# REWRITING STRUCTURED COSPANS

DANIEL CICALA

Abstract. To foster the study of networks on an abstract level, we introduce the formalism of *structured cospans*. A structured cospan is a diagram of the form $La \to x \leftarrow Lb$ built from a geometric morphism with left exact left adjoint $L \dashv R \colon \mathsf{X} \to \mathsf{A}$. We show that this construction is functorial and results in a topos with structured cospans as objects. Additionally, structured cospans themselves are compositional. Combining these two perspectives, we define a double category of structured cospans. We then leverage adhesive categories to create a theory of rewriting for structured cospans. A well-known result of graph rewriting is that a graph grammar induces the same rewrite relation as its underlying graph grammar. We generalize this result to topoi under the assumption that the subobject algebra on each context in the grammar is well-founded. This fact is used to provide a compositional framework for double pushout rewriting in a topos $\mathsf{X}$ that is the domain of a geometric morphism.

## 1. Introduction

This paper fits into a program interested in categorifying the study of compositional systems. Part of the motivation for this program is the desire to understand global behavior of systems through analyzing local components. To this end, we are introducing a sytactical device, structured cospans, to reason about open systems.

In the study of formal languages, often accompanying a syntax is a rewriting system. This is a set of rules dictating when one may replace one syntactical term for another. Each rewriting system gives rise to a rewriting relation, which is useful in situations where distinct syntactical terms have the same semantics, or meaning. An example of this arises in electrical circuits when a pair of resistors are wired in series. This has the same behavior as a single resistor whose resistance is the sum of the two original resistors. Syntacitally, these are different components, but we want to treat them the same. So an appropriate rewriting system would want to relate (not equate) these two circuits. In this paper, we introduce the theory of rewriting structured cospans.

The theory of rewriting has gone through several epochs, each more abstract from the last. Rewriting was introduced by Chomsky for formal languages. Here, the syntactical terms in consideration were strings of characters, or letters. Later, Ehrig, et. al. [**?**] used pushouts to introduce rewriting for graphs. Graph rewriting was then axiomatised by Lack and Sobociński when they defined adhesive categories [15]. While we do not need the full generality of adhesive categories, we do use rewriting of topoi, which are examples of adhesive categries [16].

We define a *grammar* to be a topos $\mathsf{X}$ paired with a set of rules $P$. Each rule in $P$ is a span $\ell \leftarrowtail k \rightarrowtail r$ in $\mathsf{X}$ such that each leg is monic. The conceipt is that the object $\ell$ can be replaced by $r$ while the common subobject $k$ is fixed. The grammar $(\mathsf{X}, P)$ induces a *derived grammar* $(\mathsf{X}, P')$ where $P'$ is comprised of any rules appearing on the bottom row of a double pushout diagram of form

$$\begin{tikzcd} \ell \arrow[d, tail] \arrow[dr, phantom, "\text{p.o.}"] & k \arrow[l, tail] \arrow[r, tail] \arrow[d, tail] \arrow[dr, phantom, "\text{p.o.}"] & r \arrow[d, tail] \\ g & d \arrow[l, tail] \arrow[r, tail] & h \end{tikzcd}$$

whose top row belongs to $P$. The study of the grammar $(\mathsf{X}, P)$ is really done by understanding the *rewrite relation* $g \rightsquigarrow^* h$ defined by, first, relating objects if there is a rule in $P'$ between them, then taking the reflexive and transitive closure.

In practice, we take $\mathsf{X}$ to be a topos which we consider to be comprised of systems together with the appropriate morphisms. Because of the important status that graphs play in the field of network theory, the archetypal topos for us is the category $\mathsf{RGraph}$ of reflexive graphs. Given a rewriting system $(\mathsf{X}, P)$, our goal is to see how we can rewrite systems $x$ by somehow decomposing it into subsystems, rewriting those, then glueing the results back together. We claim that, using structured cospans, we can do this in a way that characterizes the rerwriting relation for $(\mathsf{X}, P)$.

To accomplish this, we turn to Gadducci and Heckle [13] who introduced an inductive perspective for graph rewriting. This is closely related to our present goals so, while we work in a more general context and have different motivations than they do, we follow the framework set in their paper. In particular, they define so called "ranked graphs" which are directed graphs with a chosen set of nodes serving as inputs and outputs. Using structured cospans, we are able to provide inputs and outputs to a much wider class of objects than graphs.

To do so, we need addition data besides just $\mathsf{X}$. We begin with a geometric morphism

$$(L \colon \mathsf{A} \to \mathsf{X}) \dashv (R \colon \mathsf{X} \to \mathsf{A}),$$

which, recall, is an adjunction between topoi whose left adjoint preserves finite limits. In this setup, $\mathsf{X}$ still is our topos of systems. The new data consists of $\mathsf{A}$ which is a topos comprised of "interface types". These will serve as the boundaries along which we decompose the systems in $\mathsf{X}$. The left adjoint $L$ serves as the channel through which we port the interface types into $\mathsf{X}$ so that they can interact with the systems. The fact that this data comes in the form of a geometric morphism is used throughout this paper.

A *structured cospan* is a cospan in $\mathsf{X}$ of the form $La \to x \leftarrow Lb$. This should be thought of as consisting of a system $x$ with inputs $a$ and outputs $b$.

Structured cospans were introduced by Baez and Courser [**?**], though under weaker hypothesis than we have here. We ask for stronger assumptions in order to introduce rewriting, which they do not consider. Their primary focus was on the compositional structure, which is captured as follows. A structured cospan $La \to x \leftarrow Lb$ with outputs $b$ can be connected together with a structured cospan $Lb \to y \leftarrow Lc$ with inputs $b$ via pushout

$$\underbrace{\begin{tikzcd}[column sep=small] & x & & & y & \\ La \arrow[ur] & & Lb \arrow[ul] & Lb \arrow[ur] & & Lc \arrow[ul] \end{tikzcd}}_{\text{open sub-systems}} \xrightarrow{\text{connection}} \underbrace{\begin{tikzcd}[column sep=small] & x +_{Lb} y & \\ La \arrow[ur] & & Lc \arrow[ul] \end{tikzcd}}_{\text{composite system}}$$

This gives us a category $\mathsf{Cospan}_L$ whose objects are from $\mathsf{A}$ and whose arrows are the structured cospans.

Observe that a system $x$ without inputs and outputs can be encoded using the structured cospan $L0 \to x \leftarrow L0$. Studying $x$ locally means finding a decomposition

$$L0 \to x_1 \leftarrow Lb_1 \to x_2 \leftarrow Lb_2 \cdots Lb_{n-1} \to x_n \leftarrow L0$$

and individually looking at each factor. This is exactly what we intend to do with rewriting.

