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 write them as $x+z \to y$ and spans as $y \to x \times z$. 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 $La + Lb \to x$ to another $Lc + Ld \to y$ 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 StrCsp_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

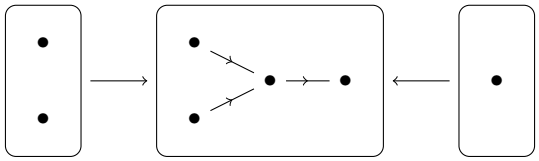
Let

$$\mathrm{RGraph} := [\, \bullet \rightrightarrows \bullet \,, \mathrm{Set}]$$

be the category of (directed reflexive multi) graphs. There is an adjunction

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

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 $La + Lb \to x$ for sets a, b, and graph x. An illustrated example, with the reflexive loops suppressed, 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 StrCsp_L does not encode the compositional structure, but we introduce a new category Csp_L in Section 2.2 for this purpose and delay further discussion along these lines until then. At present, we concern ourselves with showing that StrCsp_L 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_L *is a topos.*

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

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

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

CITE ARTIN GLUEING

Until now, the topos StrCsp_L depended on an adjunction fixed at the beginning of the section. By letting the adjunction vary, we see that $\mathrm{StrCsp}_{(-)}$ 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*

$$\mathrm{StrCsp}_{(-)} \colon [\bullet \to \bullet, \mathrm{Topos}] \to \mathrm{Topos}$$

defined by

$$\begin{array}{ccc} \mathrm{X} & \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} & \mathrm{A} \\ {\scriptstyle F}\uparrow \dashv \downarrow{\scriptstyle G} & & {\scriptstyle G'}\downarrow \vdash \uparrow{\scriptstyle F'} \\ \mathrm{X}' & \underset{L'}{\overset{R'}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} & \mathrm{A}' \end{array} \qquad \overset{\mathrm{StrCsp}_{(-)}}{\longmapsto} \qquad \mathrm{StrCsp}_L \underset{\Theta'}{\overset{\Theta}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathrm{StrCsp}_{L'}$$

which is in turn given by

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

and

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

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 StrCsp_L by ℓ and

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

in $\mathrm{StrCsp}_{L'}$ by 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) $$((f,g,h)\colon \ell \to \Theta'\ell') \mapsto ((\varepsilon' \circ F'f, \varepsilon \circ Fg, \varepsilon' \circ F'h)\colon \Theta\ell \to \ell')$$

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

(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. □

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. 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**

$$\mathrm{StrCsp}_L \to \mathrm{StrCsp}_{L'}$$

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

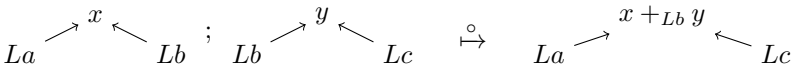
we actually only need F preserves monics, p.o.'s and p.b.'s and G p.b.'s

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 StrCsp_L 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 Csp_L the category that has the same objects as A and structured cospans $La + Lb \to x$ as arrows of type $a \to b$.

Note that composition is defined by pushout:

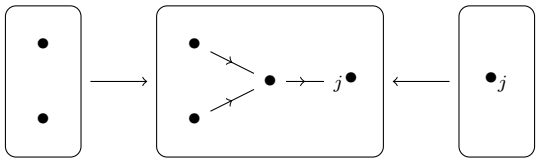


Pushouts, in a sense, are a way of glueing things together. Hence using pushouts as composition is a sensible way to model system connection. Given two systems

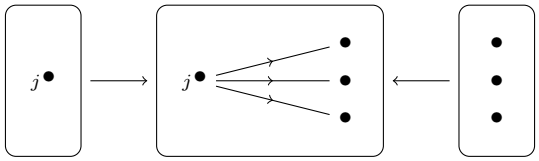
$$La + Lb \to x \quad \text{and} \quad Lb + Lc \to y$$

sharing a common interface Lb, their composition is like connecting at Lb. To illustrate how the structured cospan formalism allows us to connect systems together, we return to the open graphs example.

