COMPOSITIONAL REWRITING WITH 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 $L \dashv R\colon \mathrm{X} \to \mathrm{A}$. 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

Rewriting has a fairly remarkable history. Beginning in the confines of linguistics, Noam Chomsky [7] introduced the concept to study formal, as opposed to natural, languages. As the theory of rewriting evolved, two distinct perspective emerged. The first was an operational perspective. An operational rewriter would study by imposing a fixed number of rules, collectively called a *grammar*, to a collection of strings. These rules dictate how one can remove a sub-string and put in its place another string, thus resulting in a new "rewritten" string. The other perspective was inductive. Here, one begins with a set of "basic rules" which combine in various ways to make new rules. The semantics of rewriting is given by the aptly named "rewriting relation", defined by relating $x \rightsquigarrow^* y$ if in a finite number of steps, x can be rewritten into y via the given rules.

The next stage involved the algebraic rewriting introduced by Erhig, et. al. This stage, the so called *graph rewriting*, is characterized by moving from rewriting of one-dimension strings to two-dimensional directed multi-graphs (henceforth, graphs) [11]. There are several formalisms used to describe graph rewriting, but we only concern ourselves with the double pushout style. In this method, one starts with a set of *productions*, or rules, of graphs. These are monic legged spans of graphs. A *rewrite relation* is assigned to each set of productions, or *grammar*. This is the reflexive and transitive closure of the relation $g \rightsquigarrow h$ defined whenever there is a production $\ell \hookleftarrow k \rightarrowtail r$, a mono $m\colon \ell \rightarrowtail g$, and a double pushout diagram

$$\begin{array}{ccccc} \ell & \hookleftarrow & k & \rightarrowtail & r \\ {\scriptstyle m}\downarrow & & \downarrow & & \downarrow \\ g & \longleftarrow & k & \longrightarrow & h \end{array}$$

The operational viewpoint dominated in the theory of graph rewriting. Then Gadducci and Heckel introduced an inductive viewpoint [13]. They formalize "ranked graphs" as arrows in a category where the ordinary graphs are the endomorphisms on the initial object. A ranked graph is essentially a graph with a 3-coloring on its nodes. The colors are 'input', 'output', and 'interior'. A graph with n input nodes and m output nodes is thought of an arrow in a prop from n to m. The composition of ranked graphs is made by gluing the inputs of one graph to the outputs of another. Then given a graph grammar, one then introduces a number of 2-cells into this category in a specific way that draws a correspondence between the rewrite relation and the arrows of the hom-category $\hom(0,0)$.

The next stage in the theory of rewriting is marked by the introduction by Lack and Sobociński [15] of "adhesive categories". These give the most elegant axiomatization of graph rewriting to date. Currently, there is only a operation perspective on adhesive rewriting. The aim of this paper is to build on the ideas of Gadducci and Heckel to give an inductive viewpoint on a fragment of the theory of adhesive rewriting.

This project fits into a larger program to codify network theory into the language of category theory [6, 5, 3, 2, 4]. While graphs and similar objects play an important role in classical network theory, we hope to move beyond such combinatorial descriptions to understand what networks essentially are. While practitioners in many fields use these graph-like objects to various ends, by abstracting away the idiosyncrasies of each instance, we can study the objects themselves.

A guiding philosophy in this program is the use of *functorial semantics*. That is, we construct a syntax whereby systems are arrows in a category, thus allowing for their composition which creates larger systems. The semantics of such systems is captured by a functor on our syntax category from which we study properties of the system preserved by composition; the so-called *compositional properties*. Compositionality is desirable in the study of complex systems because global behaviors are determined by local (read "easier to understand") behaviors. One source of syntax categories is Fong's "decorate cospans" [12]. Another source, the one introduced here and by Baez and Courser [1] are the *structured cospan categories*.

