REWRITING STRUCTURED COSPANS

DANIEL CICALA

Abstract. To foster the study of networks on an abstract level, we further study the formalism of *structured cospans* introduced by Baez and Courser. 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 for 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. We generalize the result from graph rewriting stating that a graph grammar induces the same rewrite relation as its underlying graph grammar. We use this fact to prove our main result, a complete characterization of the rewriting relation for a topos X using double categories. This provides a compositional framework for rewriting systems.

1. Introduction

This paper is part of a program that aims to study open systems by treating them as morphisms in a category [2, 3, 4, 5, 7]. Baez and Courser invented structured cospans, a syntactical device to reason about open systems. In this paper, we introduce a theory of rewriting to structured cospans.

In the study of formal languages, a rewriting system often accompanies a syntax. Roughly, a rewriting system is a set of rules, each providing a way to rewrite one syntactical term into another. Each rewriting system gives rise to the *rewriting relation* that relates two terms if one can be rewritten into the other using a finite sequence of rules. This relates syntactical terms with the same meaning. An example is when one replaces two resistors wired in series with a single resistor with aggregate resistance. An appropriate rewriting system would want to relate (not equate) these.

The theory of rewriting has gone through several epochs, each more abstract from the last. Noam Chomsky introduced rewriting for formal languages [6]. Here, the syntactical terms in consideration were strings of characters, or letters. Later, Ehrig, et. al. [10] used pushouts to introduce rewriting for graphs. Graph rewriting was then axiomatised by Lack and Sobociński when they defined adhesive categories [13]. While we do not need the full generality of adhesive categories, we do use rewriting of topoi, which are examples of adhesive categories [14].

Central to rewriting is the concept of a grammar, which we define 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 with two monic legs. The idea is that r replaces ℓ and the common subobject k remains fixed. The grammar (X, P) induces a *derived grammar* (X, P') where the rules comprising

Date: December 18, 2018.

P' appear on the bottom row of a double pushout diagram of form

$$\begin{tikzcd} \ell \arrow[d, tail] & k \arrow[l, tail] \arrow[r, tail] \arrow[d, tail] \arrow[dl, phantom, "\urcorner"] \arrow[dr, phantom, "\ulcorner"] & r \arrow[d, tail] \\ g & d \arrow[l, tail] \arrow[r, tail] & h \end{tikzcd}$$

whose top row belongs to P. To study the grammar (X, P), we try to understand 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 thought of as systems together with appropriate morphisms. Our favorite example is the category RGraph of reflexive graphs. We like graphs because of their importance in network theory. We like reflexive graphs because the terminal object represents the nodes. Given a grammar (X, P), our goal is to rewrite systems x by decomposing it into subsystems, rewriting those, then connecting the results back together. Using structured cospans, we do this in a way that characterizes the rewriting relation for (X, P).

To do this, we turn to Gadducci and Heckle [12], who introduced an inductive perspective for graph rewriting. This mirrors our present goals so we follow the framework set in their paper. They define so called "ranked graphs" which are directed graphs with an interface, that is two sets of nodes, one set containing *input nodes* and the other containing *output nodes*. Using structured cospans, we are able to provide inputs and outputs to a much wider class of objects than graphs.

We make note that the terms 'input' and 'output' are not meant to imply causal structure.

To start, we need data beyond a topos 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 is still our topos of systems and A is a topos comprised of "interface types". These 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 properties of a geometric morphism arise throughout this article.

Given a geometric morphism $L \dashv R$, a *structured cospan* is a diagram in X of the form $La \to x \leftarrow Lb$. Think of this as consisting of a system x with inputs a and outputs b.

Baez and Courser introduced structured cospans [1], though to introduce rewriting, we need stronger hypotheses. They focused primary on the compositional structure, which the describe as follows. A structured cospan $La \to x \leftarrow Lb$ with outputs b can connect 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 ${}_L\mathsf{Csp}$ whose objects are from A and whose arrows are the structured cospans.

Observe that a system x with an empty interface can be encoded using the structured cospan $L0 \to x \leftarrow L0$. A local view of x is 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.