To do so, we need to put forward another perspective of structured cospans. Namely, we view them as objects with arrows of their own. These arrows are commuting diagrams

$$\begin{tikzcd} La \arrow[r] \arrow[d] & x \arrow[d] & La' \arrow[l] \arrow[d] \\ Lb \arrow[r] & y & Lb' \arrow[l] \end{tikzcd} \tag{1}$$

Structured cospans and their arrows form a category $\mathsf{StrCsp}_L$ that is the subject of our first result.

**Theorem** (2.4)**.** $\mathsf{StrCsp}_L$ *is a topos.*

Because of this theorem, we can introduce rewriting systems into structured cospans. As mentioned above, the rewriting system starts with grammars, though we consider a particular type which we call a *structured cospan grammar*. This is a grammar $(\mathsf{StrCsp}_L, P)$ where $P$ consists of rules of the form

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

As before, we associate to $(\mathsf{StrCsp}_L, P)$ the new grammar $(\mathsf{StrCsp}_L, P')$ where $P'$ is the set of rules derived from the rules in $P$ using the double pushout approach. In fact, this association is functorial. We then associate, again functorially, a double category to $(\mathsf{StrCsp}_L, P')$ with objects from $\mathsf{A}$, vertical arrows are spans in $\mathsf{A}$ with invertible legs, horizontal arrows are structured cospans in $\mathsf{StrCsp}_L$, and whose squares are generated by the rules in $P'$. The composite functor Lang is called the language functor. As we see below, the language functor is used to encode a rewriting relation on $\mathsf{X}$ inside the double category $\mathrm{Lang}(\mathsf{StrCsp}_L, P)$.

Using a double cateogry allows us to combine into a single structure the connectability (horizontal composition) and rewritability (vertical composition) of structured cospans. The fact that this actually is a double category [9, Lem. 4.2] ensures the compatibility of connecting and rewriting via the interchange law. This compatability grants us the ability to decompose a system into subsystems, rewrite those, then connect the results.

At this point, we begin to connect the rewriting relation on a grammar $(\mathsf{X}, P)$ with the language of a certain structured cospan grammer $(\mathsf{StrCsp}_L, \hat{P})$.

One result Gadducci and Heckle rely on goes back to the expressiveness of certain grammars given by Ehrig, et. al. when they introdruced graph rewriting. That is, a set of rules $\{\ell_j \leftarrowtail k_j \rightarrowtail r_j\}$ has the same rewrite relation as the set of rules $\{\ell_j \leftarrowtail k'_j \rightarrowtail r_j\}$ where $k'_j$ is the discrete graph underlying $k_j$ [11, Prop. 3.3].

Just as this fact was a keystone in Gadducci and Heckle's work, we prove a modified version of it. To understand the following statement, we use the notation $P_\flat$ to refer to the set of rules obtained from $P = \{\ell_j \leftarrowtail k_j \rightarrowtail r_j\}$ by restricting the the span lets along the counit $LRk_j \to k_j$ of the comonad $LR$.

**Theorem** (3.1)**.** *Fix a geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let $(\mathsf{X}, P)$ be a grammar such that for every $\mathsf{X}$-object $x$ in the apex of a production of $P$, the Heyting algebra* $\mathrm{Sub}(x)$ *is well-founded. The rewriting relation for a grammar $(\mathsf{X}, P)$ is equal to rewriting relation for the grammar $(\mathsf{X}, P_\flat)$.*

It follows from this theorem that, instead of working with the rewriting relation for $(\mathsf{X}, P)$, we can instead study the rewriting relation $(\mathsf{X}, P_\flat)$ because it is the same.

Now, to decompose the systems in $\mathsf{X}$ and the rules in $P$, we associate to $(\mathsf{X}, P)$ the structured cospan grammar $(\mathsf{StrCsp}_L, \hat{P})$ where $\hat{P}$ contains

$$\begin{array}{ccccc} L0 & \longrightarrow & \ell & \longleftarrow & LRk \\ \uparrow & & \uparrow & & \uparrow \\ L0 & \longrightarrow & LRk & \longleftarrow & LRk \\ \downarrow & & \downarrow & & \downarrow \\ L0 & \longrightarrow & r & \longleftarrow & LRk \end{array}$$

for every rule $LRk \to \ell \times r$ of $P_\flat$.

This structured cospan grammar effectively turns all of the systems $x$ in $\mathsf{X}$ into structured cospans $L0 \to x \leftarrow Lo$ without inputs or outputs and turns the rules from $P$ into what we can think of as generaters $\hat{P}$ for a double category. The main result of the paper is that the process complete encodes the rewriting relation for $(\mathsf{X}, P)$ inside of a vertical hom-set of a double category generated by structured cospans.

**Theorem** (3.4)**.** *Fix a geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let $(\mathsf{X}, P)$ be a grammar such that for every $\mathsf{X}$-object $x$ in the apex of a production of $P$, the Heyting algebra* $\mathrm{Sub}(x)$ *is well-founded. Given $g, h \in \mathsf{X}$, then $g \rightsquigarrow^* h$ in the rewriting relation for a grammar $(\mathsf{X}, P)$ if and only if there is a square*

$$\begin{array}{ccccc} LR0 & \to & g & \leftarrow & LR0 \\ \uparrow & & \uparrow & & \uparrow \\ LR0 & \to & d & \leftarrow & LR0 \\ \downarrow & & \downarrow & & \downarrow \\ LR0 & \to & h & \leftarrow & LR0 \end{array}$$

*in the double category* $\mathrm{Lang}(\mathsf{StrCsp}_L, \hat{P})$*.*

The hypothesis of this theorem are more mild than they seem. Asking for a monic counit is means that the comonad for this adjunction is restricting a system, that is object of $\mathsf{X}$, to the subobject consisting of all the system components that can serve as an interface. Asking that the subobject algebra be well-founded is not too restrictive given that systems of interest are typically finite anyway.