A structured cospan builds from a pushout preserving functor $L\colon \mathrm{A} \to \mathrm{X}$ between categories with pushouts. The idea is that the category X contains the system types, the category A contains the interface types, and the functor L turns the interface types into system types. A structured cospan is just a cospan $La \to x \leftarrow Lb$ in the category X. The feet of the cospan can be thought of as identifying subsystems La and Lb of x to serve as the input and output. The compositional nature of structured cospans can be seen if given another structured cospan $Lb \to y \leftarrow Lc$ whose input matched the output of the other. These compose as cospans typically do, giving $La \to x +_{Lb} y \leftarrow Lc$.

In light of the well-developed theory or rewriting that exist for the various graphs so prevalent in network theory, we intend to bring rewriting into our structured cospan framework. This brings us to the theory of adhesive categories which requires stronger hypotheses than those given in the above definition of a structured cospan. Specifically, begin with a geometric morphism, that is an adjunction $(L\colon \mathrm{A} \to \mathrm{X}) \dashv (R\colon \mathrm{X} \to \mathrm{A})$ between elementary topoi such that L preserves finite limits. In the systems analogy, L, X, and A play the same role as before and the compositional structure is still there. However, we also pursue the perspective of

structured cospans as objects between which arrows exist as commuting diagrams

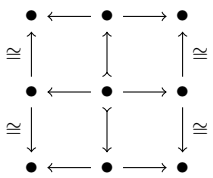
$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & La' \\ \downarrow & & \downarrow & & \downarrow \\ Lb & \longrightarrow & y & \longleftarrow & Lb' \end{array} \tag{1}$$

giving a category we denote as StrCsp_L. Requiring that L be a finitely continuous left adjoint ensures that StrCsp_L is a topos, hence adhesive [16]. This means that we can introduce the theory of rewriting to structured cospans.

As mentioned above, the operational perspective pervades adhesive rewriting. However, this perspective does not parallel the compositionality present in the structured cospan formalism. The inductive viewpoint, on the other hand, fits much more nicely into the program of network theory as functorial semantics. Both structural induction and compositionality involve building systems from pieces. This is the motivation for our main result: lifting the inductive viewpoint of graph rewriting into the context of structured cospans.

The structure of the paper is as follows. Section 2 defines structured cospans, the main object of study. There are two perspectives on structured cospans and we realize this by building two separate categories. The first category StrCsp_L, mentioned above, has structured cospans as objects and commuting diagrams (1) as arrows. The first result, and the keystone of the paper, is that StrCsp_L is a topos. This grants us access to the theory of adhesive rewriting. The second category, and the one introduced by Baez and Courser [1] views structured cospans as arrows. We then combine these two perspectives into a double category.

In Section 3, we sweep over the basics of graph rewriting and cover the inductive viewpoint. We also recall the theory of adhesive categories before applying it to structured cospans in a categorical framework. We start with *grammars*. There are two types of grammars we are interested in. The first involves pairing an adhesive category C with a set of productions P inside C. These form a category. There is a subcategory we use too, consisting of grammars (StrCsp_L, P) where P is a set of spans in StrCsp_L of the form



and fitting them into a category StrCspGram. Given a grammar (C, P), we define the rewriting relation on the objects of C using double pushouts in a manner similar to that as was done for graphs. However, we provide a functorial characterization as well. Namely, the functor $D\colon \mathrm{Gram} \to \mathrm{Gram}$ that send a grammar (C, P) to the grammar (C, P') where a production $g \leftarrow d \rightarrow h$ belongs to P' if there is a double pushout diagram

$$\begin{array}{ccccc} \ell & \longleftarrow & k & \longrightarrow & r \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ g & \longleftarrow & d & \longrightarrow & h \end{array}$$

such that the top row is a production in P. We then define a semantics functor $S\colon \mathrm{Gram} \to \mathrm{DblCat}$ valued in double categories. These semantics capture the compositionality of the structured cospans and also the rewriting. The *language* of a grammar is its image under the composite functor SD. Our main result is Theorem 3.11. Given a grammar (X, P) on a topos X fitting into a suitable geometric morphism $L \dashv R\colon \mathrm{X} \to \mathrm{A}$, then we construct a grammar (StrCsp_L, Q) such that g is related to h from the rewriting relation on the grammar (X, P) exactly when there is a square in the double category $SD(\mathrm{StrCsp}_L, Q)$ of the form

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

Simply stated, if X contains a type of network, then the rewritings on networks of type X can be understood by rewriting on pieces of the network inside of (StrCsp, Q) then gluing them back together.