To study the rewriting of a system with interface, we can introduce a category where structured cospans are objects. Morphisms of structured cospans 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 ${}_L\mathsf{StrCsp}$. Our first result is about this category.

Theorem (2.5)**.** ${}_L\mathsf{StrCsp}$ *is a topos.*

This theorem allows us to introduce rewriting systems into structured cospans. As mentioned above, the rewriting system starts with grammars, though we restrict our attention to *structured cospan grammar*. This is a grammar $({}_L\mathsf{StrCsp}, P)$ where P has 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 $({}_L\mathsf{StrCsp}, P)$ the derived grammar $({}_L\mathsf{StrCsp}, P')$ 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 define another functor assigning a double category to $({}_L\mathsf{StrCsp}, P')$ whose objects are from A, vertical arrows are spans in A with invertible legs, horizontal arrows are structured cospans in ${}_L\mathsf{StrCsp}$, and whose squares are generated by the rules in P'. The *language functor* is composite Lang. As we see below, Lang encodes a rewriting relation on X inside the double category $\mathrm{Lang}({}_L\mathsf{StrCsp}, P)$.

Using a double category allows us to combine into a single structure the composability (horizontal composition) and rewritability (vertical composition) of structured cospans. The fact that this actually is a double category [8, Lem. 4.2] provides the interchange law, which ensures the compatibility of composing and rewriting systems. This compatibility 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 grammar $({}_L\mathsf{StrCsp}, \widehat{P})$. Here, our work begins to mirror that of Gadducci and Heckle. They use a well-known fact. Consider the sets $\{\ell \leftarrowtail k \rightarrowtail r\}$ and $\{\ell \leftarrowtail k_\flat \rightarrowtail r\}$ where $k_\flat$ is the discrete graph underlying k. These each have the same rewrite relation [10, Prop. 3.3].

We prove a generalized version of this result. To understand the following statement, denote by $P_\flat$ the set of rules obtained from $P = \{\ell \leftarrowtail k \rightarrowtail r\}$ by restricting the the span legs along the counit $LRk \to k$ 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 we can study the rewriting relation for $(\mathsf{X}, P_\flat)$ instead of the rewriting relation for (X, P).

Now, to decompose the systems in X and the rules in P, we associate to (X, P) the structured cospan grammar $({}_L\mathsf{StrCsp}, \widehat{P})$ 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 turns a system x in X into a structured cospan $L0 \to x \leftarrow L0$ with an empty interface and turns the rules from P into what we can think of as generators $\widehat{P}$ for a double category. The main result of the paper is that the rewriting relation for (X, P) is encoded inside 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 $\mathrm{Lang}({}_L\mathsf{StrCsp}, \widehat{P})$.

This theorem 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 structured cospan using rewriting rules. This will give a pasting diagram of the sort

$$\begin{array}{ccccccccccc}
L0 & \longrightarrow & x_1 & \leftarrow & La_1 & & La_{n-1} & \rightarrow & x_n & \longleftarrow & L0 \\
{\cong}\uparrow & & \downarrow & & \uparrow{\cong} & & {\cong}\uparrow & & \downarrow & & \uparrow{\cong} \\
L0 & \longrightarrow & x'_1 & \leftarrow & La'_1 & \cdots & La_{n-1} & \rightarrow & x'_n & \longleftarrow & L0 \\
{\cong}\downarrow & & \downarrow & & \downarrow{\cong} & & {\cong}\downarrow & & \downarrow & & \downarrow{\cong} \\
L0 & \longrightarrow & x''_1 & \leftarrow & La''_1 & & La_{n-1} & \rightarrow & x''_n & \longleftarrow & L0 \\
 & & \vdots & & & & & & \vdots & & \\
L0 & \longrightarrow & y_1 & \leftarrow & La_1 & & La_{n-1} & \rightarrow & y_n & \longleftarrow & L0 \\
{\cong}\uparrow & & \downarrow & & \uparrow{\cong} & & {\cong}\uparrow & & \downarrow & & \uparrow{\cong} \\
L0 & \longrightarrow & y'_1 & \leftarrow & La_1 & \cdots & La_{n-1} & \rightarrow & y'_n & \longleftarrow & L0 \\
{\cong}\downarrow & & \downarrow & & \downarrow{\cong} & & {\cong}\downarrow & & \downarrow & & \downarrow{\cong} \\
L0 & \longrightarrow & y''_1 & \leftarrow & La_1 & & La_{n-1} & \rightarrow & y''_n & \longleftarrow & L0
\end{array}$$