Because this theorem completely characterizes the rewriting relation as squares framed by 0 inside of a double category, given any decomposition of a system $x$ into structured cospans

$$L0 \to x_1 \leftarrow La_1 \to x_2 \leftarrow La_2 \cdots La_{n-1} \to x_n \leftarrow L0$$

we can rewrite each of these structured cospans independently using rewriting rules on these subsystems. This will give a pasting diagram of the sort

$$
\begin{tikzcd}[row sep=small]
L0 \arrow[r] & x_1 \arrow[d, tail] & La_1 \arrow[l] & & La_{n-1} \arrow[r, tail] & x_n \arrow[d, tail] & L0 \arrow[l] \\
L0 \arrow[r] \arrow[u] \arrow[d] & x_1' \arrow[d] & La_1' \arrow[l] \arrow[u] \arrow[d] & \cdots & La_{n-1} \arrow[r, tail] \arrow[u] \arrow[d] & x_n' \arrow[d] & L0 \arrow[l] \arrow[u] \arrow[d] \\
L0 \arrow[r] & x_1'' & La_1'' \arrow[l] & & La_{n-1} \arrow[r, tail] & x_n'' & L0 \arrow[l] \\
& \vdots & & & & \vdots & \\
L0 \arrow[r] & y_1 \arrow[d, tail] & La_1 \arrow[l] & & La_{n-1} \arrow[r, tail] & y_n \arrow[d, tail] & L0 \arrow[l] \\
L0 \arrow[r] \arrow[u] \arrow[d] & y_1' \arrow[d] & La_1 \arrow[l] \arrow[u] \arrow[d] & \cdots & La_{n-1} \arrow[r, tail] \arrow[u] \arrow[d] & y_n' \arrow[d] & L0 \arrow[l] \arrow[u] \arrow[d] \\
L0 \arrow[r] & y_1'' & La_1 \arrow[l] & & La_{n-1} \arrow[r, tail] & y_n'' & L0 \arrow[l]
\end{tikzcd}
$$

Then we can compose these squares in any order to get a rewriting on the original system $x$ as desired.

The structure of this paper is as follows. Section 2 defines structured cospans and looks at the perspective of these as objects and as arrows. In particular, the subsection discussing the object perspective contains our first main result, Theorem 2.4 which states that $\mathsf{StrCsp}_L$ is a topos. Section 3 consists of three parts. The first is a brief overview of double pushout rewriting as applied to topoi. The second part contains our second result, Theorm 3.1 which is that a grammar and its associated discrete grammar have the same rewrite relation. By the associated discrete grammar, we mean that for each rule in the grammar, we are restricting the apex $k$ to its maximal interface subobject $LRk$. The final part of this section contains our main result, Theorem 3.4.

The author would like to thank John Baez and Fabio Gadducci for helpful conversations.

## 2. Structured Cospans

Structured cospans were introduced by Baez and Courser [1] to provide syntax for compositional systems. Their work has two aims: maximize the generality of the structured cospan construction using double categories and also to compare structured cospans to Fong's decorated cospans [12], an alternative syntax. Because structured cospans are a syntax, we want to set up a framework that can reflect the semantics. This paper proposes such a framework, for which we use the notion of double pushout rewriting. Due to our motivation, we assume different (but not disjoint) hypothesis than Baez and Courser. The purpose of this section is to set our hypothesis and explore the nature of structured cospans in this context. Instead of embarking on a mission to fully lay out a theory of structured cospans, we restrain ourselves to just those aspects needed to introduce rewriting.

To be specific, in this section we make explicit competing perspectives. The first is looking at structured cospans as objects of a category with appropriate morphisms between them. The second takes structured cospans as morphisms between certain "interfaces". The latter perspective encode the compositional structure. We then complete this section by marrying the two perspectives using double categories.

Fix an arbitrary geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$. This is an adjunction

$$\begin{tikzcd} \mathsf{X} \arrow[r, bend right, "R"'] \arrow[r, phantom, "\perp"] & \mathsf{A} \arrow[l, bend right, "L"'] \end{tikzcd}$$

between (elementary) topoi with $L$ left exact. Because spans and cospans factor heavily into this work, we use the notation $(f, g) \colon y \to x \times z$ for a span

$$x \xleftarrow{f} y \xrightarrow{g} z$$

and $(f, g) \colon x + z \to y$ for a cospan

$$x \xrightarrow{f} y \xleftarrow{g} z.$$

Because all of the categories in this paper have products and coproducts, this notation is sensible.

### 2.1. **Structured cospans as objects.**

**Definition 2.1.** A **structured cospan** is a cospan of the form $La + Lb \to x$. When we want to emphasize $L$, we use the term $L$-structured cospans.

The motivating force behind inventing structured cospans is to describe open systems, and so we do not hesitate to draw on the the intuition of open systems to better understand structured cospans. For instance, one should view the topos $\mathsf{X}$ 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 $\mathsf{A}$ should be thought to contain possible interfaces for the closed systems. Equipping a closed system with an interface provides the system a way to interact with compatible elements of the outside world. Such a system is no longer closed, and so we call it an *open system*. The left adjoint $L$ sends these interfaces into $\mathsf{X}$ so that they might interact with the closed systems. The right adjoint $R$ can be though of as returning all possible interface elements of a closed system.

Through this perspective, a structured cospan consists of a closed system $x$ equipped with the interface described by the arrows from $La$ and $Lb$. By ignoring questions of causality, we may safely consider $La$ as the input to $x$ and $Lb$ as the output. As expected, a morphism of open system ought to respect these components.

**Definition 2.2.** A morphism from one $L$-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

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

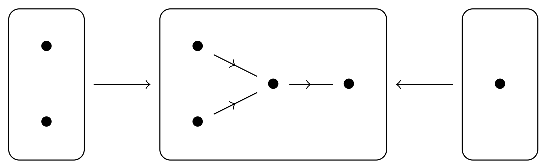
It is easy to check that $L$-structured cospans and their morphisms form a category, which we denote by $\mathsf{StrCsp}_L$.

**Example 2.3** (Open graphs)**.** Systems theory is intimately tied with graph theory. A natural example of a structured cospan is an *open graph*. While this notion is not new [10, 13], our infrastructure generalizes it.

Denote by $\mathsf{RGraph}$ the category of (directed reflexive multi-) graphs. There is an adjunction

$$\begin{tikzcd} \mathsf{RGraph} \arrow[r, bend right, "R"'] \arrow[r, phantom, "\perp"] & \mathsf{Set} \arrow[l, bend right, "L"'] \end{tikzcd}$$

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 morphisms defined as suggested by the illustration. In total, the three graphs and two graph morphisms make up a single open graph whose inputs and outputs are, respectively, the left and right-most graphs.

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 $\mathsf{StrCsp}_L$ does not encode the compositional structure, but we introduce a new category $\mathsf{Cospan}_L$ in Section 2.2 which does.

We now come to the first of our main results: that $\mathsf{StrCsp}_L$ is a topos. In terms of this paper, this result is critical because it allows for the introduction of rewriting onto structured cospans.