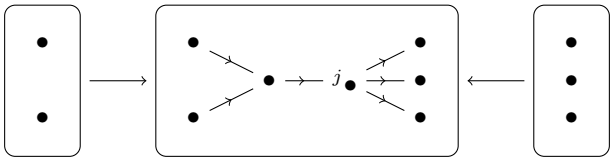
Example 2.7. The open graph



can be composed with the open graph



to obtain



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

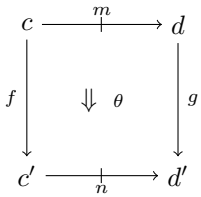
2.3. **A double category of structured cospans.** Using double categories allows us to combine into a single instrument the competing perspectives of structured cospans as objects and as morphisms. In this section, we build two double categories: one to combine our two perspectives on structured cospans and the other—which comes in two flavors—anticipates the rewriting contained in Section 3.

Because 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 double categories. For a precise definition of a symmetric monoidal double category, we point to Shulman, though for the sake of completeness, we provide the key pieces. 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. The categories C_0 and C_1 are assemble together into a double category as follows:

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 is a structure map for composing horizontal arrows. The 2-arrows can compose both horizontally and vertically.

Observe that the horizontal arrows play a two role, as objects in their origin category and arrows in the double category. This reflects the content of the categories StrCsp_L and Csp_L.

CITE COR 3.9 BAEZ COURSER STRCSP

The first double category that we define is discussed 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.

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

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

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

3. Rewriting structured cospans

At this point, we have discussed structured cospans enough to transition to rewriting structured cospans. This and the next subsection introduce the double categories that will serve as an ambience in which rewriting occurs. The appropriate style of rewriting for 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.

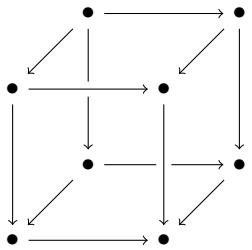
A **rewriting system**, in its most general form, is a set S together with a binary relation $\nrightarrow$. The idea is that if $s \nrightarrow r$, then s can be "replaced" by r. This is such a simple framework that, naturally, it has occured in diverse areas of study and and has many variants. After Ehrig, et. al. introduced graph rewriting, Lack and Sobocinski axiomatized it when they introduced *adhesive categories*, a place

CITE ERHIG GRAPH TRANSFORM

CITE LACK/SOBO ADHESIVE CATS

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 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*.

An adhesive category is a place where rewrite systems retain important properties of DPO graph rewriting, namely concurrency. If we were to relate this to a rewrite system as mentioned above, the set would come from the objects of an adhesive category and the binary relation from *productions*.

cite lack/sobo

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 the 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.

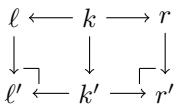
Definition 3.3. There is a category Gram of consisting of

- 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 $FP \subseteq Q$.

In Sections 3.1 and 4, we consider several variations of Gram, the so called linear and left-linear grammars, but we have no need for them yet. For now, we continue with grammars in general.

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 $k' \to \ell' \times r'$ is **derived** from p if there is a C-arrow

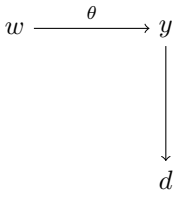
$m\colon \ell \to \ell'$ fitting into a *double pushout* diagram



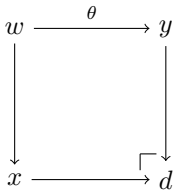
We call the arrow m a **match** of ℓ in ℓ'.

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, given a linear or left-linear production together with a match, if a pushout complement exists, the resulting derived production is unique up to isomorphism.

cite lack/sobo

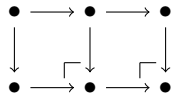
This next lemma justifies us naming D the **derivation functor**.

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.

□

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. In our context, the analogue of this is the functor $S\colon \mathrm{Gram} \to \mathrm{Cat}$, defined 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

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

Theorem 3.6. *A grammar $G \coloneqq (\mathrm{C}, P)$ induces an binary operation $\rightarrowtail$ on P_{feet} by $\ell \rightarrowtail r$ whenever there is a span $\ell \leftarrow k \to r$ in P. Denote by $\rightarrowtail^*$ the transitive and reflexive closure of $\rightarrowtail$. Then $\ell \rightarrowtail^* r$ if and only if there is an arrow $\ell \to r$ in SG.*

define this set somewhere and find better notation

Proof. If $\ell \rightarrowtail^* r$, 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 3.7. Define the **language** functor by $\mathrm{Lang} \coloneqq SD$. In particular, given a grammar G, we say that the category $\mathrm{Lang}G$ 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 3.8. *Topoi are adhesive.*