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. Here,we give our first main result: Theorem 2.5 which states that ${}_L\mathsf{StrCsp}$ 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, Theorem 3.2 which states 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, the 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]. Because structured cospans are a syntax, we want to set up a framework that can reflect semantics. We propose double pushout rewriting for this framework.

In this section, we set our hypotheses and explore structured cospans in this context. In particular, we see how, as morphisms, they encode compositional structure. We also see them as objects of a topos. Using double categories we combine the two perspectives.

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*. 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

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

Note that all categories in this paper have products and coproducts, so this notation is safe to use.

2.1. **Structured cospans as arrows.** After defining structured cospans, we look at their compositional structure. This material is in Baez and Courser's work [1], but we include it here for completeness.

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.

View X as a category of closed systems and their morphisms. By a *closed system*, we mean a system that cannot interact with the outside world. Think of A as a category of interfaces types and their morphisms. By transporting the interface types along L, they can be combined with the systems in X. A system is *open* when equipped with an interface. Open systems can interact with other compatible open systems. Finally, R returns the largest 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. Ignoring causality, we call La the input to x and Lb the output. In fact, this convention is arbitrary because ${}_L\mathsf{Csp}$ from Definition 2.2 is compact closed [8].

Definition 2.2. Denote by ${}_L\mathsf{Csp}$ 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

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

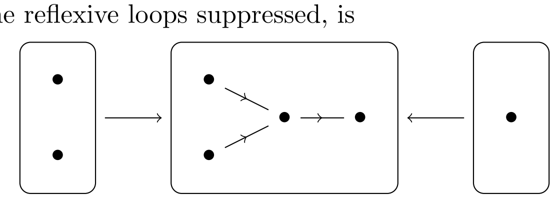
In a sense, pushouts glue objects together making it a sensible way to model system connection. The composition above is like connecting along Lb. We illustrate this with open graphs.

Example 2.3. Graph theory plays a central role in network theory. As such we take open graphs to be primary example of a structured cospan. While this notion is not new [9, 12], our infrastructure generalizes it.

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

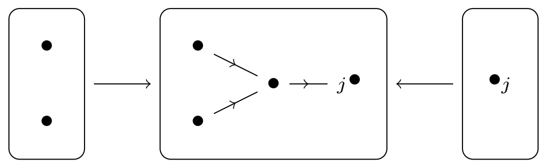
$$\begin{tikzcd} \mathsf{RGraph} \arrow[r, shift right, "R"'] \arrow[r, phantom, "\perp"] & \mathsf{Set} \arrow[l, shift right, "L"'] \end{tikzcd}$$

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

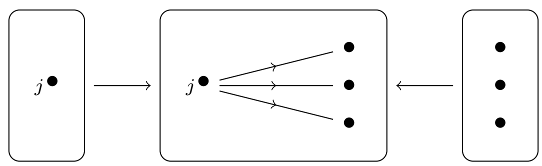


The boxed items are graphs and the arrows between boxes are graph morphisms defined as suggested by the illustration. In total, the three graphs and two graph morphisms make up a single open graph whose inputs and outputs are, respectively, the left and right-most graphs.

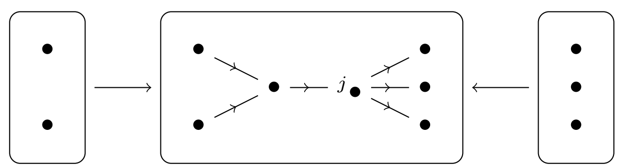
Open graphs are compositional. For instance, we can compose



with



to get the open graph



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

2.2. **Structured cospans as objects.** A morphism of open systems ought to respect the system plus its inputs and outputs.

