REWRITING STRUCTURED COSPANS

DANIEL CICALA

1. Introduction

2. Structured cospans

Structured cospans were introduced to model compositional systems. In fact, they were the second approach to model compositional systems using cospans. Fong invented *decorated cospans* , the first approach. Structure cospans are introduced here as well as by Baez and Courser. The latter has work two aims: maximize the generality of the structured cospan construction using double categories and also to compare decorated and structured cospans. Our present interests are focused on introducing rewriting on structured cospans, hence we will make a number of restrictions that Baez and Courser do not. These restriction are harmless, however, as most cases of interest fall within the our parameters.

cite decorated cospans

cite baez-courser strcsp

In this section, we will make explicit two perspectives on structured cospans, both through the language of category theory. The first is looking at structured cospans as an object with morphisms between them. The second is as a morphism between "interfaces". It is the latter perspective that encodes the compositional structure.

For good, we fix a geometric morphism $L \dashv R \colon \mathrm{X} \to \mathrm{A}$. This is an adjunction

$$\mathrm{X} \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathrm{A}$$

between (elementary) topoi with L left exact.

Our motivations being the modeling of open systems, beginning the story by fixing a geometric morphism might signal the abstract nature of this work and obfuscate the connection between structured cospans and open systems. So that we do not drift too far from our concrete motivations, we will draw analogies between our construction and open systems throughout.

Here is the first such analogy. One should view the topos A as consisting of closed systems and their morphisms. By a *closed system*, we mean a system that cannot interact with the outside world. The topos X should be thought to contain possible interfaces for the closed systems. Equipping a closed system with an interface provides an avenue for the system to now interact with compatible elements of the outside world. The interface dictates compatibility as we explore below. Such a system is no longer closed, and so we call it an *open system*. The left adjoint L sends these interfaces into X so that they might interact with the closed systems. The right adjoint R can be though of as returning a closed system's largest possible interface. A word of caution: this is an informal analogy and should only be used

to gain an intuition for the nature of structured cospans. Now, let us turn to the first of two perspectives on structured cospans.

2.1. **Structure cospans as objects.** In this section, we define a structured cospan and the appropriate notion of morphism. The main result of the section is that the corresponding category is a topos in a functorial way.

Definition 2.1. A **structured cospan** is a cospan of the form $La \to x \leftarrow Lb$.

To typeset cospans inline, we will writing them as $y \to x \times z$ and spans as $x + z \to y$. Because all of the categories in this paper will have products and coproducts, this notation is sensible.

There is no novelty in a simple cospan, but we present this as a new definition to place ourselves into the context of systems modelling. Also, the reason for restricting ourselves to a geometric morphism at this point is elusive. Especially so because the other introducory work on structured cospans by Baez and Courser did not ask for this. Let us say that this requirement arises from practical and aesthetic considerations. For one, a nicer theory develops, but also it is a sufficiently strong condition to introduce rewriting of structured cospans.

Returning briefly to the systems analogy, a structured cospan consists of a closed system x with an interface identified by the arrows from La and Lb. Ignoring questions of causality, it is safe to consider La as the input to x and Lb as the output. The sum total of the closed system x equipped with interface $La + Lb$ is our open system. As expected, a morphism of open system ought to respect these components.

Definition 2.2. A morphism from one structured cospan $x \to La \times Lb$ to another $y \to Lc \times Ld$ is a triple of arrows (f, g, h) that fit into the commuting diagram

$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & Lb \\ {\scriptstyle Lf}\downarrow & & {\scriptstyle g}\downarrow & & {\scriptstyle Lh}\downarrow \\ Lc & \longrightarrow & y & \longleftarrow & Ld \end{array}$$

It is a simple exercise to show that structured cospans and their morphisms form a category. Denote this category by $\mathrm{StrCsp}_{\mathrm{ob}}$. This subscript reminds us that structured cospans are objects in this category. In fact, we willl refer to the objects of this category as a **structured cospan category**. While this category does depend on additional data, minimally L, we suppress this from the notation and rely on context. When wanting to be explicit, we will write $\mathrm{StrCsp}_{\mathrm{ob}}(L)$.

Example 2.3 (Open graphs)**.** The field of network theory is intimately tied with graph theory [**?**]. A natural example of a structured is an *open graph*. While this notion is not new, our infrastructure generalizes it.

