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 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. [11] 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 X paired with a set of rules P. Each rule in P is a span $\ell \leftarrowtail k \rightarrowtail r$ in X such that each leg is monic. The conceipt is that the object ℓ can be replaced by r while the common subobject k is fixed. The grammar (X, P) induces a *derived grammar* (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 (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 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 RGraph of reflexive graphs. Given a rewriting system (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 (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 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, X still is our topos of systems. The new data consists of A which is a topos comprised of "interface types". These will serve as the boundaries along which we decompose the systems in X. The left adjoint L serves as the channel through which we port the interface types into 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 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 [1], 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

$$\begin{tikzcd}[column sep=small] & x & & & y & & & & x +_{Lb} y & \\ La \arrow[ur] & & Lb \arrow[ul] & Lb \arrow[ur] & & Lc \arrow[ul] \arrow[rr, "\text{connection}"] & & La \arrow[ur] & & Lc \arrow[ul] \end{tikzcd}$$

open sub-systems composite system

This gives us a category Cospan_L whose objects are from 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 StrCsp_L that is the subject of our first result.

Theorem (2.6)**.** *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 (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 (StrCsp_L, P) the new grammar (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 (StrCsp_L, P') with objects from A, vertical arrows are spans in A with invertible legs, horizontal arrows are structured cospans in 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 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 (X, P) with the language of a certain structured cospan grammer $(\mathsf{StrCsp}_L, \widehat{P})$.

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 (3.2)**.** *Fix a geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let (X, P) be a grammar such that for every 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 (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 (X, P), we can instead study the rewriting relation $(\mathsf{X}, P_\flat)$ because it is the same.

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

This structured cospan grammar effectively turns all of the systems x in 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 $\widehat{P}$ for a double category. The main result of the paper is that the process complete encodes the rewriting relation for (X, P) inside of a vertical hom-set of a double category generated by structured cospans.

Theorem (3.6)**.** *Fix a geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let (X, P) be a grammar such that for every 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 (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 $\mathrm{Lang}(\mathsf{StrCsp}_L, \widehat{P})$.

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 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

$$\begin{array}{ccccccccccc}
L0 & \longrightarrow & x_1 & \leftarrow & La_1 & & La_{n-1} & \to & x_n & \longleftarrow & L0 \\
{\scriptstyle\cong}\uparrow & & \downarrow & & \uparrow{\scriptstyle\cong} & & {\scriptstyle\cong}\uparrow & & \downarrow & & \uparrow{\scriptstyle\cong} \\
L0 & \longrightarrow & x_1' & \leftarrow & La_1' & \cdots & La_{n-1} & \to & x_n' & \longleftarrow & L0 \\
{\scriptstyle\cong}\downarrow & & \downarrow & & \downarrow{\scriptstyle\cong} & & {\scriptstyle\cong}\downarrow & & \downarrow & & \downarrow{\scriptstyle\cong} \\
L0 & \longrightarrow & x_1'' & \leftarrow & La_1'' & & La_{n-1} & \to & x_n'' & \longleftarrow & L0 \\
 & & \vdots & & & & & & \vdots & & \\
L0 & \longrightarrow & y_1 & \leftarrow & La_1 & & La_{n-1} & \to & y_n & \longleftarrow & L0 \\
{\scriptstyle\cong}\uparrow & & \downarrow & & \uparrow{\scriptstyle\cong} & & {\scriptstyle\cong}\uparrow & & \downarrow & & \uparrow{\scriptstyle\cong} \\
L0 & \longrightarrow & y_1' & \leftarrow & La_1 & \cdots & La_{n-1} & \to & y_n' & \longleftarrow & L0 \\
{\scriptstyle\cong}\downarrow & & \downarrow & & \downarrow{\scriptstyle\cong} & & {\scriptstyle\cong}\downarrow & & \downarrow & & \downarrow{\scriptstyle\cong} \\
L0 & \longrightarrow & y_1'' & \leftarrow & La_1 & & La_{n-1} & \to & y_n'' & \longleftarrow & L0
\end{array}$$

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 2.6 which states that 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 3.2 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 3.6.

