviour. PhD thesis, Technical University of Delft (2018)