Corollary 3.9. *The category StrCsp_L 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. The language $\mathrm{Lang}(\mathrm{StrCsp}_L, P)$ of a grammar of such a grammar is a subcategory of $\mathrm{SpanStrCsp}_L$) generated by commuting diagrams of form

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

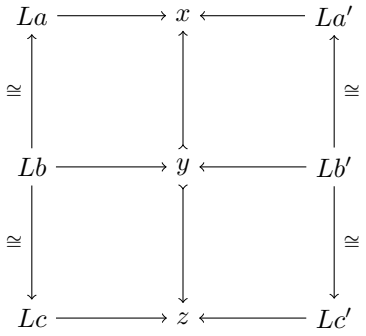
This arrow from $La + La' \to x$ to $Lc + Lc' \to z$ is a composite of spans from DP. However, this framework does not model the composition of structured cospans. In the systems analogy, we cannot connect our systems together. To do this, we categorify our construction S up to the level of a double category. Double categories support two different compositional structures, which is exactly what we

need. To do this, we need to make some minor restrictions on the productions considered. 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*.

3.1. **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 a certain subcategory generated by arrows in selected by a set of productions. Here we show that the language of a linear grammar can be structurally richer. We begin by associating a symmetric monoidal double category to each linear grammar. We show that this is a categorified version of the appropriate restriction of the language functor. The benefit of this categorification is that it contains the rewriting information, whereas the langauge functor as defined in Section 3 only retains the behavior of the rewriting system.

We begin by defining a linear grammar. Note that, here, the qualifier "linear" means more than a span with monic legs. This is to accomodate the double category structures the grammars generate, which we discuss a bit later.

Definition 3.10. The category of *linear grammars* LinGram is the subcategory of Gram on the objects (StrCsp_L, P) where P consists of productions of the form



and the morphisms are the structured cospan functors that are stable under the productions.

Another note: to avoid proliferating large diagrams, we will often denote a structured cospan by a single letter instead of drawing the cospan. This allows us to draw the diagram in Definition **??** as a diagram $r \leftarrow s \rightarrow t$, which we will understand to mean a linear production, for structured cospans r, s, and t.

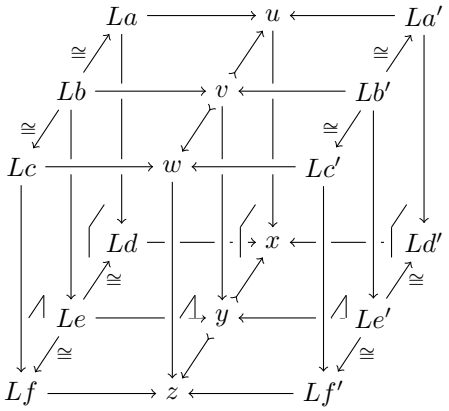
Fix a structured cospan category StrCsp_L. This is adhesive and so to choose a set of linear productions P of these structured cospans is to give a linear grammar $G := (\mathrm{StrCsp}_L, P)$. In the previous section, we defined a language functor $\mathrm{Lang}\colon \mathrm{Gram} \to \mathrm{Cat}$ that returns the language associated to a grammar. By restricting grammars to linear grammars we obtain a linear language functor $\mathrm{LinLan}\colon \mathrm{LinGram} \to \mathrm{DblCat}$, where DblCat is the category of double categories and double functors. We now commence to constructing this functor.

cite lack-sobo

Recall in the previous section that the first step to construct the language functor was to find the derived grammar $D\colon \mathrm{Gram} \to \mathrm{Gram}$. The following lemma ensures that we can restrict this functor to the linear grammar $D_{\mathrm{lin}}\colon \mathrm{LinGram} \to \mathrm{LinGram}$.

Lemma 3.11. *In an adhesive category, pushouts respect monics.*

This lemma together with the fact that topoi are regular categories, ensure that D is stable under production linearity. Observe: a double pushout diagram in StrCsp_L looks like

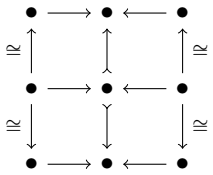


Hence, Lemma 3.11 ensures that a derived linear production is linear.