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

2. Structured Cospans

Baez and Courser introduced structured cospans to provide syntax for compositional systems [1]. Their aims are to maximize the generality of the structured cospan construction using double categories and to compare structured cospans to Fong's decorated cospans [12], an alternative contruction. Because structured cospans are a syntax, we want to set up a framework that can reflect the semantics. This paper proposes double pushout rewriting as this framework. In this section, we set our hypotheses and explore structured cospans in this context. In particular, we see how they encode compositional structure as morphisms in a category and also how they are objects of a topos. We then complete this section by marrying the two perspectives using double categories.

Fix an adjunction

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

between (elementary) topoi with L preserving finite limits. Readers familiar with topos theory will recognize this as a *geometric morphism*. Because spans and cospans factor heavily into this work, we use the notation $\langle f, g\rangle\colon y \to x \times z$ for a span

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

and $[f,g] \colon x + z \to y$ for a cospan

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

Using this notation avoids being problematic because every category within this paper has products and coproducts.

2.1. **Structured cospans as arrows.** We discuss the compositional structure of structured cospans. This material can be found in Baez and Courser's work [1], but we include it here for completeness.

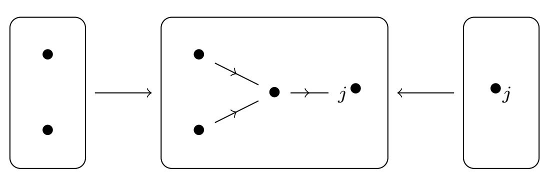
Definition 2.1. Denote by Cospan_L the category that has the same objects as A and structured cospans $La + Lb \to x$ as arrows of type $a \to b$.

Composing $La + Lb \to x$ with $Lb + Lc \to y$ uses pushout

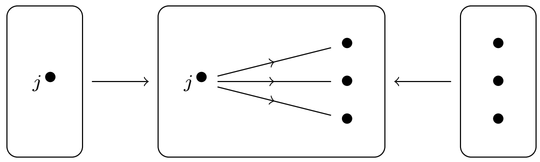
$$\begin{tikzcd} & x +_{Lb} y & \\ La \arrow[ur] & & Lc \arrow[ul] \end{tikzcd}$$

In a sense, pushouts are a way of gluing things together making it 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.

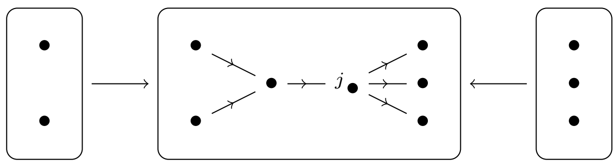
Example 2.2. The composition of



with



is the open graph



which is obtained by glueing the two open graphs together along the node j.

2.2. **Structured cospans as objects.**

Definition 2.3. 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.

Because structured cospans were invented to describe open systems, we frequently draw on this intuition. For instance, one should view X as consisting of closed systems and their morphisms. By a *closed system*, we mean a system that cannot interact with the outside world. On the other hand, A should be thought to contain interfaces types. By transporting the interface types along L, they can be combined with the systems in X. Systems equpped with an interface are called open. These can interact with compatible open systems. Finally, R returns the maximum subobject of a system that can serve as an interface.

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 consider La as the input to x and Lb as the output. It is important to note, however, that this convention is arbitrary because Cospan_L from Definition 2.1 is compact closed [9].

A morphism of open system ought to respect these components.

Definition 2.4. A morphism between L-structured cospans $La + Lb \to x$ and $Lc + Ld \to y$ is a triple of arrows (f, g, h) that fit into the commuting diagram

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

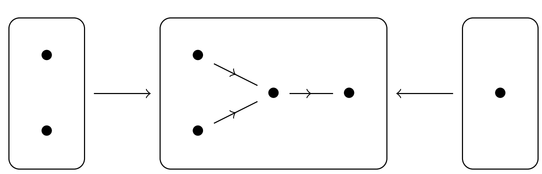
It is easy to check that L-structured cospans and their morphisms form a category, which we denote by StrCsp_L.

