REWRITING OPEN OBJECTS

DANIEL CICALA

1. Introduction

The goal of this paper is to present a bicategorical framework in which to study rewriting in open networks.

By an *open network*, we mean a network together with a boundary. To make this precise, we begin with a category of 'input and output types' **C** and another category of 'networks' **D**. To equip a network, an object of **D**, with a boundary, a pair of objects from **C**, we use an adjunction

$$\begin{tikzcd} C \arrow[r, shift left, "L"] \arrow[r, phantom, "\perp"] & D \arrow[l, shift left, "R"] \end{tikzcd}$$

With this setup, we focus on three categories . The first category, denoted $\mathbf{Span}_L$, has as objects, those from **C**, and as arrows, cospans of the form

$$Lc \to d \leftarrow Lc'$$

inside of **D**.

The second category, denoted (whatever it is), has cospans

$$Lc \to d \leftarrow Lc'$$

in **D** for objects and triples of arrows (f, g, h) such that y

$$\begin{tikzcd} Lc \arrow[r] \arrow[d, "Lf"'] & d \arrow[d, "g"'] & Lc' \arrow[l] \arrow[d, "Lh"'] \\ Lc'' \arrow[r] & d' & Lc''' \arrow[l] \end{tikzcd}$$

commutes. We show that, when **C** and **D** are topoi, then so is (*insrt*).

The third category, denoted (*insert*), again has cospans

$$Lc \to d \leftarrow Lc'$$

in **D** for objects and *cubical spans of cospans*, that is commuting diagrams

$$\begin{tikzcd} Lc \arrow[r] & d & Le \arrow[l] \\ Lc' \arrow[r] \arrow[u] \arrow[d] & d' \arrow[u] \arrow[d] & Le' \arrow[l] \arrow[u] \arrow[d] \\ Lc'' \arrow[r] & d'' & Le'' \arrow[l] \end{tikzcd}$$

for arrows.

How do these three categories help us to model open networks? To answer this, we first make the observation that cospans of the form

$$Lc \to d \leftarrow Lc'$$

have showed up in each of the above categories. We call such cospans *L-open objects*. The term "open" indicates that we are thinking of d as an object that can 'interact' with certain elements. More concretely, we say that d has inputs Lc and outputs Lc' which allow d to be glued together with any other L-open object with outputs Lc or inputs Lc'. This would give us a zig-zag which we turn into an L-open object via pushout. But this is exactly the composition in *(insert)*. Hence, through their 'openness' we can think of L-open objects as arrows. This is not the only perspective we take, however.

Through the categories $LopenD$ and $LrewriteD$, we can think of L-open objects as, well, objects. Certainly, the arrows of $LopenD$ are the best candidate for a morphism of L-open objects. We show that $LopenD$ is actually a topos. Then, by work of Lack and Sobocinski, we know that L-open objects admit a nice (double pushout) rewriting theory. The sort of rewriting theory we are interested in, and that Lack and Sobocinski study, uses spans

$$\ell \to k \leftarrow r$$

to say that the object ℓ is rewritten to the object r, where k is some interface common to both ℓ and r. Translating this to the topos $LopenD$, we consider spans of L-open objects which are exactly the arrows for $LrewriteD$. Therefore, we think of $LopenD$ as the category of L-open objects with their morphisms and $LrewriteD$ as the category of L-open objects and their rewrite rules.

HERES A CHANGE

2. A motivating example

This section serves two functions. First, we discuss the example that motivates this paper. Within our discussion, we take the opportunity to set both notation and language used in the sequel.

cite

When reading network theory literature written from the compositional perspective, one comes across the notion of an open graph. The level of formality this definition is given varies between authors, but the core idea is that an *open graph* is a **Set**-diagram $E \rightrightarrows N$ together with a subset $L \subseteq N$ equipped with a partition $L = L_{\text{in}} + L_{\text{out}}$. The conceit is that the subset of nodes L is a boundary that is accessible to other open graphs. Elements of L_{in} and L_{out} are thought of as inputs and outputs, respectively. Given two open graphs $(E \rightrightarrows N, L_{\text{in}} + L_{\text{out}})$ and $(E' \rightrightarrows N', L'_{\text{in}} + L'_{\text{out}})$, such that $L_{\text{out}} = L'_{\text{in}}$, then we can construct the graph $(E + E' \rightrightarrows (N + N')/L_{\text{out}} = L'_{\text{in}}, L_{\text{in}} + L'_{\text{out}})$. For example, . We casually add that by appropriately modifying the definition of a graph morphism, one can define a morphism of open graphs.

insert diagram D1-open graph

insert diagram D2-glueing open graphs

A primary motivation behind this construction is to model the process of connecting networks together. Although, some networks contain additional information that cannot be conveyed by an open graph as described above. To accommodate such demands, we generalize the notion of an open graph to that of an *open object* and develop some basic theory for open objects.

cite examples: circ, zx-calc, petri nets, etc

One feature that distinguishes this work from other related work is our preference for reflexive graphs over directed graphs. Before mentioning our reasons, let us clarify exactly what we mean by these two sorts of graphs. Denote by $\bullet \underset{\longrightarrow}{\overset{\longrightarrow}{\longleftarrow}} \bullet$ the category with two objects $[0]$ and $[1]$ with two arrows $s, t\colon [1] \to [0]$ and an arrow $r\colon [0] \to [1]$ that is a section to both s and t. Throughout this text, the category of reflexive graphs is $\mathbf{RGraph} \coloneqq [\bullet \underset{\longrightarrow}{\overset{\longrightarrow}{\longleftarrow}} \bullet, \mathbf{Set}]$ and the category of directed graphs is $\mathbf{Graph} \coloneqq [\bullet \rightrightarrows \bullet, \mathbf{Set}]$. The reasons for working with reflexive graphs are myriad. For one, elements $1 \to \Gamma$ of a graph Γ are not just nodes, but nodes with a loop attached. It follows that the underlying nodes functor $\mathbf{RGraph} \to \mathbf{Set}$ is representable by the terminal graph. This is not the case for the underlying nodes functor of type $\mathbf{Graph} \to \mathbf{Set}$. Also, the objects of $\mathbf{RGraph}$ are truncated simplicial sets. The advantage of this goes, perhaps, beyond the scope of this paper. Suffice to say, when working with graph relations, particularly those homotopical in nature, we desire a well-known model structure to work with. Having said this, let it be known that from this point, any reference to a “graph” will mean a “reflexive graph” unless specified otherwise. This includes cases when we modify “graph” with an adjective. For instance, by “open graph” we actually mean “open reflexive graph”.

Because open graphs are the archetypal example of an open object, it behooves us to formalize that which we have so far glossed over.

Definition 2.1. Let $L\colon \mathbf{Set} \to \mathbf{RGraph}$ be the discrete graph functor. An **open graph** is a cospan of the form $Lx \to \gamma \leftarrow Ly$.

In this definition, there is a graph γ with input nodes Lx and output nodes Ly. The functor L allows us to simultaneously treat graph boundaries $Lx + Ly$ as both separate from and part of the graph.