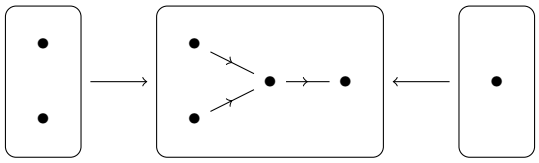
citeopengraphs

Let $\mathrm{Graph} \coloneqq [\bullet \rightrightarrows \bullet, \mathrm{Set}]$ be the category of (directed multi) graphs. There is an adjunction

$$\mathrm{Graph} \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathrm{Set}$$

where Rx is the node set of graph x and La is the edgeless graph with node set a. An **open graph** is a cospan $x \to La \times Lb$ for sets a, b, and graph x. An illustrated

example is



The boxed items are graphs and the arrows between boxes are graph morphims defined as suggested by the illustration. In total, the three graphs and two graph morphisms make up a single open graph.

Having seen this example, it becomes more apparent about how open systems can "connect" together. Given another open graph whose inputs coincide with the outputs of the graph above, we can connect the inputs and outputs together to create a new open graph. By passing from graphs to open graphs, we are introducing *compositionality*. The category $\mathrm{StrCsp}_{\mathrm{ob}}$ does not encode the compositional structure, but we introduce a new category $\mathrm{StrCsp}_{\mathrm{arr}}$ in Section 2.2 for this purpose and delay further discussion along these lines until then. At present, we concern ourselves with showing that $\mathrm{StrCsp}_{\mathrm{ob}}$ is a topos in a functorial way. While interesting in itself, this fact allows us to introduce rewriting systems to structured cospans.

Theorem 2.4. *The category* StrCsp_{ob} *is a topos.*

Proof. The category $\mathrm{StrCsp}_{\mathrm{ob}}$ constructed using the adjunction $L \dashv R \colon \mathrm{A} \to \mathrm{X}$ is equivalent to the category whose objects are cospans of form $Rx \to a \times b$ and morphisms are triples (f, g, h) fitting into the commuting diagram

$$\begin{array}{ccccc} w & \longrightarrow & Ra & \longleftarrow & x \\ {\scriptstyle f}\downarrow & & {\scriptstyle Rg}\downarrow & & {\scriptstyle h}\downarrow \\ y & \longrightarrow & Rb & \longleftarrow & z \end{array}$$

This, in turn, is equivalent to the comma category $(\mathrm{X} \downarrow \Delta R)$, where $\Delta \colon \mathrm{X} \to \mathrm{X} \times \mathrm{X}$ is the diagonal functor. But the diagonal functor is right adjoint to taking binary products. That means ΔR is also a right adjoint and, furthermore that $(X \downarrow \Delta X)$ is an instance of Artin glueing, hence a topos. CITE ARTIN GLUEING

□

Until now, the topos $\mathrm{StrCsp}_{\mathrm{ob}}$ depended on an adjunction fixed at the beginning of the section. By letting the adjunction vary, we see that $\mathrm{StrCsp}_{\mathrm{ob}}$ is actually functorial.

For the following theorem, denote by Topos the category of elementary topoi and geometric morphisms.

Theorem 2.5. *There is a functor*

$$\mathrm{StrCsp}_{ob} \colon [\bullet \to \bullet, \mathrm{Topos}] \to \mathrm{Topos}$$

defined by

$$\begin{array}{ccc} \mathrm{X} & \underset{R}{\overset{L}{\underset{\longrightarrow}{\longleftarrow}}} & \mathrm{A} \\ F \uparrow \dashv \downarrow G & & G' \downarrow \vdash \uparrow F' \\ \mathrm{X}' & \underset{L'}{\overset{R'}{\underset{\longleftarrow}{\longrightarrow}}} & \mathrm{A}' \end{array} \quad \xmapsto{\mathrm{StrCsp}_{ob}} \quad \mathrm{StrCsp}_{ob}(L) \underset{\Theta'}{\overset{\Theta}{\underset{\longleftarrow}{\longrightarrow}}} \mathrm{StrCsp}_{ob}(L')$$

which is in turn given by

$$\begin{array}{ccccc} La & \xrightarrow{m} & x & \xleftarrow{n} & Lb \\ {\scriptstyle Lf}\downarrow & & {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle Lh} \\ Lc & \xrightarrow{o} & y & \xleftarrow{p} & Ld \end{array} \quad \xmapsto{\Theta} \quad \begin{array}{ccccc} L'G'a & \xrightarrow{Gm} & Gx & \xleftarrow{Gn} & L'G'b \\ {\scriptstyle L'G'f}\downarrow & & {\scriptstyle Gg}\downarrow & & \downarrow{\scriptstyle L'G'h} \\ L'G'c & \xrightarrow{Go} & Gy & \xleftarrow{Gp} & L'G'd \end{array}$$