To build the language, after the derivation comes generating a category, which we did in Section **??** with a functor $S\colon \mathrm{Gram} \to \mathrm{Cat}$. The situation with linear grammars is more interesting. The compositionality inherent in linear grammars endows the language with a richer structure. We capture this by defining a functor $S_{\mathrm{lin}}\colon \mathrm{LinGram} \to \mathrm{DblCat}$, though this take some preparation. We begin by introducing a class of double categories that will serve as ambience in which we generate sub-double categories.

Lemma 3.12. *Let $L \dashv R\colon \mathrm{X} \to \mathrm{A}$ be a geometric morphism. There is a double category $\mathbb{M}\mathbf{onRewrite}(\mathrm{StrCsp}_L) \coloneqq (\mathbb{M}_0, \mathbb{M}_1)$ comprised of the categories*

- *$\mathbb{M}_0$ with objects from A and arrows are spans in A with invertible legs, and*
- *$\mathbb{M}_1$ with objects L-structured cospans and arrows isoclasses of linear productions, that is commuting diagrams of form*



(include compositions, structure maps, proof,relation to baez-courser thm 3.1)

have i defined linear productions as morphisms?

On objects, S_{lin} sends a grammar (StrCsp_L, P) to the sub-double category $\mathbb{P} \coloneqq (\mathbb{P}_0, \mathbb{P}_1)$ of $\mathbb{M}\mathbf{onRewrite}(\mathrm{StrCsp}_L)$ where $\mathbb{P}_0$ is generated by

- objects, those of A that occupy a corner of some linear production in P;
- arrows, those spans in A that occupy a vertical edge of some linear production in P;

and $\mathbb{P}_1$ is the replete category generated

- by horizontal arrows, those L-structured cospans that occupy a horizonal edge of some linear production in P; and
- by squares, the productions in P

Fix geometric morphisms $L \dashv R\colon \mathrm{X} \to \mathrm{A}$ and $L' \dashv R'\colon \mathrm{X}' \to \mathrm{A}'$. The image of an arrow

$$(F\colon \mathrm{X} \to \mathrm{X}', G\colon \mathrm{A} \to \mathrm{A}')\colon (\mathrm{StrCsp}_L, P) \to (\mathrm{StrCsp}_{L'}, Q)$$

under S_{lin} is the double functor $\mathbb{P} \to \mathbb{Q}$ given

- on objects by $a \mapsto Ga$
- on vertical arrows by

$$(La \xleftarrow{Lf} Lb \xrightarrow{Lg} Lc) \mapsto (L'Ga \xleftarrow{L'Gf} L'Gb \xrightarrow{L'Gg} L'Gc)$$

- on horizontal arrows by

$$(La \xleftarrow{h} x \xrightarrow{k} Lb) \mapsto (L'Ga \xleftarrow{Fh} Fx \xrightarrow{Fk} L'Gb)$$

- on squares by

$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & La' \\ \cong\uparrow & & \uparrow & & \uparrow\cong \\ Lb & \longrightarrow & y & \longleftarrow & Lb' \\ \cong\downarrow & & \downarrow & & \downarrow\cong \\ Lc & \longrightarrow & z & \longleftarrow & Lc' \end{array} \quad \mapsto \quad \begin{array}{ccccc} L'Ga & \longrightarrow & Fx & \longleftarrow & L'Ga' \\ \cong\uparrow & & \uparrow & & \uparrow\cong \\ L'Gb & \longrightarrow & Fy & \longleftarrow & L'Gb' \\ \cong\downarrow & & \downarrow & & \downarrow\cong \\ L'Gc & \longrightarrow & Fz & \longleftarrow & L'Gc' \end{array}$$

That this actually defines a double functor requires checking a number of axioms, which we place in an appendix.

4. Languages for left-linear grammars

The category of *left linear grammars* LeftLinGram is the full subcategory of Gram on objects of the form (StrCsp_L, P) where P consists of only left linear grammars

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

Appendix A. S_{LIN} is functorial

Fix a pair of adjunctions

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

To show that $S_{\mathrm{lin}}\colon \mathrm{LinGram} \to \mathrm{DblCat}$ is a functor, we first generate from a linear grammar arrow $(F, G)\colon (\mathrm{StrCsp}(L), P) \to (\mathrm{StrCsp}(L'), Q)$ a double functor $S_{\mathrm{lin}}(F, G)\colon \mathbb{P} \to \mathbb{Q}$ that we denote by S for short. To show that S is a double funtor, we use the definition presented in which lists the following requirements:

these were defined above, right?