Example 2.5 (Open graphs)**.** Systems theory is intimately tied with graph theory. As such we take open graphs to be our archetypal example of a structured cospan. While this notion is not new [10, 13], our infrastructure generalizes it.

Denote by RGraph the category of (directed reflexive multi-) graphs. There is a geometric morphism

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

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.

We now come to the first of our main results: that StrCsp_L is a topos. This result is critical for our theory because it allows the introduction of rewriting onto structured cospans.

Theorem 2.6. *For any geometric morphism $L \dashv R \colon \mathsf{A} \to \mathsf{X}$, the category StrCsp_L is a topos.*

Proof. Note that StrCsp_L 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 the coproduct functor. Therefore, ΔR is also a right adjoint so $(\mathsf{A} \times \mathsf{A} \downarrow \Delta R)$ is an instance of Artin gluing [18], hence a topos. □

We now show that the construction of StrCsp_L is actually functorial.

Theorem 2.7. *There is a functor*

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

defined by

$$\begin{tikzcd} \mathsf{X} \arrow[r, shift left, "R"'] \arrow[d, shift left, "G"] \arrow[r, phantom, "\perp"] & \mathsf{A} \arrow[l, shift left, "L"'] \arrow[d, shift left, "G'"'] \\ \mathsf{X}' \arrow[u, shift left, "F"] \arrow[r, shift left, "R'"] \arrow[r, phantom, "\top"] & \mathsf{A}' \arrow[l, shift left, "L'"] \arrow[u, shift left, "F'"'] \end{tikzcd} \quad \xmapsto{\mathsf{StrCsp}_{(-)}} \quad \begin{tikzcd} \mathsf{StrCsp}_L \arrow[r, shift left, "\Theta"] \arrow[r, phantom, "\perp"] & \mathsf{StrCsp}_{L'} \arrow[l, shift left, "\Theta'"] \end{tikzcd}$$

which is in turn given by

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

and

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

Proof. In light of Theorem 2.6, it suffices to show that $\Theta \dashv \Theta'$ gives a geometric morphism.

Denote the structured cospans

$$[m, n]\colon La + Lb \to x$$

in StrCsp_L by ℓ and

$$[m', n']\colon L'a' + L'b' \to x'$$

in $\mathsf{StrCsp}_{L'}$ by ℓ'. Denote the unit and counit for $F \dashv G$ by η, ε and for $F' \dashv G'$ by η', ε'. The assignments

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

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

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. $\square$

Despite the fact that StrCsp_L is a topos, we are interested in the arrows $\mathsf{StrCsp}_L \to \mathsf{StrCsp}_{L'}$ acting on the systems and their interfaces.

Definition 2.8. 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 $GR = R'F$.

does this determine a geometric morphism?

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

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 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 Cat. this roughly translates to a category of objects C_0 together with a category of arrows C_1 that are assembled together as follows.

- The C_0-objects are called the objects of $\mathbb{C}$.

- The C_0-arrows are called the vertical arrows in $\mathbb{C}$.
- The C_1-objects are called the the horizontal arrows in $\mathbb{C}$.
- The C_1-arrows are called the squares of $\mathbb{C}$. are the arrows of C_1.

See Figure 1 for a depiction of this data.

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

FIGURE 1. A square in a double category

In addition, there are structure maps ensuring the correct interplay between the elements of this data. The vertical arrows compose as they do in C_0 and there is a structure map for composing horizontal arrows. The squares can compose both horizontally and vertically and there is an interchange law ensuring these compositions play well together.

Observe that the horizontal arrows play two roles: as objects in their category of origin and as arrows in the double category. This reinforces our choice to organize structured cospans in 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 A-objects
- the vertical arrows $a \to b$ the 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 truly is a double category [1, Cor. 3.9]. Moreover, when A and X are cocartesian, their coproducts can be used to define a symmetric monoidal structure on $\mathbb{S}\mathsf{trCsp}_L$. This meaning of this structure is that the disjoint union of two systems can be considered a single system. Because we have no need for this structure in this paper, we say no more about it.