Immediately emerging are two perspectives on open graphs, both alluded to above. The first we call *the structured cospan perspective*. From this point of view, we want to be able to glue together suitable open graphs to form new open graphs. This leads one to consider a category with sets as objects as with open graphs $Lx \to \gamma \leftarrow Ly$ as arrows of type $x \to y$. Composition uses pushouts as is typical in cospan categories. Heuristically, pushing out can be thought of as a “categorical gluing”. The second perspective is called the *open object perspective*. Here, we treat ref diagram abv?
open graphs as mathematical objects deserving of their own morphisms. Indeed, a morphism

$$(Lx \to \gamma \leftarrow Ly) \to (Lx' \to \gamma' \leftarrow Ly')$$

of open graphs is a a commuting diagram

$$\begin{tikzcd}
\partial x \arrow[r] \arrow[d, "\partial f"'] & \gamma \arrow[d, "g"'] & \partial y \arrow[l] \arrow[d, "\partial h"] \\
\partial x' \arrow[r] & \gamma' & \partial y' \arrow[l]
\end{tikzcd}$$

This leads to another category where open graphs are objects, as opposed to arrows.

Having two categories featuring open graphs—one as arrows, the other as objects—one can construct a double category containing all of this structure. This double category is defined to have sets as 0-cells, set functions as vertical 1-cells, open graphs as horizontal 1-cells and morphisms of open graphs as 2-cells. The composition functor of this double category takes advantage of the fact that open graphs are also arrows of a category. Later, we prove that this actually forms a double include D4 composition diagram category.

input appropriate section

In fact, we do this in Section BLAH. In the following section, we generalize open graphs to 'open objects' and construct a pair of categories, one with open objects as arrows and the other with open objects as objects.

3. Open objects and structured cospans

For good, we set a cocartesian geometric morphism between (elementary) topoi

$$\begin{tikzcd} \mathbf{A} \arrow[r, shift left, "L"] \arrow[r, phantom, "\perp"] & \mathbf{X} \arrow[l, shift left, "R"] \end{tikzcd}$$

Although many of the definitions and theorems work with slightly greater levels of generality, we content ourselves with this restriction. For one, it is really not much of a constraint at all. Second, it greatly simplifies the statements of various definitions and theorems.

Additionally, we will write a cospan

$$x \to y \leftarrow z$$

as a triple (x, y, z) when our work does not require us to name the arrows.

We develop a theory of "open objects" with the intention of applying it to study open networks. These are not defined precisely at this point, but suffice to say that a typical example is a sort of graph with additional information attached to its nodes and edges. The motivating example for us is the category of open graphs. There are various definitions of open graphs in the literature *(cite)*, but we define in a way we believe is more susceptible to generalization. The motivating example is to start with the adjunction $L \dashv R \colon \mathbf{Set} \to \mathbf{RGraph}$ between sets and reflexive graphs where L gives the discrete graph on the nodes of a set and R returns the points of a graph. An **open graph** is a cospan (La, g, Lb). Think of this data as a graph g with inputs La and outputs Lb. The reason for choosing reflexive graphs instead of directed graphs is primarily aesthetic. For one, we like to think of the nodes of a graph as *points*, a notion formalized in categories by arrows from the terminal object. It is therefore morally desirable for a so-called 'discrete graph' functor to be represented by the terminal set. It is also structurally desirable as well, since working with reflexive graphs instead of directed graphs provides a left exact 'discrete graph' functor.

Our first task is to provide an abstract framework in which this example organically fits.

Definition 3.1. Denote by $\mathbf{Cospan}_L$ the category with objects from $\mathbf{A}$ and whose arrows of type $a \to b$ are cospans $La \to x \leftarrow Lb$ in $\mathbf{X}$. We call these arrows L **-structured cospans**.

Our interest lies primarily in arrows from $\mathbf{Cospan}L$. We will use such cospans to encode open networks by tying the cospan feet as choosing the inputs and outputs of a network x. At times, it is helpful for us to view these as arrows in a category, but it will often be helpful to view them as objects as well. However, in behooves us to work in maximum generality. Therefore, we do not immediately bind ourselves to working with networks even though they are the chief motivation. Instead, we refer to $\mathbf{Cospan}_L$-arrows as L**-open objects**. The "L-open" indicates that objects in $\mathbf{X}$ can interact with one another via an interface determined by L. The precise interaction is described by composing cospans in the usual manner.

Definition 3.2. Denote by **Open** the category whose objects are L -structured cospans and arrows are commuting diagrams

Even though **Open** depends on L, we can safely reject L from our notation because we have no cause to consider open objects with respect to a differently named functor.

(Is OpenOb functorial? If so, we'll need to include the functor name in the notation, at least until we prove it's functorial.)

Remark 3.3. Limits and colimits in **Open** are computed pointwise. This follows from the fact that **Open** is a subcategory of $[\bullet \to \bullet \leftarrow \bullet]\,, \mathbf{X}$.

Having both $\mathbf{Cospan}_L$ and **Open** around allow us to view open objects with two different perspectives. The perspective $\mathbf{Cospan}_L$ provides a compositional one. And **Open** sees an open object as a space. The latter perspective is substantiated by the following theorem.

Theorem 3.4. *Then* **Open** *is a topos.*

(Is this function functorial? What are geometric morphisms between topoi of the form $\mathbf{Open}\partial$? Geometric natural transformations? Does this correspond to natural transformation in category of topoi with geometric morphisms?)

Because **Open** is a topos, it admits a double pushout rewriting system, the topic of our next section.

4. Double pushout rewriting

Definition 4.1. A category with pullbacks is **adhesive** if pushouts along monics exist and are *Van Kampen.*

Theorem 4.2. *Topoi are adhesive.*

Corollary 4.3. *The category* **Open** *is adhesive.*

Remark 4.4. Moving forward, we only work with topoi with the understanding that they are adhesive, hence suitable hosting double pushout rewriting. However, much of what we uncover is generalizable to the broader world of adhesive categories.

Definition 4.5. For $\mathbf{T}$ a topos, a **T-rewrite rule** (often called a production) is a span $a \leftarrow b \rightarrow c$ inside $\mathbf{T}$. When both legs of the span are monic, we say the rewrite rule is **linear**.

Definition 4.6. Given composable arrows $a \rightarrow b \rightarrow y$ we say that an arrow $a \rightarrow x$ is a **pushout complement** if it fits into a pushout diagram

$$\begin{tikzcd} a \arrow[r] \arrow[d] & b \arrow[d] \\ x \arrow[r] & y \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

Pushout complements are unique up to isomorphism when the arrow $atob$ is monic.
(cite adhesive paper)

Definition 4.7. Fix a $\mathbf{T}$ -rewrite rule $a \leftarrow b \rightarrow c$ and a $\mathbf{T}$-arrow $a \rightarrow x$ such that $b \rightarrow a \rightarrow x$ has a pushout complement. A **derived (linear) rewrite rule** is the bottom row of the induced double pushout diagram

$$\begin{tikzcd} a \arrow[d] & b \arrow[l] \arrow[r] \arrow[d] & c \arrow[d] \\ x \arrow[ur, phantom, "\urcorner", very near start] & y \arrow[l] \arrow[r] & z \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

Definition 4.8. A **grammar** consists of a topos $\mathbf{T}$ and a set of $\mathbf{T}$ -rewrite rules. The **language** $\mathcal{L}(\Gamma)$ for a grammar Γ is a set consisting of all rewrite rules derived from Γ.