cite shulman double whatever derived double functors

(a) Functions from the objects

$$a \mapsto Fa,$$

vertical arrows

$$(a \xleftarrow{f} b \xrightarrow{g} c) \mapsto (Ga \xleftarrow{Ff} Gb \xrightarrow{Fg} Gc),$$

horizontal arrows

$$(La \xrightarrow{k} x \xleftarrow{h} Lb) \mapsto (FLa \xrightarrow{Fk} Fx \xleftarrow{Fh} FLb) = (L'Ga \xrightarrow{Fk} Fx \xleftarrow{Fh} L'Gb),$$

and squares

$$\begin{array}{ccccc} La & \xrightarrow{f''} & x & \xleftarrow{g''} & La' \\ {\scriptstyle\cong}\uparrow{\scriptstyle Lh} & & \uparrow{\scriptstyle h'} & & {\scriptstyle Lh''}\uparrow{\scriptstyle\cong} \\ Lb & \xrightarrow{f'} & y & \xleftarrow{g'} & Lb' \\ {\scriptstyle\cong}\downarrow{\scriptstyle Lk} & & \downarrow{\scriptstyle k'} & & {\scriptstyle Lk''}\downarrow{\scriptstyle\cong} \\ Lc & \xrightarrow[f]{} & z & \xleftarrow[g]{} & Lc' \end{array} \quad \mapsto \quad \begin{array}{ccccc} L'Ga & \xrightarrow{Ff''} & Fx & \xleftarrow{Fg''} & L'Ga' \\ {\scriptstyle\cong}\uparrow{\scriptstyle L'Gh} & & \uparrow{\scriptstyle Fh'} & & {\scriptstyle L'Gh''}\uparrow{\scriptstyle\cong} \\ L'Gb & \xrightarrow{Ff'} & Fy & \xleftarrow{Fg'} & L'Gb' \\ {\scriptstyle\cong}\downarrow{\scriptstyle Fk} & & \downarrow{\scriptstyle Fk'} & & {\scriptstyle L'Gk''}\downarrow{\scriptstyle\cong} \\ L'Gc & \xrightarrow[Ff]{} & Fz & \xleftarrow[Fg]{} & L'Gc' \end{array}$$

(b) For each object a of $\mathbb{P}$, a pair of squares

$$\begin{array}{ccc} Sa & \xrightarrow{S\,\mathrm{id}} & Sa \\ {\scriptstyle\mathrm{id}}\downarrow & \Downarrow S_a & \downarrow{\scriptstyle\mathrm{id}} \\ Sa & \xrightarrow[\mathrm{id}]{} & Sa \end{array} \quad \text{and} \quad \begin{array}{ccc} Sa & \xrightarrow{\mathrm{id}} & Sa \\ {\scriptstyle\mathrm{id}}\downarrow & \Downarrow S^a & \downarrow{\scriptstyle S\,\mathrm{id}} \\ Sa & \xrightarrow[\mathrm{id}]{} & Sa \end{array}$$

in $\mathbb{Q}$ both of which are

$$\begin{array}{ccccc} L'Ga & \longrightarrow & L'Ga & \longleftarrow & L'Ga \\ \uparrow & & \uparrow & & \uparrow \\ L'Ga & \longrightarrow & L'Ga & \longleftarrow & L'Ga \\ \downarrow & & \downarrow & & \downarrow \\ L'Ga & \longrightarrow & L'Ga & \longleftarrow & L'Ga \end{array}$$

filled with identity arrows.

(c) For each composable pair

$$f\colon b \to a \times c \quad \text{and } g\colon d \to c \times e$$

of vertical arrows in $\mathbb{P}$, an invertible square

$$\begin{array}{ccc} Sa & \xrightarrow{\mathrm{id}} & Sa \\ {\scriptstyle Sf}\downarrow & & \\ Sb & \Downarrow S^{gf} & \downarrow{\scriptstyle Sgf} \\ {\scriptstyle Sg}\downarrow & & \\ Sc & \xrightarrow[\mathrm{id}]{} & Sc \end{array} \;\coloneqq\; \begin{array}{ccccc} L'Ga & \longrightarrow & L'Ga & \longleftarrow & L'Ga \\ \uparrow & & \uparrow & & \uparrow \\ L'(Gb +_{Gc} Gd) & \to & L'Gb +_{L'Gc} L'Gd & \longleftarrow & L'G(b +_c d) \\ \downarrow & & \downarrow & & \downarrow \\ L'Ge & \longrightarrow & L'Ge & \longleftarrow & L'Ge \end{array}$$

in $\mathbb{Q}$. The arrows to the pushout in the center of the diagram are the isomorphisms that arise from L' and G both preserving pushouts.