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: the generalization to rewriting topoi of a result on the expressivity of certain graph grammars. We then apply this rewriting theory to structured cospans. In doing so, we introduce some new categorical bookkeeping devices that show the rewrite relation is functorial. This section also 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 provides a justification the study of systems using structured cospans. .

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 are examples of adhesive categories [16] so it follows that we can bring a theory of rewriting to structured cospans.

3.1. **Rewriting topoi.** Fix a topos C. Rewriting starts with the notion of a **rewrite rule**, or simply **rule**. This is a span

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

in C with monic legs. We continue to denote spans by $k \to \ell \times r$ but note that specifying it is a rule indicates both legs are monic. The conciept of a rule is that ℓ is replaced by r while k a fixed subobject common to both. We can apply this rule to objects $m \colon \ell \rightarrowtail g$ having ℓ as a subobject if there exists a **pushout complement**, 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 when it does and the map $k \to \ell$ is monic, then it is unique up to isomorphism [15, Lem. 15].

Given a rule $k \to l \times r$ together with a suitable $\ell \rightarrowtail g$, we obtain a **derived rule** $d \to g \times h$ on the bottom row of the *double pushout diagram*

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

The span $d \to g \times h$ is a rule because pushouts preserve monics in topoi [15, Lem. 12]. The intuition of this diagram is that $\ell \rightarrowtail g$ gives an instance of ℓ in g. This instance is replaced with r in a coherent manner, resulting in a new object h.

A topos C together with a finite set P of rules $\{k_j \to \ell_j \times r_j\}$ in C is called a **grammar**. An arrow of grammars $(\mathsf{C}, P) \to (\mathsf{D}, Q)$ is a monic preserving functor $F \colon \mathsf{C} \to \mathsf{D}$ such that for each rule $\langle f, g\rangle \colon k \to \ell \times r$ in P, the rule $\langle Ff, Fg\rangle \colon Fk \to F\ell \times Fr$ is isomorphic to a rule in Q. Together these form a category Gram.

Every grammar (C, P) gives rise to a relation $\rightsquigarrow^*$ on the objects of 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 too small to capture the full behavior induced by a grammar. 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 with 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 determines 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. **Expressivity of underlying discrete grammars.** In this section, we generalize a result [11, Prop. 3.3] from the theory of rewriting graphs into the theory of rewriting topoi.

To place this result into our current context, let $\flat \colon \mathsf{RGraph} \to \mathsf{RGraph}$ denote the underlying discrete graph comonad. The counit is an monic $\flat g \rightarrowtail g$ that includes the underlying discrete graph $\flat g$ into g. Given a grammar (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 (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 (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.

Example 3.1. 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. In the case of RGraph, the subobject algebra of any finite graph is well-founded.

Theorem 3.2. *Fix a geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let (X, P) be a grammar such that for every 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 (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

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

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 apply rewriting topoi to rewriting structured cospans which is possible because of(Theorem 2.6).

There is a subcategory $\mathsf{StrCspGram}$ of Gram whose objects are (StrCsp_L, P) where P consists of rules of the form

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

and the morphisms are the structured cospan functors (Definition 2.8) that are stable under the grammars. The objects do run through L and P.

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.

Lemma 3.3. *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* (StrCsp_L, P')*, where* P' *consists of all rules* $h \to g \times d$ *witnessing the relation* $g \rightsquigarrow h$ *with respect to* (StrCsp_L, P)*. On arrows,* $DF\colon D(\mathsf{StrCsp}_L, P) \to D(\mathsf{StrCsp}_{L'}, Q)$ *is defined exactly as* F*. Moreover, the identity on* $\mathsf{StrCspGram}$ *is a subfunctor of* D*.*

this proof may need to be fleshed out

Proof. That $D(\mathsf{StrCsp}_L, P)$ 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 $D(\mathsf{StrCsp}_L, P)$ 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 (StrCsp_L, P) via a triple of identity arrows. Hence the identity functor on StrCsp_L turns (StrCsp_L, P) into a subobject of $D(\mathsf{StrCsp}_L, P)$. $\square$

This lemma sends each grammar to a new grammar consisting of all derived rules. That D is idempotent means that a rule derived from a derived rule can be derived from the original rule. That identity is a subfunctor of D means that the derived grammar contains all of the rules of the original grammar.