and

$$\begin{array}{ccccc} L'a' & \xrightarrow{m'} & x' & \xleftarrow{n'} & L'b' \\ {\scriptstyle L'f'}\downarrow & & {\scriptstyle g'}\downarrow & & \downarrow{\scriptstyle L'h'} \\ L'c' & \xrightarrow{o'} & y' & \xleftarrow{p'} & L'd' \end{array} \quad \xmapsto{\Theta'} \quad \begin{array}{ccccc} LF'a' & \xrightarrow{Fm'} & Fx' & \xleftarrow{Fn'} & LF'b' \\ {\scriptstyle LF'f'}\downarrow & & {\scriptstyle Fg'}\downarrow & & \downarrow{\scriptstyle LF'h'} \\ LF'c' & \xrightarrow{Fo'} & Fy' & \xleftarrow{Fp'} & LF'd' \end{array}$$

Proof. In light of Lemma 2.4, it suffices to show that $\Theta \dashv \Theta'$ gives a geometric morphism.

Denote the structured cospans

$$La \xrightarrow{m} x \xleftarrow{n} Lb$$

in $\mathrm{StrCsp}_{\mathrm{ob}}(L)$ by

$$L'a' \xrightarrow{m'} x' \xleftarrow{n'} L'b'$$

in $\mathrm{StrCsp}_{\mathrm{ob}}(L')$ by ℓ and ℓ', respectively. Also, denote the unit and counit for $F \dashv G$ by η, ε and for $F' \dashv G'$ by η', ε'. The assignments

$$(1) \qquad ((f,g,h)\colon \ell \to \Theta'\ell') \mapsto ((\varepsilon' \circ F'f, \varepsilon \circ Fg, \varepsilon' \circ F'h)\colon \Theta\ell \to \ell')$$

$$(2) \qquad ((f',g',h')\colon \Theta\ell \to \ell') \mapsto ((G'f' \circ \eta', Gg' \circ \eta, G'h' \circ \eta')\colon \ell \to \Theta'\ell')$$

(3)

give a bijection $\hom(\Theta\ell, \ell') \simeq \hom(\ell, \Theta'\ell')$. Moreover, it is natural in ℓ and ℓ'. This rests on the natural maps η, ε, η', and ε'. The left adjoint Θ' preserves finite limits because they are taken pointwise and L, F, and F' all preserve finite limits. □

Despite $\mathrm{StrCsp}_{\mathrm{ob}}$ being a functor, we will often let it refer to some arbitrary image of the functor. Our meaning should be clear from the context.

(I need notational fixes in this section. (1) I have StrCsp meaning three different things, a category, a functor, and...ooops. (2) I also need to define the category of structured cospan categories. I don't want to make this a 2-category if I can help it. The maps between structured cospan cats should be pairs of functors strictly commuting with the boundary and interior of the structured cospans. Also, even though the category of structured cospans is a subcat of Topos, I don't actually want the functors to be geometric morphisms. Where should I define this category?)

Even though a structured cospan category is actually a topos, and its information is drawn from a geometric morphism of topoi, the morphims between structured cospan categories that we are interested in do not involve geometric morphisms at all. Given a pair of geometric morphisms from

$$\mathrm{X} \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathrm{A} \qquad \text{and} \qquad \mathrm{X} \underset{R'}{\overset{L'}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathrm{A}'$$

a **structured cospan functor**

$$\mathrm{StrCsp}_{\mathrm{ob}}(L) \to \mathrm{StrCsp}_{\mathrm{ob}}(L')$$

consists of a a pair of functors $F\colon \mathrm{X} \to \mathrm{X}'$ and $G\colon \mathrm{A} \to \mathrm{A}'$ such that $FL = L'F$ and $GR = R'F$. Of course, structured cospan categories and their morphisms form a category, but we leave it unnamed.

2.2. **Structured cospans as morphisms.** We now turn to capturing the compositional structure that truly motivates the invention of structured cospans. To do this, we shift perspectives of structured cospans as objects in $\mathrm{StrCsp}_{\mathrm{ob}}$ to structured cospans as morphisms, which of course, are composable.

Continue to fix a geometric morphism $L \dashv R\colon \mathrm{X} \to \mathrm{A}$.

Definition 2.6. Denote by $\mathrm{StrCsp}_{\mathrm{arr}}$ the category that has the same objects as A and structured cospans $x \to La \times Lb$ as arrows of type $a \to b$.