Remark 4.9. Note that a language is a set of arrows from **Span**($\mathbf{T}$. This observation is put to use later on.

Remark 4.10. The notion of 'adhesive categories' does not show up in the definition of grammar or language. So, of course, one can talk about grammars in *any* category. However, working within an adhesvie category ensures we have nice properties.
(what nice properties?)

5. Non-linear rewriting

(Current goal : define double category non-linear rewriting. subgoals: define object // arrow categories)

Throughout this section, we refer to diagrams with the form

$$
\begin{tikzcd}
La \arrow[r] & x & La' \arrow[l] \\
Lb \arrow[u, "L\alpha"] \arrow[r] \arrow[d, "L\beta"'] & y \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u, "L\alpha'"'] \arrow[d, "L\beta'"] \\
Lc \arrow[r] & z & Lc' \arrow[l]
\end{tikzcd}
\tag{1}
$$

where α, α', β, and β' are isomorphisms in $\mathbf{A}$

5.1. A double category for non-linear rewriting.

Lemma 5.1. *There is a symmetric monoidal category* $(\mathbf{core}(\mathbf{Span}(\mathbf{A})), \otimes, I, \tau)$ *defined as follows:*

- $\mathbf{core}(\mathbf{Span}(\mathbf{A}))$ *is the subcategory of* $\mathbf{Span}(\mathbf{A})$ *consisting of all objects and whose arrows have invertible legs,*
- $\otimes$ *is the pointwise application of* $+$*,*
- I *is the span consisting of identities on* $0_{\mathbf{A}}$*,*
- τ *is the pointwise application of* $\tau_{\mathbf{A}}$*.*

Proof. The only non-trivial thing to check is that the interchange law holds between tensor and composition. That is, given two pairs of composable spans $a \leftarrow b \rightarrow c$, $c \leftarrow d \rightarrow e$ and $a' \leftarrow b' \rightarrow c'$, $c' \leftarrow d' \rightarrow e'$, we show that the span obtained by tensoring before composing

$$a + a' \leftarrow (b + b') \times_{c+c'} (d + d') \rightarrow e + e'$$

is equal to the span obtained by composing before tensoring

$$a + a' \leftarrow (b \times_c d) + (b' \times_{c'} d') \rightarrow e + e'.$$

In this context, equality means isomorphic as spans. But this follows from local cartesian closedness, because pullback functors are all left adjoints.

□

Lemma 5.2. *There is a symmetric monoidal preorder* $(\mathbf{P}, \otimes, I, \tau)$ *defined as follows:*

- $\mathbf{P}$ *has* L*-open objects and an arrow* $(La, x, La') \leq (Lc, x, Lc')$ *whenever there is commuting diagram of form* (1)*,*
- $\otimes$ *is given by*

$$
\begin{tikzcd}
La \arrow[r] & x & Lb \arrow[l] \\
La' \arrow[u] \arrow[r] \arrow[d] & x' \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u] \arrow[d] \\
La'' \arrow[r] & x'' & Lb'' \arrow[l]
\end{tikzcd}
\otimes
\begin{tikzcd}
Lc \arrow[r] & y & Ld \arrow[l] \\
Lc' \arrow[u] \arrow[r] \arrow[d] & y' \arrow[u] \arrow[d] & Ld' \arrow[l] \arrow[u] \arrow[d] \\
Lc'' \arrow[r] & y'' & Ld'' \arrow[l]
\end{tikzcd}
:=
\begin{tikzcd}
L(a+c) \arrow[r] & x+y & L(b+d) \arrow[l] \\
L(a'+c') \arrow[u] \arrow[r] \arrow[d] & x'+y' \arrow[u] \arrow[d] & L(b'+d') \arrow[l] \arrow[u] \arrow[d] \\
L(a''+c'') \arrow[r] & x''+y'' & L(b''+d'') \arrow[l]
\end{tikzcd}
$$

- I *is given by a pair of identities on* $L0_{\mathbf{A}}$

- *τ is given by*

$$
\begin{tikzcd}
L(a+b) \arrow[r] & x+y & L(c+d) \arrow[l] \\
L(a'+b') \arrow[u] \arrow[d] \arrow[r] & x'+y' \arrow[u] \arrow[d] & L(c'+d') \arrow[l] \arrow[u] \arrow[d] \\
L(a''+b'') \arrow[r] & x''+y'' & L(c''+d'') \arrow[l]
\end{tikzcd}
\overset{\tau}{\mapsto}
\begin{tikzcd}
L(b+a) \arrow[r] & (y+x) & L(d+c) \arrow[l] \\
L(b'+a') \arrow[u] \arrow[d] \arrow[r] & (y'+x') \arrow[u] \arrow[d] & L(d'+c') \arrow[l] \arrow[u] \arrow[d] \\
L(b''+a'') \arrow[r] & (y''+x'') & L(d''+c'') \arrow[l]
\end{tikzcd}
$$

Proof. The only non-trivial thing to check is that the tensor and composition satisfy interchange. That is, given two pairs of composable arrows

$$
\begin{tikzcd}
La \arrow[r] & v & La' \arrow[l] \\
Lb \arrow[u] \arrow[d] \arrow[r] & w \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u] \arrow[d] \\
Lc \arrow[r] & x & Lc' \arrow[l]
\end{tikzcd}
\qquad
\begin{tikzcd}
La'' \arrow[r] & v' & La''' \arrow[l] \\
Lb'' \arrow[u] \arrow[d] \arrow[r] & w' \arrow[u] \arrow[d] & Lb''' \arrow[l] \arrow[u] \arrow[d] \\
Lc'' \arrow[r] & x' & Lc''' \arrow[l]
\end{tikzcd}
$$

$$
\begin{tikzcd}
Lc \arrow[r] & x & Lc' \arrow[l] \\
Ld \arrow[u] \arrow[d] \arrow[r] & y \arrow[u] \arrow[d] & Ld' \arrow[l] \arrow[u] \arrow[d] \\
Le \arrow[r] & z & Le' \arrow[l]
\end{tikzcd}
\qquad
\begin{tikzcd}
Lc'' \arrow[r] & x' & Lc''' \arrow[l] \\
Ld'' \arrow[u] \arrow[d] \arrow[r] & y' \arrow[u] \arrow[d] & Ld''' \arrow[l] \arrow[u] \arrow[d] \\
Le'' \arrow[r] & z' & Le'''' \arrow[l]
\end{tikzcd}
$$

we want to show that the resulting arrow obtained by tensoring before composing

$$
\begin{tikzcd}
L(a+a'') \arrow[dr] & & \\
& v+v' & L(a'+a'') \arrow[l] \\
L((b+b'') \times_{(c+c'')} (d+d'')) \arrow[uu] \arrow[dd] \arrow[dr] & & \\
& (w+w') \times_{x+x'} (y+y') \arrow[uu] \arrow[dd] & L(b'+b''') \times_{(c'+c''')} (d'+d''') \arrow[l] \arrow[uu] \arrow[dd] \\
L(e+e'') \arrow[dr] & & \\
& z+z' & L(e'+e''') \arrow[l]
\end{tikzcd}
$$

is equal to the arrow obtained by composing before

$$
\begin{tikzcd}[column sep=small]
L(a+a'') \arrow[dr] & & \\
& v+v' & \\
L((b \times_c d)+(b'' \times_{c''} d'')) \arrow[uu] \arrow[dd] \arrow[dr] & & L(a'+a'') \arrow[ul] \\
& (w \times_x y)+(w' \times_{x'} y') \arrow[uu] \arrow[dd] & \\
L(e+e'') \arrow[dr] & & L((b' \times_{c'} d')+(b''' \times_{c'''} d''')) \arrow[uu] \arrow[dd] \arrow[ul] \\
& z+z' & \\
& & L(e'+e''') \arrow[ul]
\end{tikzcd}
$$