(d) For each composable pair

$$h\colon La + Lb \to x \quad \text{and} \quad k\colon Lb + Lc \to y$$

of horizontal arrows in $\mathbb{P}$, an invertible square

$$\begin{array}{ccc} Sa & \xrightarrow{Skh} & Sc \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow S_{kh} & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \underset{Sh}{\to} Sb \underset{Sk}{\to} & Sc \end{array} \coloneqq \begin{array}{ccccc} L'Ga & \to & F(x \times_{Lb} y) & \leftarrow & L'Gc \\ \uparrow & & \uparrow & & \uparrow \\ L'Ga & \to & Fx \times_{FLb} Fy & \leftarrow & L'Gc \\ \downarrow & & \downarrow & & \downarrow \\ L'Ga & \to & Fx \times_{L'Gb} Fy & \leftarrow & L'Gc \end{array}$$

where the arrows from the pullback in center of the diagram are the canoncial isomorphisms that follow from the properties that F preserves pullbacks and $FL = L'G$.

(e) The following coherence axioms hold for vertical arrows $f\colon a \to b$, $g\colon b \to c$, and $h\colon c \to d$ in $\mathbb{P}$

$$\begin{array}{ccccc} Sa & \overset{\mathrm{id}}{\to} & Sa & \overset{\mathrm{id}}{\to} & Sa \\ {\scriptstyle Sf}\downarrow & & \downarrow{\scriptstyle Sgf} & & \\ Sb & \Downarrow_{Sgf} & & & \\ {\scriptstyle Sf}\downarrow & & \Downarrow_{S^{h(gf)}} & & {\scriptstyle Shgf} \\ Sc & \overset{\mathrm{id}}{\to} & Sc & & \\ {\scriptstyle Sh}\downarrow & \Downarrow_{\mathrm{id}} & \downarrow{\scriptstyle Sh} & & \\ Sd & \underset{\mathrm{id}}{\to} & Sd & \underset{\mathrm{id}}{\to} & Sd \end{array} = \begin{array}{ccccc} Sa & \overset{\mathrm{id}}{\to} & Sa & \overset{\mathrm{id}}{\to} & Sa \\ {\scriptstyle Sf}\downarrow & \Downarrow_{\mathrm{id}} & \downarrow{\scriptstyle Sgf} & & \\ Sb & \overset{\mathrm{id}}{\to} & Sb & & \\ {\scriptstyle Sf}\downarrow & & \Downarrow_{S^{h(gf)}} & & {\scriptstyle Shgf} \\ Sc & \Downarrow_{Sgf} & {\scriptstyle Sh}\downarrow & & \\ {\scriptstyle Sh}\downarrow & & \downarrow & & \\ Sd & \underset{\mathrm{id}}{\to} & Sd & \underset{\mathrm{id}}{\to} & Sd \end{array}$$