**Theorem 2.4.** *The category* $\mathsf{StrCsp}_L$ *is a topos.*

*Proof.* The category $\mathsf{StrCsp}_L$ constructed using the geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$ 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

$$\begin{tikzcd} w \arrow[r] \arrow[d, "f"'] & Ra \arrow[d, "Rg"'] & x \arrow[l] \arrow[d, "h"'] \\ y \arrow[r] & Rb & z \arrow[l] \end{tikzcd}$$

This, in turn, is equivalent to the comma category $(\mathsf{A} \times \mathsf{A} \downarrow \Delta R)$, where $\Delta \colon \mathsf{A} \to \mathsf{A} \times \mathsf{A}$ is the diagonal functor. But the diagonal functor is right adjoint to taking binary coproducts. That means $\Delta R$ is also a right adjoint and, furthermore, that $(\mathsf{A} \times \mathsf{A} \downarrow \Delta R)$ is an instance of Artin gluing [18], hence a topos. $\square$

We now show that the construction of $\mathsf{StrCsp}_L$ is actually functorial.

**Theorem 2.5.** *There is a functor*

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

*defined by*

$$
\begin{tikzcd}
\mathsf{X} \arrow[rr, shift left, leftarrow, "L"] \arrow[rr, shift right, "R"'] \arrow[rr, phantom, "\perp"] \arrow[dd, shift left, "G"] \arrow[dd, phantom, "\dashv"] & & \mathsf{A} \arrow[dd, shift right, "G'"'] \arrow[dd, phantom, "\vdash"] \\
& & \\
\mathsf{X}' \arrow[uu, shift left, "F"] \arrow[rr, shift left, "R'"] \arrow[rr, shift right, leftarrow, "L'"'] \arrow[rr, phantom, "\top"] & & \mathsf{A}' \arrow[uu, shift right, "F'"']
\end{tikzcd}
\quad \overset{\mathsf{StrCsp}_{(-)}}{\longmapsto} \quad
\begin{tikzcd}
\mathsf{StrCsp}_L \arrow[rr, shift left, "\Theta"] \arrow[rr, phantom, "\perp"] & & \mathsf{StrCsp}_{L'} \arrow[ll, shift left, "\Theta'"]
\end{tikzcd}
$$

*which is in turn given by*

$$
\begin{tikzcd}
La \arrow[r, "m"] \arrow[d, "Lf"'] & x \arrow[d, "g"'] & Lb \arrow[l, "n"'] \arrow[d, "Lh"'] \\
Lc \arrow[r, "o"] & y & Ld \arrow[l, "p"']
\end{tikzcd}
\quad \overset{\Theta}{\longmapsto} \quad
\begin{tikzcd}
L'G'a \arrow[r, "Gm"] \arrow[d, "L'G'f"'] & Gx \arrow[d, "Gg"'] & L'G'b \arrow[l, "Gn"'] \arrow[d, "L'G'h"'] \\
L'G'c \arrow[r, "Go"] & Gy & L'G'd \arrow[l, "Gp"']
\end{tikzcd}
$$

*and*

$$
\begin{tikzcd}
L'a' \arrow[r, "m'"] \arrow[d, "L'f'"'] & x' \arrow[d, "g'"'] & L'b' \arrow[l, "n'"'] \arrow[d, "L'h'"'] \\
L'c' \arrow[r, "o'"] & y' & L'd' \arrow[l, "p'"']
\end{tikzcd}
\quad \overset{\Theta'}{\longmapsto} \quad
\begin{tikzcd}
LF'a' \arrow[r, "Fm'"] \arrow[d, "LF'f'"'] & Fx' \arrow[d, "Fg'"'] & LF'b' \arrow[l, "Fn'"'] \arrow[d, "LF'h'"'] \\
LF'c' \arrow[r, "Fo'"] & Fy' & LF'd' \arrow[l, "Fp'"']
\end{tikzcd}
$$

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

Denote the structured cospans

$$(m, n) \colon La + Lb \to x$$

in $\mathsf{StrCsp}_L$ by $\ell$ and

$$(m', n') \colon L'a' + L'b' \to x'$$

in $\mathsf{StrCsp}_{L'}$ by $\ell'$. Also, denote the unit and counit for $F \dashv G$ by $\eta$, $\varepsilon$ and for $F' \dashv G'$ by $\eta'$, $\varepsilon'$. The assignments

$$((f, g, h) \colon \ell \to \Theta'\ell') \mapsto ((\varepsilon' \circ F'f, \varepsilon \circ Fg, \varepsilon' \circ F'h) \colon \Theta\ell \to \ell') \tag{2}$$

$$((f', g', h') \colon \Theta\ell \to \ell') \mapsto ((G'f' \circ \eta', Gg' \circ \eta, G'h' \circ \eta') \colon \ell \to \Theta'\ell') \tag{3}$$

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

Even though $\mathsf{StrCsp}_L$ is a topos, and we are heavily dependent on the topos theory, we are not currently interested in developing the theory of stuctured cospans interal to $\mathsf{Topos}$. The primary reason is that the sort of morphisms $\mathsf{StrCsp}_L \to \mathsf{StrCsp}_{L'}$ we are interested in are not geometric morphisms, but instead are the following.

**Definition 2.6.** A **structured cospan functor** is a pair of finitely continuous and cocontinuous functors $F \colon \mathsf{X} \to \mathsf{X}'$ and $G \colon \mathsf{A} \to \mathsf{A}'$ such that $FL = L'F$ and $GR = R'F$.

Structured cospan categories and their morphisms do form a category, but we leave it unnamed.

2.2. **Structured cospans as arrows.** We now turn to capturing the compositional structure that truly motivates the invention of structured cospans. To do this, we shift perspectives from structured cospans as objects in $\mathsf{StrCsp}_L$ to structured cospans as morphisms.

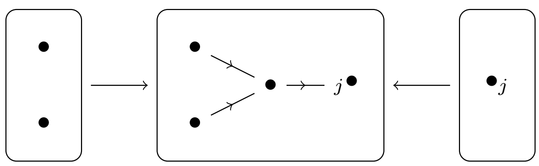
**Definition 2.7.** Denote by $\mathsf{Cospan}_L$ the category that has the same objects as $\mathsf{A}$ and structured cospans $La + Lb \to x$ as arrows of type $a \to b$.

Note that the composition of $La+Lb \to x$ with $Lb+Lc \to y$ is given by pushout:

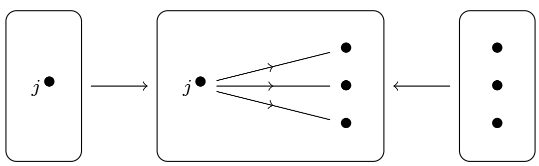
$$\begin{tikzcd} & x +_{Lb} y & \\ La \arrow[ur] & & Lc \arrow[ul] \end{tikzcd}$$

Pushouts, in a sense, are a way of gluing things together. Hence using pushouts as composition is a sensible way to model system connection. The composition above is like connecting along $Lb$. To illustrate this we return to the open graphs example.