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

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

Given a structured cospan grammar (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 $\mathrm{Lang} \coloneqq SD$.

We are now closing in on the main result. To prove it, we require the following lemma.

Lemma 3.4. *If $x \rightsquigarrow^* y$ and $x' \rightsquigarrow^* y'$, then $x + x' \rightsquigarrow^* y + y'$*

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

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

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

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

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. $\square$

The desire for the main result is that it opens the possibility to study systems, as represented by objects in a topos X, locally. To reiterate, using structured cospans, we equip systems with interfaces that allow us to connect them together. Another way to view this is that system decomposed into sub-systems can be studied individually then reconnected along the interfaces structured cospans provide. The manner in which the main result can accomplish this is discussed below the theorem, for which we need the following definition.

Definition 3.5. Associate to a grammar (X, P) the structured cospan grammar (StrCsp_L, P') where P' contains

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

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

Prior to stating the theorem, we note that this is a generalization of work by Gadducci and Heckle [13] and the structure of our proof is an appropriate modification of theirs.

Theorem 3.6. *Fix a geometric morphism $L \dashv R \colon \mathsf{X} \to \mathsf{A}$ with monic counit. Let (X, P) be a grammar such that for every 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 (X, P) if and only if there is a square*

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

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

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

But $w + \ell \rightsquigarrow^* y + r$ as seen in Lemma 3.4.

□

With this result, we have completely described the rewrite relation for a grammar (X, P) with squares in $\mathrm{Lang}(\mathsf{StrCsp}_L, P')$ framed by the initial object of X. These squares are rewrites of a closed system in the sense that the interface is empty. We can instead begin with a closed system x in X as represented by a horizontal arrow $L0 + L0 \to x$ in $\mathrm{Lang}(\mathsf{StrCsp}_L, P')$ and decompose it into a composite of sub-systems, that is a sequence of composable horizontal arrows

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

Rewriting can be performed on each of these sub-systems

$$\begin{array}{ccccccccccc}
L0 & \longrightarrow & x_1 & \leftarrow & La_1 & & La_{n-1} & \rightarrowtail & x_n & \longleftarrow & L0 \\
{\scriptstyle =}\uparrow & & \downarrow & & \uparrow{\scriptstyle \cong} & & {\scriptstyle =}\uparrow & & \downarrow & & \uparrow{\scriptstyle \cong} \\
L0 & \longrightarrow & x_1' & \leftarrow & La_1' & \cdots & La_{n-1} & \rightarrowtail & x_n' & \longleftarrow & L0 \\
{\scriptstyle =}\downarrow & & \downarrow & & \downarrow{\scriptstyle \cong} & & {\scriptstyle =}\downarrow & & \downarrow & & \downarrow{\scriptstyle \cong} \\
L0 & \longrightarrow & x_1'' & \leftarrow & La_1'' & & La_{n-1} & \rightarrowtail & x_n'' & \longleftarrow & L0 \\
 & & \vdots & & & & & & \vdots & & \\
L0 & \longrightarrow & y_1 & \leftarrow & La_1 & & La_{n-1} & \rightarrowtail & y_n & \longleftarrow & L0 \\
{\scriptstyle =}\uparrow & & \downarrow & & \uparrow{\scriptstyle \cong} & & {\scriptstyle =}\uparrow & & \downarrow & & \uparrow{\scriptstyle \cong} \\
L0 & \longrightarrow & y_1' & \leftarrow & La_1 & \cdots & La_{n-1} & \rightarrowtail & y_n' & \longleftarrow & L0 \\
{\scriptstyle =}\downarrow & & \downarrow & & \downarrow{\scriptstyle \cong} & & {\scriptstyle =}\downarrow & & \downarrow & & \downarrow{\scriptstyle \cong} \\
L0 & \longrightarrow & y_1'' & \leftarrow & La_1 & & La_{n-1} & \rightarrowtail & y_n'' & \longleftarrow & L0
\end{array}$$

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.