Note that composition is defined by pushout:

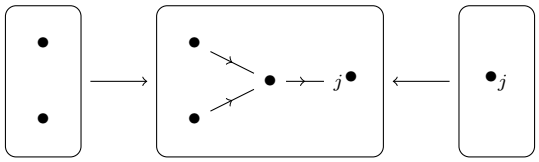
$$La \xrightarrow{\ x\ } Lb \quad ; \quad Lb \xrightarrow{\ y\ } Lc \quad \overset{\circ}{\mapsto} \quad La \xrightarrow{\ x +_{Lb} y\ } Lc$$

Pushouts, in a sense, are a way of glueing things together. Composition as using pushouts is sensible to model system connection; we have two systems

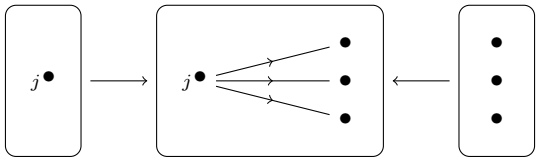
$$x \to La \times Lb \text{ and} y \to Lb \times Lc$$

whose composition is like connecting them along the common interface Lb. To illustrate how the structured cospan formalism allows us to connect systems together, we return to the open graphs example.

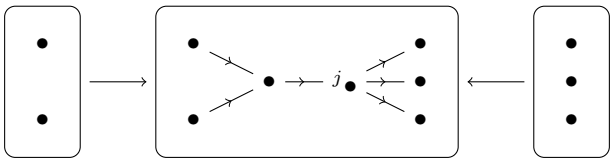
Example 2.7. The open graph



can be composed with the open graph



to obtain



This composition glued the two open graphs together along the node j.

2.3. **A bicategory of structured cospans.** Using bicategories allows us to combine into a single instrument the competing perspectives of structured cospans as objects and as morphisms. In this section, we build two bicategories: one to combine our two perspectives on structured cospans and the other, which comes in two flavors, anticipates the rewriting contained in Section 3. Many of the details of these constructions are already contained in other work, so we will focus on the ideas and point to the proofs as needed.

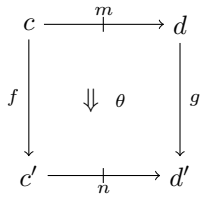
Because we the ingredients for our constructions come from a geometric morphism, there is an implicit cocartesian structure on the topoi involved. This cocartesianness can be lifted up to the bicategories, but checking all of the axiom is a tedious process. Instead, we will rely on a result of Shulman and build cocartesian double categories instead of bicategories. The relevant axioms are much easier to check. Shulman's result then allows us to chisel the cocartesian double category down to a cocartesian bicategory. The reason that we prefer bicategories to double categories is that they are more familar objects with which to work and the information lost in passing from double to bicategory, in our case, is minimal.

citeshulman-constructing

For a precise definition of a symmetric monoidal double category, we point to Shulman, though for the sake of completeness, here are the key ingredients. A double category $\mathbb{C}$ consists of a pair of categories $(\mathrm{C}_0, \mathrm{C}_1)$ with some additional data that binds them together, which we ignore. The pieces of the two categories become the following in the double category:

citeshulman-constructing

- the $\mathbb{C}$-*objects* are exactly C_0-the objects,
- the $\mathbb{C}$-*vertical arrows* $c \to d$ between C-objects are exactly the C_0 arrows,
- the $\mathbb{C}$-*horizontal arrows* $c \nrightarrow d$ between $\mathbb{C}$-objects are the C_1-objects together with some structure maps assigning the domain and codomain, and
- the *squares* of $\mathbb{C}$ are



$$c, c', d, d' \in \mathrm{ob}(\mathrm{C}_0)$$
$$f, g \in \mathrm{arr}(\mathrm{C}_0)$$
$$m, n \in \mathrm{ob}(\mathrm{C}_1)$$
$$\theta \in \mathrm{arr}(\mathrm{C}_1)$$

  are the arrows of C_1 together with structure maps attaching the surrounding vertical arrows.

The vertical arrows compose as they do in C_0 and there are structure maps that introduce a composition for the horizontal arrows. The 2-arrows can compose horizontally and vertically.

We say that a 2-arrow between identity vertical arrows is **globular**. Inside every double category $\mathbb{C}$ is a bicategory $H(\mathbb{C})$ comprised of the objects, horizontal arrows, and globular 2-arrows of $\mathbb{C}$.

Observe that the horizontal arrows play a dual role, as objects in their origin category and arrows in the double category. This reflects the dual perspective on structured cospans.

Now, let us define the first cocartesian bicategory, via a cocartesian double category.