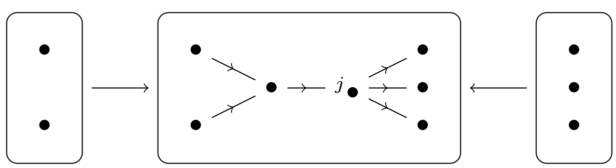
**Example 2.8.** The open graph




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. For a precise definition of a symmetric monoidal double category, we point to Shulman [17], though for the sake of completeness, we list the key pieces. A (psuedo) double category $\mathbb{C}$ is a category weakly internal to $\mathsf{Cat}$. Roughly, this is a pair of categories $(\mathsf{C}_0, \mathsf{C}_1)$ assembled together as follows.

- The $\mathbb{C}$-objects are exactly the $\mathsf{C}_0$-objects.
- The vertical arrows $c \to d$ in $\mathbb{C}$ between $\mathsf{C}$-objects are exactly the $\mathsf{C}_0$-arrows.
- The horizontal arrows $c \nrightarrow d$ in $\mathbb{C}$ between $\mathbb{C}$-objects are $\mathsf{C}_1$-objects.

- The squares of $\mathbb{C}$ are

$$\begin{tikzcd} c \arrow[r, "m", "|" marking] \arrow[d, "f"'] \arrow[dr, phantom, "\Downarrow\theta"] & d \arrow[d, "g"] \\ c' \arrow[r, "n"', "|" marking] & d' \end{tikzcd} \qquad \begin{gathered} c, c', d, d' \in \mathrm{ob}(\mathsf{C}_0) \\ f, g \in \mathrm{arr}(\mathsf{C}_0) \\ m, n \in \mathrm{ob}(\mathsf{C}_1) \\ \theta \in \mathrm{arr}(\mathsf{C}_1) \end{gathered}$$

are the arrows of $\mathsf{C}_1$.

In addition, there are structure maps ensuring the correct interplay between the elements of this data. The vertical arrows compose as they do in $\mathsf{C}_0$ and there is a structure map for composing horizontal arrows. The squares can compose both horizontally and vertically.

Observe that the horizontal arrows play two roles: as objects in their origin category and arrows in the double category. This reflects the content of the categories $\mathsf{StrCsp}_L$ and $\mathsf{Cospan}_L$. Here is a first example of a double category.

**Definition 2.9.** There is a double category $\mathbb{S}\mathsf{trCsp}_L \coloneqq (\mathsf{A}, \mathsf{StrCsp}_L)$ :

- the objects are the $\mathsf{A}$-objects
- the vertical arrows $a \to b$ the $\mathsf{A}$-arrows,
- the horizontal arrows $a \nrightarrow b$ are the cospans $La + Lb \to x$, and
- the squares are the commuting diagrams

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

Baez and Courser proved that this actually is a double category [1, Cor. 3.9]. Moreover, when $\mathsf{A}$ and $\mathsf{X}$ are cocartesian, their coproducts can be used to define a tensor product on $\mathbb{S}\mathsf{trCsp}_L$. This tensor encodes the idea that the disjoint union of considering the disjoint union of two systems as a single system. Because we have no need for this structure in this paper, we say no more about it.

*Remark* 2.10. Double categories are a nice way of capturing both the object-ness and arrow-ness of structured cospans. An alternative would be to use bicategories, but this doesn't reflect the nature of structured cospans as faithfully as does double categories.

## 3. Rewriting

We begin this final section by recalling the basics of double pushout rewriting within the context of topoi. We also present the second of our main results: a generalization, from rewriting graphs to rewriting topoi, about the expressiveness of certain graph grammars. We then apply this rewriting theory to structured cospans. In doing so, we introduce some new categorical bookkeeping devices that shows that the rewrite relation is obtained functorially. This section contains our main result which is a generalization of work by Gadducci and Heckle [13]. However, this result is not simply a mere generalization but justifies the study of systems using structured cospans. We end the section by exploring this justification.

Double pushout rewriting was introduced for graphs by Ehrig, et. al. [11]. It has since undergone extensive study and generalization. Currently, the most general setting to contain a rich theory of rewriting is adhesive categories, introduced by Lack and Sobociński [15]. Topoi, such as structured cospan categories, are examples of adhesive categories [16] so the theory we are developing in this paper admits rewriting.

Because topoi are the greatest level of generality we need, we only recall rewriting at this level. Of course, these concepts hold for adhesive categories in general, but restricting to topoi allows us to avoid an unessecary digression

3.1. **Rewriting topoi.** Fix a topos $\mathsf{C}$. Since rewriting in topoi is abstracted from graph rewriting, the archetypal topos for us is $\mathsf{RGraph}$.

Rewriting starts with the notion of a **rewrite rule**, or simply **rule**. This is a span

$$\ell \leftarrowtail k \rightarrowtail r$$

in $\mathsf{C}$ with monic legs. We continue to denote spans by $k \to \ell \times r$ and specifying it is a rule indicates that the legs are monic. The conciept of a rule is that $\ell$ is replaced by $r$ with $k$ a fixed subobject common to both. We can then apply this rule to suitable objects having $\ell$ as a subobject. Suitability for $m \colon \ell \rightarrowtail g$ means a **pushout complement** exists, that is an object $d$ fitting into a pushout diagram

$$\begin{tikzcd} \ell \arrow[d, "m"'] \arrow[dr, phantom, "\text{p.o.}"] & k \arrow[l] \arrow[d] \\ g & d \arrow[l] \end{tikzcd}$$

A pushout complement need not exist, but if it does and the map $k \to \ell$ is monic, then it is unique up to isomorphism [15, Lem. 15]. Given a rule together with a suitable $g$, we obtain a **derived rule** on the bottom row of the *double pushout diagram*

$$\begin{tikzcd} \ell \arrow[d, tail] \arrow[dr, phantom, "\text{p.o.}"] & k \arrow[l, tail] \arrow[r, tail] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r \arrow[d] \\ g & d \arrow[l] \arrow[r] & h \end{tikzcd}$$

Indeed, the span $d \to g \times h$ is a rule because pushouts preserve monics in topoi [15, Lem. 12]. The intuition of this is that we are identifying an instance of $\ell$ in $g$ and replacing it with $r$ in a cohesive manner, thus resulting in a new object $h$.