These arrows are parallel, hence equal by definition.

□

Lemma 5.3. *The preorder* $\mathbf{P}$ *is symmetric.*

Proof. Any arrow $(La, x, La') \leq (Lc, z, Lc')$ in $\mathbf{P}$ gives an arrow $(Lc, z, Lc') \leq (La, x, La')$ by taking the dual span of L-structured cospans.

□

Theorem 5.4. *There is a symmetric monoidal double category* $(\mathbb{R}\mathbf{ewrite}_L, \otimes, I, \tau)$ *consisting of the following data*

(a) object category $\mathbb{R}_0 \coloneqq \mathbf{core}(\mathbf{Span}(\mathbf{A}))$*;*
(b) arrow category $\mathbb{R}_1 \coloneqq \mathbf{P}$*;*
(c) unit functor $U \colon \mathbb{R}_0 \to \mathbb{R}_1$ *defined by*

$$
\begin{tikzcd}
a \\
b \arrow[u, "f"] \arrow[d, "g"] \\
c
\end{tikzcd}
\quad \mapsto \quad
\begin{tikzcd}
La \arrow[r, "\mathrm{id}"] & La & La \arrow[l, "\mathrm{id}"'] \\
Lb \arrow[u, "Lf"] \arrow[d, "Lg"] \arrow[r, "\mathrm{id}"] & Lb \arrow[u, "Lf"] \arrow[d, "Lg"] & Lb \arrow[l, "\mathrm{id}"'] \arrow[u, "Lf"] \arrow[d, "Lg"] \\
Lc \arrow[r, "\mathrm{id}"] & Lc & Lc \arrow[l, "\mathrm{id}"']
\end{tikzcd}
$$

(d) source and target functors $S, T \colon \mathbb{R}_1 \to \mathbb{R}_0$ *respectively defined by*

$$
\begin{tikzcd}
La \arrow[r] & x & La' \arrow[l] \\
Lb \arrow[u, "Lf"] \arrow[d, "Lg"'] \arrow[r] & y \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u] \arrow[d] \\
Lc \arrow[r] & z & Lc' \arrow[l]
\end{tikzcd}
\mapsto
\begin{tikzcd}
a \\
b \arrow[u, "f"] \arrow[d, "g"'] \\
c
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
La \arrow[r] & x & La' \arrow[l] \\
Lb \arrow[u] \arrow[d] \arrow[r] & y \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u, "Lf"'] \arrow[d, "Lg"'] \\
Lc \arrow[r] & z & Lc' \arrow[l]
\end{tikzcd}
\mapsto
\begin{tikzcd}
a' \\
b' \arrow[u, "f"] \arrow[d, "g"'] \\
c'
\end{tikzcd}
$$

(e) composition functor $\odot \colon \mathbb{R}_1 \times_{\mathbb{R}_0} \mathbb{R}_1 \to \mathbb{R}_1$ *defined by*

$$
\begin{tikzcd}
La \arrow[r] & x & La' \arrow[l] \\
Lb \arrow[u] \arrow[d] \arrow[r] & y \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u] \arrow[d] \\
Lc \arrow[r] & z & Lc' \arrow[l]
\end{tikzcd}
\odot
\begin{tikzcd}
La' \arrow[r] & x' & La'' \arrow[l] \\
Lb' \arrow[u] \arrow[d] \arrow[r] & y' \arrow[u] \arrow[d] & Lb'' \arrow[l] \arrow[u] \arrow[d] \\
Lc' \arrow[r] & z' & Lc'' \arrow[l]
\end{tikzcd}
\coloneqq
\begin{tikzcd}
La \arrow[r] & x +_{La'} x' & La'' \arrow[l] \\
Lb \arrow[u] \arrow[d] \arrow[r] & y +_{Lb'} y' \arrow[u] \arrow[d] & Lb'' \arrow[l] \arrow[u] \arrow[d] \\
Lc \arrow[r] & z +_{Lc'} z' & Lc'' \arrow[l]
\end{tikzcd}
$$

which uses pushouts in $\mathbf{X}$ *and their universal properties*

(f) The tensor $\otimes$ *is given by*

$$\begin{tikzcd} La \arrow[r] & x & Lb \arrow[l] \\ La' \arrow[u] \arrow[d] \arrow[r] & x' \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u] \arrow[d] \\ La'' \arrow[r] & x'' & Lb'' \arrow[l] \end{tikzcd} \otimes \begin{tikzcd} Lc \arrow[r] & y & Ld \arrow[l] \\ Lc' \arrow[u] \arrow[d] \arrow[r] & y' \arrow[u] \arrow[d] & Ld' \arrow[l] \arrow[u] \arrow[d] \\ Lc'' \arrow[r] & y'' & Ld'' \arrow[l] \end{tikzcd} := \begin{tikzcd} L(a+c) \arrow[r] & x+y & L(b+d) \arrow[l] \\ L(a'+c') \arrow[u] \arrow[d] \arrow[r] & x'+y' \arrow[u] \arrow[d] & L(b'+d') \arrow[l] \arrow[u] \arrow[d] \\ L(a''+c'') \arrow[r] & x''+y'' & L(b''+d'') \arrow[l] \end{tikzcd}$$

(g) a monoidal unit I *defined by*

$$I := (LI_{\mathbf{A}} \to LI_{\mathbf{A}} \leftarrow LI_{\mathbf{A}})$$

(h) and braiding τ *defined by*

$$\begin{tikzcd} L(a+b) \arrow[r] & x+y & L(c+d) \arrow[l] \\ L(a'+b') \arrow[u] \arrow[d] \arrow[r] & x'+y' \arrow[u] \arrow[d] & L(c'+d') \arrow[l] \arrow[u] \arrow[d] \\ L(a''+b'') \arrow[r] & x''+y'' & L(c''+d'') \arrow[l] \end{tikzcd} \overset{\tau}{\mapsto} \begin{tikzcd} L(b+a) \arrow[r] & (y+x) & L(d+c) \arrow[l] \\ L(b'+a') \arrow[u] \arrow[d] \arrow[r] & (y'+x') \arrow[u] \arrow[d] & L(d'+c') \arrow[l] \arrow[u] \arrow[d] \\ L(b''+a'') \arrow[r] & (y''+x'') & L(d''+c'') \arrow[l] \end{tikzcd}$$