The first double category that we define is also contained in the related work by Baez and Courser . There, they prove that it actually is a double category, so we content ourselves to simply provide the definition.

CITE COR 3.9 BAEZ COURSER STRCSP

Definition 2.8. There is a cocartesian double category $\mathbb{S}\mathrm{stCsp} \coloneqq (\mathrm{A}, \mathrm{StrCsp}_{\mathrm{ob}})$:

- the objects are all A-objects
- the vertical arrows $a \to b$ all A-arrows,
- the horizontal arrows $a \nrightarrow b$ are $\mathrm{StrCsp}_{\mathrm{ob}}$-objects, that is cospans $x \to La \times Lb$, and
- the squares are commuting diagrams

$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & Lb \\ {\scriptstyle Lf}\downarrow & & {\scriptstyle g}\downarrow & & {\scriptstyle Lh}\downarrow \\ Lc & \longrightarrow & y & \longleftarrow & Ld \end{array}$$

The tensor is pointwise application of the coproduct

$$\begin{array}{ccccc} La & \to & x & \leftarrow & Lb \\ \downarrow & & \downarrow & & \downarrow \\ Lc & \to & y & \leftarrow & Ld \end{array} \quad + \quad \begin{array}{ccccc} Lw' & \to & a' & \leftarrow & Lx' \\ \downarrow & & \downarrow & & \downarrow \\ Ly' & \to & b' & \leftarrow & Lz' \end{array} \quad \coloneqq \quad \begin{array}{ccccc} L(w+w') & \to & a+a' & \leftarrow & L(x+x') \\ \downarrow & & \downarrow & & \downarrow \\ L(y+y') & \to & b+b' & \leftarrow & L(z+z') \end{array}$$

As mentioned above, we prefer to work with symmetric monoidal bicategories and use double categories as a convenient way to arrive there. Courser has already proved that the *horizontal bicategory* of $\mathbb{S}\mathrm{stCsp}$ exists, and so we simply describe it.

CITE COURSER STR-CSP, COR 3.10

Definition 2.9. There is a symmetric monoidal bicategory **StrCsp** consisting of

- A-objects for objects,
- structured cospans $x \to La \times Lb$ as arrows of type $a \to b$, and
- maps of cospans as 2-arrows.

3. Rewriting structured cospans

At this point, we have discussed structured cospans enough to begin the transition to rewriting structured cospans. This and the next subsection introduce the bicategories that will serve as the ambient bicategories in which rewriting occurs. The appropriate style of rewriting that we on structured cospans is called double pushout (DPO) rewriting. In this section, we will briefly cover the basics of DPO rewriting and see how this translates to structured cospans.

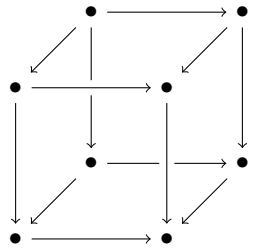
A **rewriting system**, in its most general form, is a set S together with a binary relation $\nrightarrow$. The idea is that if $s \nrightarrow r$, then s can be "replaced" by r. This is such a simple framework that, naturally, it has occured in diverse areas of study and has many different manifestations. After Ehrig, et. al. introduced graph rewriting, Lack and Sobocinski axiomatized it when they introduced *adhesive categories*, a place in which one can develop well-behaved double pushout rewriting systems. This is precisely what we plan to do with structured cospans. After a recalling the

CITE ERHIG GRAPH TRANSFORM

CITE LACK/SOBO ADHESIVE CATS

basics of DPO rewriting in adhesive categories, we introduce the rewriting theory of structured cospans.

While a rewriting system often begins with a set, we begin with an adhesive category. Adhesivity is an exactness condition that ensures certain pushouts behave as they do in Set. This is captured by sayin that the pushout satifies the **Van Kampen condition**, which states that, given a commuting cube



whose bottom face is the pushout and back faces are pullbacks, then the front faces are pullbacks if and only if the top face is a pushout.

Definition 3.1. A category with pullbacks is **adhesive** if pushouts along monics exist and are *Van Kampen*.

cite lack/sobo

An adhesive category is a place where rewrite systems retain important properties of DPO graph rewriting, namely concurrency. While the set component of a rewrite system will come from the objects of an adhesive category, the binary relation will be given by *productions*.

Definition 3.2. A **linear production** p is a span

$$\ell \hookleftarrow k \rightarrowtail r$$

with monic legs. A **left-linear production** is a span whose left leg is monic. In cases where a disinction is unimportant, we simply write "production".