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 ${}_L\mathsf{StrCsp}$.

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

Theorem 2.5. *For any geometric morphism $L \dashv R \colon \mathsf{A} \to \mathsf{X}$, the category ${}_L\mathsf{StrCsp}$ is a topos.*

Proof. Note that ${}_L\mathsf{StrCsp}$ 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 [16], hence a topos. $\square$

We now show that constructing ${}_L\mathsf{StrCsp}$ is functorial.

Theorem 2.6. *There is a functor*

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

defined by

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

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 2.5, 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. $\square$

The morphisms ${}_L\mathsf{StrCsp} \to_{L'} \mathsf{StrCsp}$ that we are interested in act on the systems and their interfaces.

Definition 2.7. 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 form a category which we leave unnamed.

2.3. **A double category of structured cospans.** We use (pseudo) double categories 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 [15], though we list the key components for the sake of completeness. A *(pseudo) double category* $\mathbb{C}$ is a category weakly internal to Cat, which roughly translates to a category of objects $\mathbb{C}_0$ together with a category of arrows $\mathbb{C}_1$ that assemble together as follows.

- The $\mathbb{C}_0$-objects are called the objects of $\mathbb{C}$.
- The $\mathbb{C}_0$-arrows are called the vertical arrows in $\mathbb{C}$.
- The $\mathbb{C}_1$-objects are called the the horizontal arrows in $\mathbb{C}$.
- The $\mathbb{C}_1$-arrows are called the squares of $\mathbb{C}$. are the arrows of $\mathbb{C}_1$.

See Figure 1 for a depiction of this data.

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

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 $\mathbb{C}_0$ and there is a structure map for composing horizontal arrows. The squares compose horizontally and vertically and, moreover, an interchange law ensures the proper interplay of these compositions.

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 2.8. There is a double category ${}_L\mathbb{C}\mathrm{sp} := (\mathsf{A}, {}_L\,\mathsf{StrCsp})$:

- 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 ${}_L\mathbb{C}\mathrm{sp}$. The 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 of a result about the expressiveness of certain graph grammars. We apply this rewriting theory to structured cospans. In doing so, we show the rewrite relation is functorial. This section also contains our main result which is a generalization of work by Gadducci and Heckle [12]. Yet, this result is not a mere generalization but provides a justification the study of systems using structured cospans.

Ehrig, et. al. [10] invented double pushout rewriting on graphs. 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 [13]. Topoi are examples of adhesive categories [14] so it follows that we can bring a theory of rewriting to structured cospans.

3.1. **Rewriting in 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 $k \to \ell \times r$ with the the caveat that, whenever we say that a span is a rule, we understand that both legs are monic. The conceit of a rule is that r replaces ℓ while the common subobject k remains fixed. 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

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

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 [13, 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*

$$\begin{tikzcd} \ell \arrow[d, tail] \arrow[dr, phantom, "\urcorner"] & k \arrow[l, tail] \arrow[r, tail] \arrow[d] \arrow[dr, phantom, "\ulcorner"] & r \arrow[d] \\ g & d \arrow[l] \arrow[r] & h \end{tikzcd}$$

The span $d \to g \times h$ is a rule because pushouts preserve monics in topoi [13, Lem. 12]. The intuition of this diagram is that $\ell \rightarrowtail g$ gives an instance of ℓ in g which r replaces, 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 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. But 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 multi-step rewrites. That is, there may be derived rules witnessing $g \rightsquigarrow g'$ and $g' \rightsquigarrow g''$ but not a derived rule witnessing $g \rightsquigarrow g''$. Yet, 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. So, the relation we actually want is the reflexive and transitive closure of $\rightsquigarrow$, which we denote by $\rightsquigarrow^*$. This, we call the **rewrite relation**. Every grammar determines a unique rewrite relation in a functorial way. Though, restrict our attention to showing this in the context of structured cospan categories.