Proof. Composition is functorial because $\mathbb{R}_1$ is a preorder. It is straightforward to check that $S;U = \mathrm{id} = T;U$ and that applying S and T to

$$\begin{tikzcd} La \arrow[r] & x & La' \arrow[l] \\ Lb \arrow[u] \arrow[d] \arrow[r] & y \arrow[u] \arrow[d] & Lb' \arrow[l] \arrow[u] \arrow[d] \\ Lc \arrow[r] & z & Lc' \arrow[l] \end{tikzcd} \odot \begin{tikzcd} La' \arrow[r] & x' & La'' \arrow[l] \\ Lb' \arrow[u] \arrow[d] \arrow[r] & y' \arrow[u] \arrow[d] & Lb'' \arrow[l] \arrow[u] \arrow[d] \\ Lc' \arrow[r] & z' & Lc'' \arrow[l] \end{tikzcd}$$

respectively returns (La, Lb, Lc) and (La'', Lb'', Lc'').

The associator, plus left and right unitors are defined using universal properties. Therefore, $\mathbb{R}\mathbf{ewrite}_L$ is a double category.

We now show that it is symmetric monoidal. For this, we follow Shulman's unpacking of Definition (blah). Lemmas **??** and **??** show that our object and arrow categories are symmetric monoidal. We have that $U(0)$ is the pair of identities on $L0$ and that the source S and target T functors are strict monoidal by construction.

citation

Next, given two pairs of composable vertical arrows

$$La \xrightarrow{f} w \xleftarrow{g} Lb \qquad Lb \xrightarrow{f'} x \xleftarrow{g'} Lc$$

$$La' \xrightarrow{h} y \xleftarrow{k} Lb' \qquad Lb' \xrightarrow{h'} z \xleftarrow{k'} Lc'$$

we construct an invertible 2-cell (denoted $\mathfrak{X}$ by Shulman) of form

$$
\begin{tikzcd}
L(a+a') \arrow[r] & (w+y) +_{L(b+b')} (x+z) & L(c+c') \arrow[l] \\
L(a+a') \arrow[r] \arrow[u] \arrow[d] & (w+y) +_{L(b+b')} (x+z) \arrow[u] \arrow[d, "\theta"'] & L(c+c') \arrow[l] \arrow[u] \arrow[d] \\
L(a+a') \arrow[r] & (w +_{Lb} x) + (y +_{Le} z) & L(c+c') \arrow[l]
\end{tikzcd}
\tag{2}
$$

The cospans along the top and bottom of (2) follow from, respectively, tensoring before composition and composing before tensoring. The map θ is constructed below. Denote the monoidal structure map by s, a canonical inclusion by ι, and a canonical quotient by q. The cospan along the top of (2) has arrows from the diagram

$$
\begin{tikzcd}
L(a+a') \arrow[r] \arrow[d, "s"'] & (w+y) +_{L(b+b')} (x+z) & L(c+c') \arrow[l] \arrow[d, "s"'] \\
La + La' \arrow[d, "f+h"'] & & Lc + Lc' \arrow[d, "g'+k'"'] \\
w+y \arrow[r, hook] & (w+y)+(x+z) \arrow[uu, two heads, "q"] & x+z \arrow[l, hook']
\end{tikzcd}
$$

and the cospan along the bottom has arrows from the diagram

$$
\begin{tikzcd}
L(a+a') \arrow[r] \arrow[d, "s"'] & (w +_{Lb} x) + (y +_{Lb'} z) & L(c+c') \arrow[l] \arrow[d, "s"'] \\
La + La' \arrow[d, "f+h"'] & & Lc + Lc' \arrow[d, "g'+k'"'] \\
w+y \arrow[r, hook, "\iota+\iota"] & (w+x)+(y+z) \arrow[uu, two heads, "q+q"] & x+z \arrow[l, hook', "\iota+\iota"']
\end{tikzcd}
$$

The arrow θ in (2) exists because of the universal property of a pushout. The diagram

$$\begin{tikzcd}
L(b+b') \arrow[rr, "s"] \arrow[d, "s"'] & & Lb+Lb' \arrow[r, "g+g'"] & w+y \arrow[d, "\iota+\iota"] \\
Lb+Lb' \arrow[d, "h+h'"'] & & & (w+x)+(y+z) \arrow[d, "q+q"] \\
x+z \arrow[r, "\iota+\iota"] & (w+x)+(y+z) \arrow[rr, "q+q"] & & (w+_{Lb}x)+(y+_{Lb'}z)
\end{tikzcd}$$

commutes because the equations $g;\iota;q = h;\iota;q$ and $g';\iota;q = h';\iota;q$ hold. Indeed, these equations are exactly those from the pushout squares of $w+_{Lb}x$ and $y+_{Lb'}z$. It follows that θ fits into diagram (**??**). Because $\mathbb{R}_1$ is a symmetric preorder (Lemma **??**), the 2-cell (**??**) is invertible as required.

Next, for objects a and b, we need an invertible 2-cell (denoted $\mathfrak{u}$ by Shulman) $U(a+b) \to Ua+Ub$. Again, to Lemma 5.3 ensures that all 2-cells are invertible. Therefore, the 2-cell

$$\begin{tikzcd}
L(a+b) \arrow[r] & L(a+b) & L(a+b) \arrow[l] \\
L(a+b) \arrow[r] \arrow[u] \arrow[d] & L(a+b) \arrow[u] \arrow[d, "s"'] & L(a+b) \arrow[l] \arrow[u] \arrow[d] \\
L(a+b) \arrow[r, "s"] & La+Lb & L(a+b) \arrow[l, "s"']
\end{tikzcd}$$

provides $\mathfrak{u}$

It remains to check that various coherence diagrams commute. Each coherence diagram lives in the arrow category $\mathbb{R}_1$ which is a preorder, so commutes automatically.

∎

Theorem 5.5. *The double category* $\mathbb{R}\mathbf{ewrite}_L$ *is fibrant.*

Proof. A companion for the vertical 1-cell $a \xrightarrow{f} b \xleftarrow{g} c$ consists of the horizontal 1-cell $La \xrightarrow{Lf^{-1}} Lb \xleftarrow{Lg^{-1}} Lc$ together with the 2-cells

$$\begin{tikzcd}
La \arrow[r, "Lf^{-1}"] & Lb & Lc \arrow[l, "Lg^{-1}"'] \\
Lb \arrow[r, "\mathrm{id}"] \arrow[u, "Lf"] \arrow[d, "Lg"'] & Lb \arrow[u, "\mathrm{id}"] \arrow[d, "Lg"'] & Lc \arrow[l, "Lg^{-1}"'] \arrow[u, "\mathrm{id}"'] \arrow[d, "\mathrm{id}"] \\
Lc \arrow[r, "\mathrm{id}"] & Lc & Lc \arrow[l, "\mathrm{id}"']
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
La \arrow[r, "\mathrm{id}"] & La & La \arrow[l, "\mathrm{id}"'] \\
La \arrow[r, "Lf"] \arrow[u, "\mathrm{id}"] \arrow[d, "\mathrm{id}"'] & Lb \arrow[u, "Lf"] \arrow[d, "\mathrm{id}"'] & Lb \arrow[l, "\mathrm{id}"'] \arrow[u, "Lf"'] \arrow[d, "Lg"] \\
La \arrow[r, "Lf"] & Lb & Lc \arrow[l, "Lg^{-1}"']
\end{tikzcd}$$

The equations hold because $\mathbb{R}\mathbf{ewrite}_L$ is locally posetal.