The way to consider a span in this context is that we are starting with some object ℓ and replacing it with r, while k gives the part of ℓ and r that is fixed throughout the rewrite. In the style of DPO rewriting, a rewriting system is often called a grammar. Specifically a **grammar** $G \coloneqq (\mathrm{C}, P)$ consists of an adhesive category C and a set of productions P in C. The collective feet of the spans in P give us the set of objects that forms part of a rewriting system. However, this only forms part of the picture, in that a grammar is really the seed of a rich language.

Definition 3.3. There is a category Gram of consisting of

- grammars (C, P) as objects, and
- as arrows $(\mathrm{C}, P) \to (\mathrm{D}, Q)$ are functors $F\colon \mathrm{C} \to D$ such that $FP \subseteq Q$.

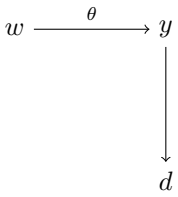
Definition 3.4. Given a grammar $G \coloneqq (\mathrm{C}, P)$ and a production $p \coloneqq (\ell \leftarrow k \rightarrow r)$ in P, we say that the production $\ell' + r' \to k'$ is **derived** from p if there is a C-arrow $m\colon \ell \to D\ell$ fitting into a *double pushout* diagram

$$\begin{array}{ccccc} \ell & \longleftarrow & k & \longrightarrow & r \\ \downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow \\ \ell' & \longleftarrow & k' & \longrightarrow & r' \end{array}$$

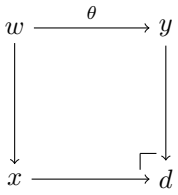
We call the arrow m a **match** of ℓ in ℓ'.

Denote by DG the grammar consisting of all productions derived from G. We will abuse notation slightly and denote by DP the set of productions derived from P. Thus, DG and (C, DP) refer to the same object in Gram.

Given a pair of arrows



that fit into a pushout



we call the object x the **pushout complement**. If θ is monic, then the pushout complement is unique up to isomorphism. This ensures that a linear or left-linear production together with a match determine a derived production up to isomorphism.

cite lack/sobo

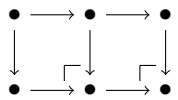
This next lemma justifies us naming D the **derivation functor**.

Lemma 3.5. *The construction* $D\colon \mathrm{Gram} \to \mathrm{Gram}$ *is functorial. Moreover*

(a) id *on* Gram *is a subfunctor of* D*,*
(b) D *is idempotent*
(c) (anything other awesom properties?)

Proof. (a) It suffices to find a monic $G \to DG$ for an arbitrary grammar $G \coloneqq (\mathrm{C}, P)$. We claim that the identity functor id_{C} gives the monic we seek. Any production is a derivation of itself via a triple of identity arrows, and so id_{C} gives an arrow $G \to DG$. It is straighforward to check that this is a monic.

(b) This is equivalent to saying that, for a set P of productions, $DP = DDP$, which follows from the fact that the outer box of the diagram



is a pushout.

□

Another important functor for us generates a category from a grammar.

Definition 3.6. Define a functor $S\colon \mathrm{Gram} \to \mathrm{Cat}$ as follows:

- for a grammar $G \coloneqq (\mathrm{C}, P)$, let SG be the subcategory of $\mathrm{Span}(\mathrm{C})$ generated by the productions in P, and

- for an arrow of grammars $F\colon (\mathrm{C}, P) \to (\mathrm{D}, Q)$, let SF be the functor given by $SFc = Fc$ and

$$(c \xleftarrow{f} d \xrightarrow{g} e) \mapsto (Fc \xleftarrow{Ff} Fd \xrightarrow{Fg} Fe)$$

In classical rewriting theory, that is when working with a set together with a binary relation, one is typically more interested in the transitive and reflexive closure of the relation. This relation gives for each element of your set, all the possible elements that one can obtain by successivly applying production rules. This is in opposition to the given binary relation which simply gives what one can obtain by a single application of a production rule. The functor $S\colon \mathrm{Gram} \to \mathrm{Cat}$ is the analog of this in our context.

define this set somewhere and find better notation

Theorem 3.7. *A grammar $G \coloneqq (\mathrm{C}, P)$ induces an binary operation $\twoheadrightarrow$ on P_{feet} by $\ell \twoheadrightarrow r$ whenever there is a cospan $\ell \leftarrow k \to r$ in P. Denote by $\twoheadrightarrow^*$ the transitive and reflexive closure of $\twoheadrightarrow$. Then $\ell \twoheadrightarrow^* r$ if and only if there is an arrow $\ell \to r$ in SG.*

Proof. If $\ell \twoheadrightarrow^* r$, then there is a zig-zag from ℓ to r comprised of spans in P or identity spans. This zig-zag gives a sequence of composable arrows in SG. Conversely, if there there is an arrow $\ell \to r$ in SG, then this arrow is a composite of generating arrows, which are exactly productions in P.