3.2. **Expressiveness of underlying discrete grammars.** In this section, we generalize a result [10, Prop. 3.3] from the theory of rewriting graphs into the theory of rewriting in topoi.

The statement of this result, in our current context, begins with the underlying discrete graph comonad $\flat\colon \mathsf{RGraph} \to \mathsf{RGraph}$. The monic counit $\flat g \rightarrowtail g$ 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 settle a few things. 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 a 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

$$\begin{tikzcd} \ell \arrow[d] \arrow[dr, phantom, "\urcorner"] & k \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\ulcorner"] & r \arrow[d] \\ g & d \arrow[l] \arrow[r] & h \end{tikzcd}$$

arising from P, there is a derivation

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

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 apply rewriting in topoi to rewriting structured cospans which is possible because of Theorem 2.5.

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

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

and the morphisms are the structured cospan functors (Definition 2.7) that are stable under the grammars. The objects run through L and P.

Recall we associate a relation $\rightsquigarrow$ to each grammar, then take its reflexive and transitive closure to get the rewrite relation $\rightsquigarrow^*$. We now show that this is a functor 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 $D \colon \mathsf{StrCspGram} \to \mathsf{StrCspGram}$. On objects define $D({}_L\mathsf{StrCsp}, P)$ to be the grammar $({}_L\mathsf{StrCsp}, P')$, where P' consists of all rules $h \to g \times d$ witnessing the relation $g \rightsquigarrow h$ with respect to $({}_L\mathsf{StrCsp}, P)$. On arrows, define $DF \colon D({}_L\mathsf{StrCsp}, P) \to D({}_{L'}\mathsf{StrCsp}, Q)$ to be F. Moreover, the identity on $\mathsf{StrCspGram}$ is a subfunctor of D.*

Proof. That $D({}_L\mathsf{StrCsp}, P)$ actually gives a grammar follows from the fact that pushouts respect monics in a topos [13, Lem. 12].

To show that D is idempotent, we show that for any grammar $({}_L\mathsf{StrCsp}, P)$, we have $D({}_L\mathsf{StrCsp}, P) = DD({}_L\mathsf{StrCsp}, P)$. Rules in $DD({}_L\mathsf{StrCsp}, P)$ are the bottom row of a double pushout diagram whose top row is a rule in $D({}_L\mathsf{StrCsp}, P)$, which in turn is the bottom row of a double pushout diagram whose top row is in $({}_L\mathsf{StrCsp}, P)$. Thus, a rule in $DD({}_L\mathsf{StrCsp}, P)$ is the bottom row of a double pushout diagram whose top row is in $({}_L\mathsf{StrCsp}, P)$. See Figure 2.

$$\begin{tikzcd} g \arrow[d] \arrow[dr, phantom, "\urcorner"] & d \arrow[l] \arrow[r] \arrow[d] & h \arrow[d] \arrow[dl, phantom, "\ulcorner"] \\ g' \arrow[d] \arrow[dr, phantom, "\urcorner"] & d' \arrow[l] \arrow[r] \arrow[d] & h' \arrow[d] \arrow[dl, phantom, "\ulcorner"] \\ g'' & d'' \arrow[l] \arrow[r] & h'' \end{tikzcd}$$

FIGURE 2. Stacked double pushout diagrams

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

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 is derivable from the original rule. That identity is a subfunctor of D means that the derived grammar contains all the rules of the original grammar.

To define S, we reference the double category $\mathbb{M}\text{onSpCsp}(\mathsf{C})$ for a topos C introduced in [8]. 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 $({}_L\mathsf{StrCsp}, P)$, observe that the productions in P are admissible as squares in $\mathbb{M}\text{onSpCsp}(\mathsf{X})$. Denote by $S({}_L\mathsf{StrCsp}, P)$ the sub-double category of $\mathbb{M}\text{onSpCsp}(\mathsf{X})$ that is full on objects, vertical and horizontal arrows, and generated by the rules in P. This assignment is functorial because

$$(F, G)\colon ({}_L\mathsf{StrCsp}, P) \to (StrCsp, P)$$

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

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

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

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

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

realize $x + x' \rightsquigarrow^* y + y'$ 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 main result ensures 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 we decompose a system into sub-systems which are each studied then reconnected along the interfaces that structured cospans provide. We discuss the manner in which the main result does this below the theorem, for which we need the following definition.