A conjoint for the vertical 1-cell $a \xrightarrow{f} b \xleftarrow{g} c$ consists of opposite horizontal 1-cell $Lc \xrightarrow{Lg^{-1}} Lb \xleftarrow{Lf^{-1}} La$ together with the same 2-cells as the companion. The equations hold because $\mathbb{R}\mathbf{ewrite}_L$ is locally posetal.

□

5.2. **A bicategory for non-linear rewrites.**

Corollary 5.6. *The horizontal edge bicategory* $\mathbf{Rewrite}_L \coloneqq \mathcal{H}(\mathbb{R}\mathbf{ewrite}_L)$ *in the sense of Shulman is symmetric monoidal.*

Proof. This follows from Theorem 5.1 in (shulman cite)

□

Lemma 5.7. *Every 1-arrow of* $\mathbf{Rewrite}$ *is a left and right adjoint.*

Proof. It is straightforward to check that (La, x, Lb) is both the left and right adjoint of (Lb, x, La).

□

Definition 5.8. (cite carb & walts)]

Let $\mathbf{B}$ be a bicategory whose hom-categories are posets. A **Cartesian structure** on $\mathbf{B}$ consists of a tensor product $\otimes$ on $\mathbf{B}$ and a cocommutative comonoid structure $(\delta_x, \varepsilon_x, \sigma_x)$ on every object x in $\mathbf{B}$. In addition, this data satisfies two axioms. First, every 1-arrow $f\colon x \to y$ is a lax comonoid homomorphism, that is there are 2-arrows $\delta_y f \Rightarrow (f \otimes f)\delta_x$ and $\eta_y f \Rightarrow \eta_x$. Second, comultiplication and counit have right adjoints δ_x^*, ε_x^*. A Cartesian bicategory is said to be a **bicategory of relations** if every object is a Frobenius object.

Theorem 5.9. *The bicategory* $\mathbf{Rewrite}_L$ *is a bicategory of relations in the sense of Carboni and Walters.*

Proof. We start by observing that $\mathbb{R}\mathbf{ewrite}$ is locally posetal because parallel 2-arrows are identified. The tensor product is provided in 5.6. We now show, in order, that each object has a cocommutative comonoid structure whose adjoints give a commutative monoid structure. These are compatible via the Frobenius equation. Finally, every 1-arrow is a lax comonoid homomorphism.

Given an object a in $\mathbb{R}\mathbf{ewrite}$, we use the folding map $\Delta_a\colon a + a \to a$ in $\mathbf{A}$ to define comultiplication $\delta_a\colon a \to a + a$ as the cospan

$$\delta_a\colon La \to La \xleftarrow{L\delta_a} L(a+a)$$

and use the initial map to define the counit $\varepsilon_a\colon a \to 0_a$ as the cospan $(La \to La \leftarrow L0_a)$.

The associativity and unity 2-arrows appear canonically, as does cocommutativity.

From that cocommutative comonoid structure, we obtain the commutative monoid structure by taking adjoints of all the 1-arrows (see 5.7).

The Frobenius equations are witnessed by the commuting diagram

$$\begin{tikzcd}
 & La & \\
L(a+a) \arrow[ur] \arrow[r] \arrow[dr] & La \arrow[u] \arrow[d] & L(a+a) \arrow[ul] \arrow[l] \arrow[dl] \\
 & La &
\end{tikzcd}$$

populated with arrows $L\delta_a$.

Finally, we need to check that any 1-arrow $La \xrightarrow{f} x \xleftarrow{g} Lb$ is a lax comonoid homomorphism. The lax comultiplication structure map comes from the commutating diagram

$$\begin{tikzcd}
 & x & \\
La \arrow[ur, "f"] \arrow[r] \arrow[dr, "\iota"'] & La + L(b+b) \arrow[u] \arrow[d] & L(b+b) \arrow[l] \arrow[ul, "L\delta_b ; g"'] \arrow[dl, "s; g+g"] \\
 & La +_{L(a+a)} (x+x) &
\end{tikzcd}$$

made with f, g, the monoidal structure map $s \colon L(b+b) \to Lb + Lb$ and canonical arrows. The lax unit structure map comes from the commuting diagram

$$\begin{tikzcd}
 & x & \\
La \arrow[ur, "f"] \arrow[r] \arrow[dr] & La \arrow[u, "f"] \arrow[d] & L0 \arrow[l] \arrow[ul] \arrow[dl] \\
 & La &
\end{tikzcd}$$

□

- (here's a list of facts and questions from Carb/Walt)
- (what are the maps in Rewrite)
- (what are monads and their Kleisli constructions in Rewrite?)

Corollary 5.10. *The bicategory* $\mathbf{Rewrite}_L$ *is compact closed.*

Corollary 5.11. *Freyd's modular law is satisfied:* $f(g \cap h) \Leftarrow h(g^\circ \cap s)r$.

(is$^\circ$ taking $a \to b \leftarrow c$ to $c \to b \leftarrow a$?)

Corollary 5.12. *Is it functionally complete? i.e. for each arrow* $r \colon x \to I$*, there is a map* $i \colon x_r \to x$ *such that* $i^\circ i = 1$ *and* $ti^\circ = r$*, where* t *is a canonical map* $x_r \to I$*. If so, then the subcategory of* **Rewrite** *of maps (arrows with a right adjoint) is regular.*

5.3. **Double categories from non-linear grammars.** (the below material on grammars was copied directly from the linear case and needs to be non-linearized.)

Lemma 5.13. *Fix a grammar* Γ *of non-linear rewrite rules from the topos* **Open***. Then each element of the language* $\mathcal{L}(\Gamma)$ *represents an arrow in* $\mathbf{P}$ *from 5.2. These arrows generate a symmetric monoidal sub-category of* $\mathbf{P}$ *which we denote by* $\mathbf{P}_\Gamma$*.*

(this can probably be worded and denoted better)

(does this need to be unpacked?)

Definition 5.14. Suppose each element from a grammar Γ in **Open** has form (1). Denote by $\mathbb{L}(\Gamma)$ the symmetric monoidal double subcategory of $\mathbb{M}\mathbf{on}\mathbb{R}\mathbf{ewrite}_L$ whose object category is $\mathbf{core}(\mathbf{Span}(\mathbf{A}))$ and arrow category is $\mathbf{P}_\Gamma$.

(how does the language of a grammar related to the double category?)