A topos $\mathsf{C}$ together with a finite set $P$ of rules $\{k_j \to \ell_j \times r_j\}$ in $\mathsf{C}$ is called a **grammar**. An arrow of grammars $(\mathsf{C}, P) \to (\mathsf{D}, Q)$ is a generic functor $F \colon \mathsf{C} \to \mathsf{D}$ such that $FP \subseteq Q$. Together these form a category $\mathsf{Gram}$.

Every grammar $(\mathsf{C}, P)$ gives rise to a relation on the objects of $\mathsf{C}$ defined by $g \rightsquigarrow h$ whenever there exists a rule $d \to g \times h$ derived from a production in $P$. However, this relation is not sufficient. For one, it is not true in general that $x \rightsquigarrow x$ holds. Also, it doesn't capture multistep rewrites. That is, perhaps there are derived rules witnessing $g \rightsquigarrow g'$ and $g' \rightsquigarrow g''$ but not a derived rule witnessing $g \rightsquigarrow g''$. However, we want to be able to relate a pair of objects if one can be rewritten into another after a finite sequence of derived rules. Therefore, the relation we actually want is the reflexive and transitive closure of $\rightsquigarrow$, which we denote by $\rightsquigarrow^*$. This is called the **rewrite relation**. Every grammar gives rise to a unique rewrite relation. Moreover, this can be done functorially, though we content ourselves to restrict our attention to showing this in the context of structured cospan categories.

3.2. **Generalizing a result from graph rewriting.** In this section, we lift a well-known result [11, Prop. 3.3] from the theory of rewriting graphs into the theory of rewriting topoi.

The original result is as follows. Let $\flat\colon \mathsf{RGraph} \to \mathsf{RGraph}$ denote the underlying discrete graph comonad. Given a grammar $(\mathsf{RGraph}, P)$, define a new grammar $(\mathsf{RGraph}, P_\flat)$ where $P_\flat$ consists of rules $k_\flat \hookrightarrow k \to \ell \times r$ for each rule $k \to \ell \times r$ in $P$. Then a graph $g$ is related to a graph $h$ with respect to the rewrite relation induced by $(\mathsf{RGraph}, P)$ if and only if $g$ is related to $h$ with respect to the rewriting relation induced by $(\mathsf{RGraph}, P_\flat)$.

To generalize this result, we first need a few notions. Fix a geometric morphism $L \dashv R\colon \mathsf{X} \to \mathsf{A}$. Denote by $(\mathsf{X}, P_\flat)$ the **discrete grammar** underlying $(\mathsf{X}, P)$. This consists of all rules obtained by pulling back $k \to \ell \times r$ by the counit $LRk \to k$ for each rule in $P$.

Recall that a poset is **well-founded** if every non-empty subset has a minimal element. Whenever the axiom of choice is present, well-foundedness is equivalent to the lack of infinite descending chains. For a relevant example, as the axiom of choice holds in any presheaf category, the Heyting algebra $\mathrm{Sub}(x)$ for any finite-set valued presheaf $x$ is well-founded.

**Theorem 3.1.** *Fix a geometric morphism $L \dashv R\colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let $(\mathsf{X}, P)$ be a grammar such that for every $\mathsf{X}$-object $x$ in the apex of a production of $P$, the Heyting algebra $\mathrm{Sub}(x)$ is well-founded. The rewriting relation for a grammar $(\mathsf{X}, P)$ is equal to rewriting relation for the grammar $(\mathsf{X}, P_\flat)$*

*Proof.* For any derivation

$$\begin{tikzcd} \ell \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & k \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r \arrow[d] \\ g & d \arrow[l] \arrow[r] & h \end{tikzcd}$$

arising from $P$, there is a derivation

$$\begin{tikzcd} \ell \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & k \arrow[l] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & LRk \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & k \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r \arrow[d] \\ g & d \arrow[l] & w \arrow[l] \arrow[r] & d \arrow[r] & h \end{tikzcd}$$

where

$$w \coloneqq \bigwedge \{z \colon z \wedge k = x\} \vee LRk.$$

Note that $w \vee k = x$ and $w \wedge k = LRy$ which gives that the two inner squares of the lower diagram are pushouts. $\square$

3.3. **Rewriting structured cospans.** We now turn to rewriting structured cospans. The ability of structured cospans to give a nice theory of rewriting lies in the fact that they form a topos (Theorem 2.4). The first thing we do is appropriately restrict $\mathsf{Gram}$ to a subcategory $\mathsf{StrCspGram}$. The objects are $(\mathsf{StrCsp}_L, P)$ where $P$

consists of rules of the form

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

and the morphisms are the structured cospan functors (Definition 2.6) that are stable under the grammars.

Recall that to each grammar is associated a relation $\rightsquigarrow$ and its reflexive transitive closure, the rewrite relation$\rightsquigarrow^*$. We now show that this can be done functorially via a composite of two functors, $D \colon \mathsf{StrCspGram} \to \mathsf{StrCspGram}$ and $S \colon \mathsf{StrCspGram} \to \mathsf{DblCat}$, which we now define.

is composition preserved?

**Lemma 3.2.** *There is an idempotent functor $D \colon \mathsf{StrCspGram} \to \mathsf{StrCspGram}$. It is defined on objects by setting $D(\mathsf{StrCsp}_L, P)$ to be the grammar $(\mathsf{StrCsp}_L, P')$, where $P'$ consists of all rules $h \to g \times d$ witnessing the relation $g \rightsquigarrow h$ with respect to $(\mathsf{StrCsp}_L, P)$. On arrows, $DF \colon D(\mathsf{StrCsp}_K, P) \to D(\mathsf{StrCsp}_L, Q)$ is defined exactly as $F$. Moreover, the identity on $\mathsf{StrCspGram}$ is a subfunctor of $D$.*

*Proof.* That $D(\mathsf{StrCsp}, P)$ actually gives a grammar follows from the fact that pushouts respect monics in a topos [15, Lem. 12].

That $D$ is idempotent is equivalent to saying that, for a set $P$ of rules, $g \rightsquigarrow h$ with respect to $D(\mathsf{StrCsp}_L Gram, P)$ if and only if $g \rightsquigarrow h$ with respect to $DD(\mathsf{StrCspGram}_L, P)$. This follows from the fact that the outer box of the diagram

$$\begin{tikzcd} \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & \bullet \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & \bullet \arrow[d] \\ \bullet \arrow[r] & \bullet \arrow[r] & \bullet \end{tikzcd}$$

is a pushout.