Definition 3.5. Associate to a grammar (X, P) the structured cospan grammar $({}_L\mathsf{StrCsp}, P')$ 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 $P_\flat$.

Before stating the theorem, we note that this is a generalization of work by Gadducci and Heckle [12] 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 $\mathrm{Lang}({}_L\mathsf{StrCsp}, P')$.

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

$$\begin{tikzcd}
\ell \arrow[d] \arrow[dr, phantom, "\urcorner"] & LRk \arrow[l] \arrow[r] \arrow[d] \arrow[dr, phantom, "\ulcorner"] & r \arrow[d] \\
g & d \arrow[l] \arrow[r] & h
\end{tikzcd}$$

the desired square is the horizontal composition of

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

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 $\mathrm{Lang}({}_L\mathsf{StrCsp}, P')$. 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 $\mathrm{Lang}({}_L\mathsf{StrCsp}, P')$ 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 $\mathrm{Lang}({}_L\mathsf{StrCsp}, P')$ 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

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

The composite of these squares is a rewriting of the original system.

References

[1] J. Baez, K. Courser. Structured cospans. *In preparation.* (Referred to on page 2, 5, 6, 9.)

[2] J. Baez, B. Fong. A compositional framework for passive linear networks. *Theory Appl. Categ.* 33, No. 38, 1158–1222. 2018 (Referred to on page 1.)

[3] J. Baez, B. Fong, B. Pollard. A compositional framework for Markov processes. *J. Math. Phys.* 57, No. 3: 033301. 2016. (Referred to on page 1.)

[4] J. Baez, B. Pollard. A compositional framework for reaction networks. *Rev. Math. Phys.* 29, No. 9, 1750028. 2017. (Referred to on page 1.)

[5] J. Baez, J. Master. Open Petri nets. *arXiv preprint* arXiv:1808.05414. 2018. (Referred to on page 1.)

[6] N. Chomsky. On certain formal properties of grammars. *Inf. Control.* No. 2. 137-167. 1959. (Referred to on page 1.)

[7] D. Cicala. Spans of cospans. *Theory Appl. Categ.* 33, No. 6, 131–147. 2018. (Referred to on page 1.)

[8] D. Cicala, K. Courser. Spans of cospans in a topos. *Theory Appl. Categ.* 33, No. 1, 1–22. 2018. (Referred to on page 3, 6, 13.)

[9] L. Dixon, A. Kissinger. Open-graphs and monoidal theories. *Math. Structures Comput. Sci.*, **23**, No. 2, 308–359. 2013. (Referred to on page 6.)

[10] H. Ehrig, M. Pfender, 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. (Referred to on page 1, 3, 10, 11.)

[11] B. Fong. Decorated cospans. *Theory Appl. Categ.* 30, Paper No. 33, 1096–1120. 2015. (Referred to on page .)

[12] F. Gadducci, R. Heckel. An inductive view of graph transformation.*International Workshop on Algebraic Development Techniques*. 223–237. Springer. 1998. (Referred to on page 2, 6, 10, 14.)

[13] S. Lack, P. Sobociński. Adhesive categories. In *International Conference on Foundations of Software Science and Computation Structures*, 273–288. Springer. 2004. (Referred to on page 1, 10, 12.)

[14] S. Lack, P. Sobociński. Toposes are adhesive. *International Conference on Graph Transformation*. Springer. 2006. (Referred to on page 1, 10.)

[15] M. Shulman. Constructing symmetric monoidal bicategories. *arXiv preprint* arXiv:1004.0993 2010. (Referred to on page 9.)

[16] G. Wraith. Artin gluing. *J. Pure Appl. Algebra* **4**, 345–348. 1974. (Referred to on page 8.)

Email address: `cicala@math.ucr.edu`

Department of Mathematics, University of California, Riverside, Riverside, California 92521