Theorem 5.15. *For any a pair of structured cospans* (La, x, Lb) *and* $(La', x', Lb'$ *in a language* $\mathcal{L}(\Gamma)$*, there are corresponding horizonal 1-cells in* $\mathbb{L}(\Gamma)$*. Then* (La, x, Lb) *can be written into* (La', x', Lb') *in* $\mathcal{L}(\Gamma)$ *if and only if there is a 2-cell between the corresponding horizontal 1-cells in* $\mathbb{L}(\Gamma)$*.*

6. Linear rewriting

Throughout this section, we will refer to diagrams of the form

X shd b topos for intrchng

remind linear meaning

(3)
$$\begin{tikzcd} La \arrow[r] & x & La' \arrow[l] \\ Lb \arrow[u, "L\alpha"] \arrow[r] \arrow[d, "L\beta"'] & y \arrow[u, tail] \arrow[d, tail] & Lb' \arrow[l] \arrow[u, "L\alpha'"'] \arrow[d, "L\beta'"] \\ Lc \arrow[r] & z & Lc' \arrow[l] \end{tikzcd}$$

where α, α', β, and β' are invertible. The arrows marked "$\rightarrowtail$" are monic.

6.1. A double category for linear rewriting.

Definition 6.1. Consider, again, a cocartesian category with pullbacks $(\mathbf{A}, +, 0_{\mathbf{A}}, \tau_{\mathbf{A}})$ that is locally cartesian closed and a monoidal category $(\mathbf{core}(\mathbf{Span}(\mathbf{A})), \otimes, I, \tau)$ as in Lemma **??**.

Lemma 6.2. *There is a symmetric monoidal category $(\mathbf{C}, \otimes, I, \tau)$ defined as follows:*

- *$\mathbf{C}$ has L-structured cospans (open objects?) for objects and isomorphism classes of spans open objects of form (**??**)*
- *$\otimes$ is given by*

$$\begin{tikzcd}[column sep=small] La \arrow[r] & x & Lb \arrow[l] \\ La' \arrow[u] \arrow[r] \arrow[d] & x' \arrow[u, tail] \arrow[d, tail] & Lb' \arrow[l] \arrow[u] \arrow[d] \\ La'' \arrow[r] & x'' & Lb'' \arrow[l] \end{tikzcd} \otimes \begin{tikzcd}[column sep=small] Lc \arrow[r] & y & Ld \arrow[l] \\ Lc' \arrow[u] \arrow[r] \arrow[d] & y' \arrow[u, tail] \arrow[d, tail] & Ld' \arrow[l] \arrow[u] \arrow[d] \\ Lc'' \arrow[r] & y'' & Ld'' \arrow[l] \end{tikzcd} := \begin{tikzcd}[column sep=small] L(a+c) \arrow[r] & x+y & L(b+d) \arrow[l] \\ L(a'+c') \arrow[u] \arrow[r] \arrow[d] & x'+y' \arrow[u, tail] \arrow[d, tail] & L(b'+d') \arrow[l] \arrow[u] \arrow[d] \\ L(a''+c'') \arrow[r] & x''+y'' & L(b''+d'') \arrow[l] \end{tikzcd}$$

- *I is given by a pair of identities on $L0_A$*
- *τ is given by*

$$\begin{tikzcd}[column sep=small] L(a+b) \arrow[r] & x+y & L(c+d) \arrow[l] \\ L(a'+b') \arrow[u] \arrow[r] \arrow[d] & x'+y' \arrow[u, tail] \arrow[d, tail] & L(c'+d') \arrow[l] \arrow[u] \arrow[d] \\ L(a''+b'') \arrow[r] & x''+y'' & L(c''+d'') \arrow[l] \end{tikzcd} \overset{\tau}{\mapsto} \begin{tikzcd}[column sep=small] L(b+a) \arrow[r] & (y+x) & L(d+c) \arrow[l] \\ L(b'+a') \arrow[u] \arrow[r] \arrow[d] & (y'+x') \arrow[u, tail] \arrow[d, tail] & L(d'+c') \arrow[l] \arrow[u] \arrow[d] \\ L(b''+a'') \arrow[r] & (y''+x'') & L(d''+c'') \arrow[l] \end{tikzcd}$$

Proof. Composition preserves monics because pullbacks do. Tensoring preserves monics because pushouts in adhesive categories preserve monics. Interchange holds between tensor and composition because pullback functors are left adjoints, thus preserve pushout. The remainder of the proof is a routine checking of axioms which we leave to the reader. $\square$

Lemma 6.3. *There is a symmetric monoidal double category* $(\mathbb{M}\mathbf{on}\mathbb{R}\mathbf{ewrite}_L, \otimes, I, \tau)$*. The double category* $\mathbb{M}\mathbf{on}\mathbb{R}\mathbf{ewrite}_L$ *consists of the object category* $\mathbb{M}_0 \coloneqq \mathbf{core}(\mathbf{Span}(\mathbf{A}))$*; arrow category* $\mathbb{M}_1 \coloneqq \mathbf{C}$*; unit functor* $U \colon \mathbb{M}_0 \to \mathbb{M}_1$ *defined by*

$$
\begin{tikzcd}
a & & La \arrow[r, "\mathrm{id}"] & La & La \arrow[l, "\mathrm{id}"'] \\
b \arrow[u, "f"] \arrow[d, "g"'] & \mapsto & Lb \arrow[r, "\mathrm{id}"] \arrow[u, "Lf"] \arrow[d, "Lg"] & Lb \arrow[u, tail, "Lf"] \arrow[d, tail, "Lg"] & Lb \arrow[l, "\mathrm{id}"'] \arrow[u, "Lf"] \arrow[d, "Lg"] \\
c & & Lc \arrow[r, "\mathrm{id}"] & Lc & Lc \arrow[l, "\mathrm{id}"']
\end{tikzcd}
$$

source and target functors $S, T \colon \mathbb{M}_1 \to \mathbb{M}_0$ *respectively defined by*

$$
\begin{tikzcd}
La \arrow[r] & x & La' \arrow[l] & & a \\
Lb \arrow[r] \arrow[u, "Lf"] \arrow[d, "Lg"'] & y \arrow[u, tail] \arrow[d, tail] & Lb' \arrow[l] \arrow[u] \arrow[d] & \mapsto & b \arrow[u, "f"] \arrow[d, "g"'] \\
Lc \arrow[r] & z & Lc' \arrow[l] & & c
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
La \arrow[r] & x & La' \arrow[l] & & a' \\
Lb \arrow[r] \arrow[u] \arrow[d] & y \arrow[u, tail] \arrow[d, tail] & Lb' \arrow[l] \arrow[u, "Lf"'] \arrow[d, "Lg"'] & \mapsto & b' \arrow[u, "f"] \arrow[d, "g"'] \\
Lc \arrow[r] & z & Lc' \arrow[l] & & c'
\end{tikzcd}
$$

and composition functor $\odot \colon \mathbb{M}_1 \times_{\mathbb{M}_0} \mathbb{M}_1 \to \mathbb{M}_1$ *defined by*