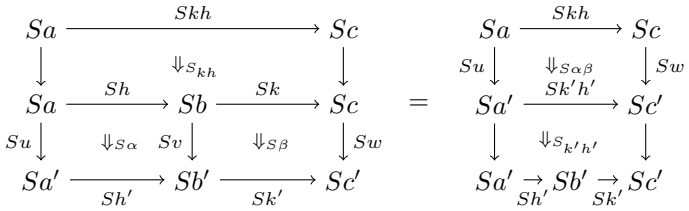
$$\begin{array}{ccc} Sa & \overset{\mathrm{id}}{\to} & Sa \\ {\scriptstyle Sf}\downarrow & \Downarrow_{\mathrm{id}} & \downarrow{\scriptstyle Sf} \\ Sb & \overset{\mathrm{id}}{\to} & Sb \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{Sb} & \downarrow{\scriptstyle S\,\mathrm{id}} \\ Sb & \underset{\mathrm{id}}{\to} & Sb \end{array} = \begin{array}{ccc} Sa & \overset{\mathrm{id}}{\to} & Sa \\ {\scriptstyle Sf}\downarrow & \Downarrow_{S^{\mathrm{id}\,f}} & \\ Sb & & {\scriptstyle Sf}\downarrow \\ {\scriptstyle S\,\mathrm{id}}\downarrow & & \\ Sb & \underset{\mathrm{id}}{\to} & Sb \end{array} \qquad \begin{array}{ccc} Sa & \overset{\mathrm{id}}{\to} & Sa \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{S^a} & \downarrow{\scriptstyle S\,\mathrm{id}} \\ Sa & \overset{\mathrm{id}}{\to} & Sa \\ {\scriptstyle Sf}\downarrow & \Downarrow_{\mathrm{id}} & \downarrow{\scriptstyle Sf} \\ Sb & \underset{\mathrm{id}}{\to} & Sb \end{array} = \begin{array}{ccc} Sa & \overset{\mathrm{id}}{\to} & Sa \\ {\scriptstyle S\,\mathrm{id}}\downarrow & \Downarrow_{S^{f\,\mathrm{id}}} & \\ Sa & & {\scriptstyle Sf}\downarrow \\ {\scriptstyle Sf}\downarrow & & \\ Sb & \underset{\mathrm{id}}{\to} & Sb \end{array}$$

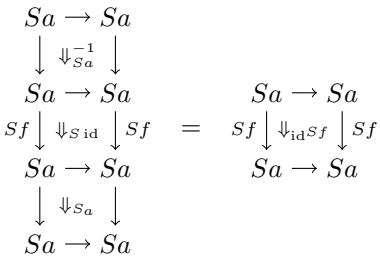
(f) For horizonal arrows $f\colon a \to b$, $g\colon b \to c$, and $h\colon c \to d$, the following coherence conditions

$$\begin{array}{ccccc} Sa & & \xrightarrow{\quad Shgf \quad} & & Sd \\ {\scriptstyle \mathrm{id}}\downarrow & & \Downarrow_{S_{h(gh)}} & & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \xrightarrow{Sgf} & Sc & \xrightarrow{Sh} & Sd \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{S_{gf}} & {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{\mathrm{id}} & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \underset{Sf}{\to} & Sb & \underset{Sg}{\to} Sc \underset{Sh}{\to} & Sd \end{array} = \begin{array}{ccccc} Sa & & \xrightarrow{\quad Shgf \quad} & & Sc \\ {\scriptstyle \mathrm{id}}\downarrow & & \Downarrow_{S_{(hg)f}} & & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \overset{Sf}{\to} & Sb & \xrightarrow{\quad Shg \quad} & Sd \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{\mathrm{id}} & {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{S_{hg}} & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \underset{Sf}{\to} & Sb & \xrightarrow[\quad Sg \quad]{} & Sd \end{array}$$

$$\begin{array}{ccccc} Sa & \overset{S\,\mathrm{id}}{\to} & Sa & \overset{Sf}{\to} & Sb \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{S\,\mathrm{id}} & {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{\mathrm{id}} & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \underset{\mathrm{id}}{\to} & Sa & \underset{Sf}{\to} & Sb \end{array} = \begin{array}{ccc} Sa & \xrightarrow{Sf\,\mathrm{id}} & Sb \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{S_{f\,\mathrm{id}}} & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \underset{\mathrm{id}}{\to} Sa \underset{Sf}{\to} & Sb \end{array} \qquad \begin{array}{ccccc} Sa & \overset{Sf}{\to} & Sb & \overset{S\,\mathrm{id}}{\to} & Sb \\ {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{\mathrm{id}} & {\scriptstyle \mathrm{id}}\downarrow & \Downarrow_{b} & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \underset{Sf}{\to} & Sb & \underset{\mathrm{id}}{\to} & Sb \end{array} = \begin{array}{ccc} Sa & \xrightarrow{S\,\mathrm{id}\,f} & Sb \\ {\scriptstyle \mathrm{id}}\downarrow & S_{\mathrm{id}\,f} & \downarrow{\scriptstyle \mathrm{id}} \\ Sa & \underset{Sf}{\to} Sb \underset{\mathrm{id}}{\to} & Sb \end{array}$$

(g) The double naturality equations





hold for horizontal arrows and the transposes for vertical arrows.

References