□

Definition 3.8. Define the **language** functor by $\mathcal{L} \coloneqq SD$. In particular, given a grammar G, we say that the category $\mathcal{L}G$ is the **language of** G.

At this point, the fact that categories of structured cospans are topoi (cf. Theorem 2.4) becomes relevant. In their work on adhesive categories, Lack and Sobocinski proved the following theorem.

cite topos adhesive

Theorem 3.9. *Topoi are adhesive.*

Corollary 3.10. *The category* StrCsp_{ob} *is adhesive.*

This corollary ensures that we are working within reasonable constraints to define rewriting of structured cospans. That is, we can equip a structured cospan category with a set of productions and study the resulting grammar.

Definition 3.11. Denote by StrCspGram the full subcategry of Gram consisting of grammars (C, P) such that C is a structured cospan category.

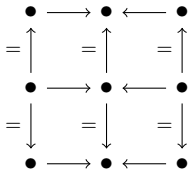
Restricting $\mathcal{L}$ along the inclusion of StrCspGram into Gram, we see that assigning the language to a grammar is functorial even if that grammar is from a structured cospan category.

To be clear, the language of a grammar of structured cospans $\mathcal{L}(\mathrm{StrCsp}_{\mathrm{ob}}(L), P)$ is a subcategory of $\mathrm{Span}(\mathrm{StrCsp}_{\mathrm{ob}}(L))$ generated by commuting diagrams of form

$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & La' \\ \uparrow & & \uparrow & & \uparrow \\ Lb & \longrightarrow & y & \longleftarrow & Lb' \\ \downarrow & & \downarrow & & \downarrow \\ Lc & \longrightarrow & z & \longleftarrow & Lc' \end{array}$$

This arrow from $x \to La \times La'$ to $z \to Lc \times Lc'$ is a composite of spans from DP.

We restrict our attention to structured cospan grammars whose productions have the form



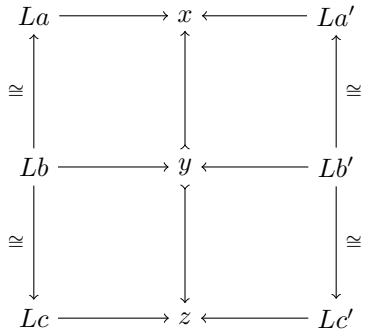
The pragmatic reason is that when rewriting open systems, which we are formalizing with structured cospans, such productions will ensure that all morphisms in the language will fix the inputs and outputs. Another reason is that we actually get a richer structure when making this restriction: the language is a bicategory, not just a category.

So far, we have discussed productions without distinguishing between linear and left-linear productions. Here is the point where this distinction becomes important, as the theory of languages for *linear grammars* diverges from the theory of *left-linear grammars*.

4. Languages for linear grammars

In this section, we pick up by studying languages for linear grammars, defined below. In Section **??**, we showed that the language of a grammar is the free category generated by arrows coming from a set of productions. Here we show that the language of a linear grammar can be structurally richer, depending on the nature of the set of productions. We begin by constructing a double category valued language functor. Then we we show that we can actually get a language that is a symmetric monoidal double category, symmetric monoidal bicategory, or compact closed bicategory by placing the right conditions on the grammars.

Definition 4.1. We say that a grammar (C, P) is *linear* if $\mathrm{C} \coloneqq \mathrm{StrCsp}(L)$ is a structured cospan category and the productions in P are of the form



Denote by $\mathrm{Gram}_{\mathrm{lin}}$ the full subcategory of Gram on the linear grammars.

There is a reasonable objection to calling such a grammar "linear" because we are restricting the adhesive category and the productions beyond simply linear productions. However, for our purposes, "linear" serves as an adequate qualifier.

To avoid a proliferation of large diagrams, we will refer denote a structured cospan by a single letter as opposed to drawing the cospan. This allows us to draw

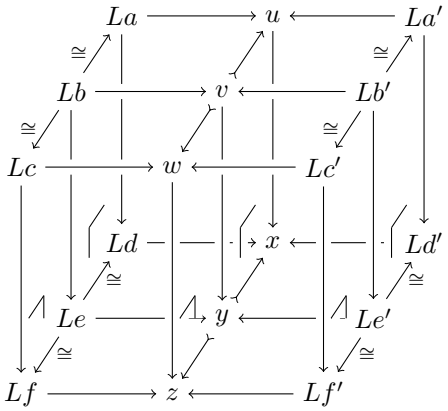
the diagram in Definition 4.1 as a span $r \leftarrow s \rightarrow t$ for structured cospans r, s, and t.