$$
\begin{tikzcd}
La \arrow[r] & x & La' \arrow[l] \\
Lb \arrow[r] \arrow[u] \arrow[d] & y \arrow[u, tail] \arrow[d, tail] & Lb' \arrow[l] \arrow[u] \arrow[d] \\
Lc \arrow[r] & z & Lc' \arrow[l]
\end{tikzcd}
\odot
\begin{tikzcd}
La' \arrow[r] & x' & La'' \arrow[l] \\
Lb' \arrow[r] \arrow[u] \arrow[d] & y' \arrow[u, tail] \arrow[d, tail] & Lb'' \arrow[l] \arrow[u] \arrow[d] \\
Lc' \arrow[r] & z' & Lc'' \arrow[l]
\end{tikzcd}
\coloneqq
\begin{tikzcd}
La \arrow[r] & x +_{La'} x' & La'' \arrow[l] \\
Lb \arrow[r] \arrow[u] \arrow[d] & y +_{Lb'} y' \arrow[u, tail] \arrow[d, tail] & Lb'' \arrow[l] \arrow[u] \arrow[d] \\
Lc \arrow[r] & z +_{Lc'} z' & Lc'' \arrow[l]
\end{tikzcd}
$$

which uses pushouts in $\mathbf{X}$ *and their universal properties. The tensor, monoidal unit, and braiding are given as in Lemma* **??**.

Proof. This double category is equivalent to the symmetric monoidal double category $\mathbb{M}\mathbf{onicSp}(\mathbf{Csp}(T))$ introduced in Lemma 4.4 of [1]. □

Theorem 6.4. *The double category* $\mathbb{M}\mathbf{on}\mathbb{R}\mathbf{ewrite}L$ *is isofibrant.*

Proof. See Lemma 4.5 of [1]. □

Theorem 6.5. *The horizontal edge bicategory* $\mathbf{MonRewrite}L \coloneqq \mathcal{H}\left(\mathbb{M}\mathbf{on}\mathbb{R}\mathbf{ewrite}L\right)$ *in the sense of Shulman is symmetric monoidal. Moreover, if the monoidal products* $\otimes_{\mathbf{A}}$ *and* $\otimes_{\mathbf{X}}$ *are coproducts, then the symmetric monoidal bicategory* $\mathbf{MonRewrite}L$ *is compact closed.*

Proof. This follows from Shulman [2]. □

6.2. **Double categories from linear grammars.**

Lemma 6.6. *Any rewrite rule derived from a rewrite rule of form* (**??**) *is of the same form. In particular, if all elements of a grammar* Γ *have this form, then so do the elements of the language* $\mathcal{L}(\Gamma)$.

Proof. In a topos, pushouts preserve monics and isomorphisms. □

Lemma 6.7. *Fix a grammar Γ of linear rewrite rules from the topos* **Open**, *hence of form* (**??**). *Then each element of the language $\mathcal{L}(\Gamma)$ represents an arrow in* **C** *from 6.2. These arrows generate a symmetric monoidal sub-category of* **C** *which we denote by $\mathbf{C}_\Gamma$.*

(this can probably be worded and denoted better)

Let us unpack this lemma. A linear rewrite rule inside **Open** is a span of open objects with form (**??**). This rule rewrites (La, x, Lb) into (Le, z, Lf). Take an arrow

$$(La, x, Lb) \to (La', w, Lb')$$

of open objects whose composite formed with

$$(Lc, y, Ld) \to (La, x, Lb)$$

has a pushout complement. A derived rewrite rule is the bottom face of the commuting diagram

$$\begin{tikzcd}
 & & Lb \arrow[dl] \arrow[ddd] & & Ld \arrow[ll, "L\alpha"'] \arrow[rr, "L\beta"] \arrow[dl] \arrow[ddd] & & Lf \arrow[dl] \arrow[ddd] \\
 & x \arrow[ddd] & & y \arrow[ll, tail] \arrow[rr, tail] \arrow[ddd] & & z \arrow[ddd] & \\
La \arrow[ur] \arrow[ddd] & & Lc \arrow[ll, "L\alpha'"'] \arrow[rr, "L\beta'"] \arrow[ur] \arrow[ddd] & & Le \arrow[ur] \arrow[ddd] & & \\
 & & Lb' \arrow[dl] & & Ld' \arrow[ll, "L\gamma"'] \arrow[rr, "L\delta"] \arrow[dl] & & Lf' \arrow[dl] \\
 & x' & & y' \arrow[ll, tail] \arrow[rr, tail] & & z' & \\
La' \arrow[ur] & & Lc' \arrow[ll, "L\gamma'"'] \arrow[rr, "L\delta'"] \arrow[ur] & & Le' \arrow[ur] & &
\end{tikzcd} \tag{4}$$

Note that this is a commutative diagram in a topos **X**. The monics on the bottom face of (**??**) follow from those on the top face because pushouts preserve isomorphisms and monics in topoi. Also, the pushouts *are* are in the image of L because, as a left adjoint, L preserves pushouts. Finally, the arrows $L\alpha'$, $L\beta'$, $L\gamma'$, and $L\delta'$ are invertible because isomorphisms are preserved by pushout in topoi.

(does this olny tell us that $L\alpha'$ etc are iso but not α' etc? Do we need further that isos are reflected by L?) c When we start with a collection of linear rewriting rules Γ, our language $\mathcal{L}(\Gamma)$ is also a set of linear rewriting rules. But every rewriting rule is a certain span of open objects, so we can pass from $\mathcal{L}(\Gamma)$ to the set consisting of the isoclass of each rewriting rule in $\mathcal{L}(\Gamma)$. This gives us a set of arrows in **C** with which we generate a symmetric monoidal subcategory $\mathbf{C}_\Gamma$ of Γ.

Definition 6.8. Suppose each element from a grammar Γ in **Open** is of the form (**??**). Denote by $\mathbb{L}(\Gamma)$ the symmetric monoidal double subcategory of $\mathbb{M}\mathbf{on}\mathbb{R}\mathbf{ewrite}_L$ whose object category is $\mathbf{core}(\mathbf{Span}(\mathbf{A}))$ and arrow category is $\mathbf{C}_\Gamma$.

(how does the language of a grammar related to the double category?)

Theorem 6.9. *For any a pair of structured cospans (La, x, Lb) and $(La', x', Lb'$ in a language $\mathcal{L}(\Gamma)$, there are corresponding horizonal 1-cells in $\mathbb{L}(\Gamma)$. Then (La, x, Lb)*

can be written into (La', x', Lb') *in* $\mathcal{L}(\Gamma)$ *if and only if there is a 2-cell between the corresponding horizontal 1-cells in* $\mathbb{L}(\Gamma)$.

7. Postamble

Remark 7.1. Is $\mathbf{Span}\colon \mathbf{Topos} \to \mathbf{SMC}$ given by sending T to $(\mathbf{Span}(T), +)$ where $+$ is taken pointwise a functor? How about **Cospan** ?

References

[1] D. Cicala and K. Courser, "Spans of cospans in a topos," *Theory Appl. Categ.*, vol. 33, pp. Paper No. 1, 1–22, 2018.

[2] M. Shulman, "Constructing symmetric monoidal bicategories," *arXiv preprint arXiv:1004.0993*, 2010.