The identity is a subfunctor of $D$ because $\ell \rightsquigarrow r$ for any production $k \to \ell \times r$ in $(\mathsf{StrCsp}_L, P)$ via a triple of identity arrows. Hence the identity functor on $\mathsf{StrCsp}_L$ turns $(\mathsf{StrCsp}_L, P)$ into a subobject of $D(\mathsf{StrCsp}_L, P)$. ☐

what's the meaning of this lemma?

To define $S$, we reference the double category $\mathbb{M}\mathsf{onSpCsp}(\mathsf{C})$ for a topos $\mathsf{C}$ introduced in [9]. The objects are those in $\mathsf{C}$, the vertical arrows are spans with invertible legs in $\mathsf{C}$, the horizontal arrows are cospans in $\mathsf{C}$, and the squares are diagrams in $\mathsf{C}$ with shape

$$\begin{tikzcd} \bullet \arrow[r] & \bullet & \bullet \arrow[l] \\ \bullet \arrow[u, "\cong"] \arrow[r] \arrow[d, "\cong"'] & \bullet \arrow[u, tail] \arrow[d, tail] & \bullet \arrow[l] \arrow[u, "\cong"'] \arrow[d, "\cong"] \\ \bullet \arrow[r] & \bullet & \bullet \arrow[l] \end{tikzcd}$$

Given a structured cospan grammar $(\mathsf{StrCsp}_L, P)$, observe that the productions in $P$ are admissible as squares in $\mathbb{M}\mathsf{onSpCsp}(\mathsf{X})$. Denote by $S(\mathsf{StrCsp}_L, P)$ the sub-double category of $\mathbb{M}\mathsf{onSpCsp}(\mathsf{X})$ that is full on objects, vertical and horizontal arrows, and generated by the productions in $P$. This assignment is functorial

double check this

because
$$(F,G)\colon (\mathsf{StrCsp}_L, P) \to (\mathsf{StrCsp}_{L'}, P')$$
gives a mapping between the generators of $S(\mathsf{StrCsp}_L, P)$ and $S(\mathsf{StrCsp}_{L'}, P')$. Composition holds because $F$ and $G$ both preserve pullbacks and pushouts. This allows us to define the language functor $\mathrm{Lang} := SD$.

rotate to main result

Here is a quick lemma that we use in the next theorm.

**Lemma 3.3.** *If $x \rightsquigarrow^* y$ and $x' \rightsquigarrow^* y'$, then $x + x' \rightsquigarrow^* y + y'$*

*Proof.* It the derivation $x \rightsquigarrow^* y$ comes from a string of double pushout diagrams

$$\begin{tikzcd}
\ell_1 \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & k_1 \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r_1 \arrow[dr] & & \ell_2 \arrow[dl] \arrow[dr, phantom, "\text{p.o.}"] & k_2 \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r_2 \arrow[d] & \cdots & \ell_n \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & k_n \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r_n \arrow[d] \\
x & d_1 \arrow[l] \arrow[rr] & & w_1 & & d_2 \arrow[ll] \arrow[r] & w_2 & & w_{n-1} & d_n \arrow[l] \arrow[r] & y
\end{tikzcd}$$

and the derivation $x \rightsquigarrow^* y$ comes from a string of double pushout diagrams

$$\begin{tikzcd}
\ell'_1 \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & k'_1 \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r'_1 \arrow[dr] & & \ell'_2 \arrow[dl] \arrow[dr, phantom, "\text{p.o.}"] & k'_2 \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r'_2 \arrow[d] & \cdots & \ell'_n \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & k'_m \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r'_m \arrow[d] \\
x' & d'_1 \arrow[l] \arrow[rr] & & w'_1 & & d'_2 \arrow[ll] \arrow[r] & w'_2 & & w'_{m-1} & d'_m \arrow[l] \arrow[r] & y'
\end{tikzcd}$$

then $x + x' \rightsquigarrow^* y + y'$ is realized by concatenating to the end of first string with $x'$ summed with the bottom row the second string with $y$ summed on the bottom row. □

The desire behind the main result is the ability to study systems, as represented by objects in a topos $\mathsf{X}$, locally. The mechanism (that is, structured cospans) by which we do this is to equip systems with interfaces that allow us to connect sub-systems together. Another way to view this is that given a system can be decomposed into sub-systems. These can be studied individually then reconnected along the interfaces this mechanism provides. The manner in which the main result can accomplish this is discussed below the theorem

We need the following definition. Associate to a grammar $(\mathsf{X}, P)$ the structured cospan grammar $(\mathsf{StrCsp}_L, P')$ where $P'$ contains

$$\begin{tikzcd}
L0 \arrow[r] & \ell & LRk \arrow[l] \\
L0 \arrow[r] \arrow[u] \arrow[d] & LRk \arrow[u] \arrow[d] & LRk \arrow[l] \arrow[u] \arrow[d] \\
L0 \arrow[r] & r & LRk \arrow[l]
\end{tikzcd}$$

for every rule $LRk \to \ell \times r$ of $P_\flat$.

**Theorem 3.4.** *Fix a geometric morphism $L \dashv R\colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let $(\mathsf{X}, P)$ be a grammar such that for every $\mathsf{X}$-object $x$ in the apex of a production of $P$, the Heyting algebra $\mathrm{Sub}(x)$ is well-founded. Given $g$, $h \in \mathsf{X}$, then $g \rightsquigarrow^* h$ in the* *rewriting relation for a grammar* $(\mathsf{X}, P)$ *if and only if there is a square*

$$\begin{tikzcd} LR0 \arrow[r] & g & LR0 \arrow[l] \\ LR0 \arrow[r] \arrow[u] \arrow[d] & d \arrow[u] \arrow[d] & LR0 \arrow[l] \arrow[u] \arrow[d] \\ LR0 \arrow[r] & h & LR0 \arrow[l] \end{tikzcd}$$

*in the double category* $\mathrm{Lang}(\mathsf{StrCsp}_L, P')$.