Fix a structured cospan category $\mathrm{StrCsp}_{\mathrm{ob}}(L)$. This is adhesive and so to choose a set of linear productions P of these structured cospans is to give a linear grammar $G \coloneqq (\mathrm{StrCsp}_{\mathrm{ob}}(L), P)$. In the previous section, we defined the language functor $\mathcal{L}\colon \mathrm{Gram} \to \mathrm{Cat}$ that returns the language associated to a grammar. By restricting grammars to linear grammars we obtain a linear language functor $\mathcal{L}_{\mathrm{lin}}\colon \mathrm{Gram}_{\mathrm{lin}} \to \mathrm{DblCat}$ valued in double categories. The following lemma ensures that $\mathrm{Gram}_{\mathrm{lin}}$ is a valid codomain for this functor.

cite lack-sobo

Lemma 4.2. *In an adhesive category, pushouts respect monics.*

This lemma, together with the fact that topoi are regular categories, ensure that the restriction of the derivation functor to $\mathrm{Gram}_{\mathrm{lin}}$ returns a linear grammar. Indeed, double pushout diagram in $\mathrm{StrCsp}(L)$ looks like

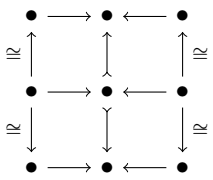


Hence, given a linear production of structured cospans, Lemma 4.2 ensures that any derived production is also linear.

To build the language, after the derivation comes generating a category, which we did in Section **??** with a functor $S\colon \mathrm{Gram} \to \mathrm{Cat}$. In this section, we define a similar functor $S_{\mathrm{lin}}\colon \mathrm{Gram}_{\mathrm{lin}} \to \mathrm{DblCat}$. Before we show how S_{lin} works, we recall a few important concepts.

Lemma 4.3. *Given a topos* T*, there is a symmetric monoidal double category* $\mathbb{M}$**onRewrite**(T) *whose*

- *objects are the* T*-objects,*
- *vertical arrows are isomorphism classes of spans with invertible legs in* T*,*
- *horizontal arrows are cospans in* T*, and*
- *squares are commuting diagrams of the form*



in T

cite spcsptopos

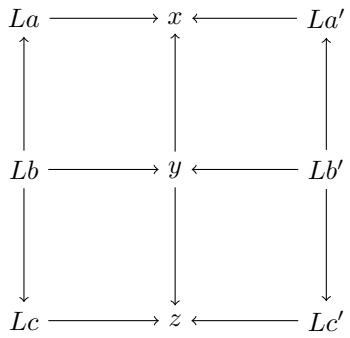
The symmetric monoidal operation is defined pointwise.

The compositions in this category follow as they do in $\mathrm{Span}(\mathrm{StrCsp}_{\mathrm{ob}}(L))$ and $\mathrm{StrCsp}_{\mathrm{arr}}(L)$.

Note that a linear production taken from a structured cospan category $\mathrm{StrCsp}_{\mathrm{ob}}(L)$ with invertible maps for legs has the correct form to be a square in $\mathbb{M}\mathbf{onRewrite}(\mathrm{X})$ where X is the domain of L. Thus a linear grammar $(\mathrm{StrCsp}_{\mathrm{ob}}(L), P)$ generates a sub-double category $\mathbb{P}$ of $\mathbb{M}\mathbf{onRewrite}(\mathrm{X})$. In fact, we can do so functorially.

Theorem 4.4. *There is a functor* $S_{lin}\colon \mathrm{Gram}_{lin} \to \mathrm{DblCat}$ *given by*

- $(\mathrm{StrCsp}_{ob}(L), P) \mapsto \mathbb{P}$,
- $F\colon (\mathrm{StrCsp}_{ob}(L), P) \to (\mathrm{StrCsp}_{ob}(L'), P')$ *is sent to the double functor* $\mathbb{P} \to \mathbb{P}'$ *defined on squares by*



Lemma 4.5. *The horizontal bicategry* **MonRewrite**(T) *of* $\mathbb{M}\mathbf{onRewrite}(\mathrm{T})$ *is compact closed.*

cite spcsptopos

5. Languages for left-linear grammars