*Proof.* We show sufficiency by induction on the length of the derivation. If $g \rightsquigarrow h$

$$\begin{tikzcd} \ell \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & LRk \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\text{p.o.}"] & r \arrow[d] \\ g & d \arrow[l] \arrow[r] & h \end{tikzcd}$$

the desired square is the horizontal composition of

$$\begin{tikzcd} L0 \arrow[r] & \ell & LRk \arrow[l] \arrow[r] & d & L0 \arrow[l] \\ L0 \arrow[r] \arrow[u] \arrow[d] & LRk \arrow[u] \arrow[d] & LRk \arrow[l] \arrow[r] \arrow[u] \arrow[d] & d \arrow[u] \arrow[d] & L0 \arrow[l] \arrow[u] \arrow[d] \\ L0 \arrow[r] & r & LRk \arrow[l] \arrow[r] & d & L0 \arrow[l] \end{tikzcd}$$

The left square is a generator and the right square is the identity on the horizontal arrow $LRk + L \to d$. The square for a derivation $g \rightsquigarrow^* h \rightsquigarrow j$ is the vertical composition of

$$\begin{tikzcd} L0 \arrow[r] & g & L0 \arrow[l] \\ L0 \arrow[r] \arrow[u] \arrow[d] & d \arrow[u] \arrow[d] & L0 \arrow[l] \arrow[u] \arrow[d] \\ L0 \arrow[r] & h & L0 \arrow[l] \\ L0 \arrow[r] \arrow[u] \arrow[d] & e \arrow[u] \arrow[d] & L0 \arrow[l] \arrow[u] \arrow[d] \\ L0 \arrow[r] & j & L0 \arrow[l] \end{tikzcd}$$

The top square is from $g \rightsquigarrow^* h$ and the second from $h \rightsquigarrow j$.

Conversely, proceed by structural induction on the generating squares of $\mathrm{Lang}(\mathsf{StrCsp}_L, P')$. It suffices to show that the rewrite relation is preserved by vertical and composition by a generating square. Suppose we have a square

$$\begin{tikzcd} L0 & w \arrow[l] \arrow[r] & L0 \\ L0 \arrow[u] \arrow[d] & x \arrow[l] \arrow[r] \arrow[u] \arrow[d] & L0 \arrow[u] \arrow[d] \\ L0 & y \arrow[l] \arrow[r] & L0 \end{tikzcd}$$

corresponding to a derivation $w \rightsquigarrow^* y$. Composing this vertically with a generating square, which must have form

$$\begin{tikzcd} L0 & y \arrow[l] \arrow[r] & L0 \\ L0 \arrow[u] \arrow[d] & L0 \arrow[l] \arrow[r] \arrow[u] \arrow[d] & L0 \arrow[u] \arrow[d] \\ L0 & z \arrow[l] \arrow[r] & L0 \end{tikzcd}$$

corresponding to a production $0 \to y + z$ gives

$$\begin{tikzcd} L0 & w \arrow[l] \arrow[r] & L0 \\ L0 \arrow[u] \arrow[d] & L0 \arrow[l] \arrow[r] \arrow[u] \arrow[d] & L0 \arrow[u] \arrow[d] \\ L0 & z \arrow[l] \arrow[r] & L0 \end{tikzcd}$$

which corresponds to a derivation $w \rightsquigarrow^* y \rightsquigarrow z$. Composing horizontally with a generating square

$$\begin{tikzcd} L0 & \ell \arrow[l] \arrow[r] & L0 \\ L0 \arrow[u] \arrow[d] & LRk \arrow[l] \arrow[r] \arrow[u] \arrow[d] & L0 \arrow[u] \arrow[d] \\ L0 & r \arrow[l] \arrow[r] & L0 \end{tikzcd}$$

corresponding with a production $LRk \to \ell + r$ results in the square

$$\begin{tikzcd} L0 & w + \ell \arrow[l] \arrow[r] & L0 \\ L0 \arrow[u] \arrow[d] \arrow[r] & x + LRk \arrow[u] \arrow[d] & L0 \arrow[u] \arrow[d] \arrow[l] \\ L0 & y + r \arrow[l] \arrow[r] & L0 \end{tikzcd}$$

But $w + \ell \rightsquigarrow^* y + r$ as seen in Lemma 3.3.

$\square$

With this result, we have completely described the rewrite relation for a grammar $(\mathsf{X}, P)$ with those squares in $\mathrm{Lang}(\mathsf{StrCsp}_L, P')$ framed by the initial object of $\mathsf{X}$. These squares are rewrites of a closed system, which may be difficult to understand. We can instead begin with a closed system as represented by a horizontal arrow in $\mathrm{Lang}(\mathsf{StrCsp}_L, P')$ and decompose it into a composite of easier to understand sub-systems, rather a sequence of composable horizontal arrows. Rewriting can be performed on each of these sub-systems which, of course, is represented by squares. The composite of these squares gives are rewriting of the original system.

## References

[1] J. Baez, K. Courser. Structured cospans. *In preparation*.
[2] J. Baez, J. Foley, J. Moeller, B. Pollard. Network Models. *arXiv preprint* arXiv:1711.00037. 2017.

[3] J. Baez, B. Fong. A compositional framework for passive linear networks. *arXiv preprint* arXiv:1504.05625. 2015.

[4] J. Baez, B. Fong, B. Pollard. A compositional framework for Markov processes. *J. Math. Phys.* 57, No. 3: 033301. 2016.

[5] J. Baez, B. Pollard. A compositional framework for reaction networks. *Rev. Math. Phys.* 29, No. 9, 1750028. 2017.

[6] J. Baez, J. Master. Open Petri Nets. *arXiv preprint* arXiv:1808.05415. 2018.

[7] N. Chomsky. On Certain Formal Properties of Grammars. *Inf. Control.* No. 2. 137-167. 1959.

[8] D. Cicala. Spans of cospans. *Theory Appl. Categ.* 33, No. 6, 131–147. 2018.

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

[10] L. Dixon, and A. Kissinger. Open-graphs and monoidal theories. *Math. Structures Comput. Sci.*, **23**, No. 2, 308–359. 2013.

[11] H. Ehrig, M. Pfender, and H.J. Schneider. Graph-grammars: An algebraic approach. In *Switching and Automata Theory, 1973. SWAT'08. IEEE Conference Record of 14th Annual Symposium on*, 167–180. IEEE. 1973.

[12] B. Fong. Decorated cospans. *Theory Appl. Categ.* 30, Paper No. 33, 1096–1120. 2015.

[13] F. Gadducci, R. Heckel. An inductive view of graph transformation.*International Workshop on Algebraic Development Techniques*. 223–237. Springer, Berlin. 1998.

[14] A. Habel, J. M uller, D. Plump. Double-pushout graph transformation revisited. *Math. Structures Comput. Sci.* 11. No. 5. 637–688. 2001).

[15] S. Lack, and P. Sobociński. Adhesive categories. In *International Conference on Foundations of Software Science and Computation Structures*, 273–288. Springer, Berlin, Heidelberg. 2004.

[16] S. Lack, P. Sobociński. Toposes are adhesive. *International Conference on Graph Transformation.* Springer, Berlin, Heidelberg. 2006.

[17] M. Shulman. Constructing symmetric monoidal bicategories. *arXiv preprint* arXiv:1004.0993. 2010.

[18] G. Wraith. Artin gluing. *J. Pure Appl. Algebra* **4**, 345–348. 1974.