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

2. Structured Cospans

Structured cospans are introduced here, as well as by Baez and Courser [1]. Their purpose is to provide syntax for compositional systems. Baez and Courser's work has two aims: maximize the generality of the structured cospan construction using double categories and also to compare structured cospans to Fong's decorated cospans [12]. Our interests here are instead to discuss introducing rewriting structured cospans. This requires us to make restrictions not needed by Baez and Courser. These restriction are harmless, however, as most cases of interest fall within our parameters.

In this section, we make explicit two perspectives on structured cospans. The first is looking at structured cospans as objects of a category with appropriate morphisms between them. The second perspective takes structured cospans to be morphisms between "interfaces". The latter perspective encode the compositional structure.

For this section, fix an arbitrary geometric morphism $L \dashv R\colon \mathrm{X} \to \mathrm{A}$. This is an adjunction

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

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

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

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

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

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

2.1. **Structured cospans as objects.**

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

There is no novelty in a simple cospan, but we use this language to contextualize our work. Also, the purpose behind the restriction to a geometric morphism is not currently clear. Let us say that this requirement arises from practical and aesthetic considerations. For one, a nicer theory develops, but also it is a sufficiently strong condition to introduce rewriting of structured cospans.

Given that we are motivated by open systems, beginning by fixing geometric morphism may signal the abstract nature of this work and cloud the connection between structured cospans and open systems. So that we do not drift too far from our concrete motivations, we will draw analogies between our construction and open systems throughout.

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

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

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

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

It is a simple exercise to show that structured cospans and their morphisms form a category. Denote this category by StrCsp_L.

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

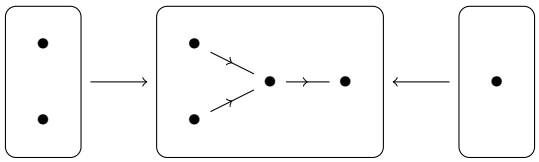
Let

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

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

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

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



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

Having seen this example, it becomes more apparent about how open systems can "connect" together. Given another open graph whose inputs coincide with the outputs of the graph above, we can connect the inputs and outputs together to create a new open graph. By passing from graphs to open graphs, we are introducing *compositionality*. The category StrCsp_L does not encode the compositional structure, but we introduce a new category Cospan_L in Section **??**. Now, we show that StrCsp_L is a topos in a functorial way. While interesting in itself, this fact allows us to introduce rewriting systems to structured cospans.

Theorem 2.4. *The category* StrCsp_L *is a topos.*

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

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

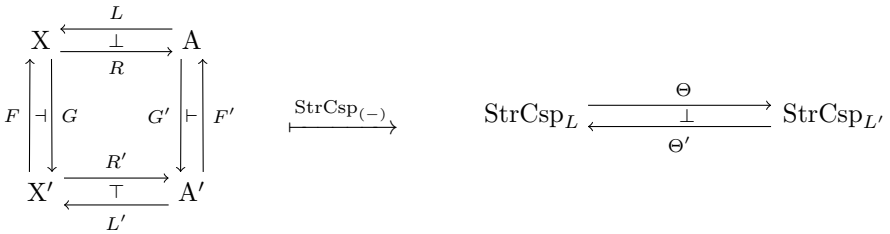
This, in turn, is equivalent to the comma category $(\mathrm{A} \times \mathrm{A} \downarrow \Delta R)$, where $\Delta \colon \mathrm{A} \to \mathrm{A} \times \mathrm{A}$ is the diagonal functor. But the diagonal functor is right adjoint to taking binary coproducts. That means ΔR is also a right adjoint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

Better yet, the construction $\mathrm{StrCsp}_{(-)}$ is actually functorial. For the following theorem, denote by Topos the category of elementary topoi and geometric morphisms.

Theorem 2.5. *There is a functor*

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

defined by



which is in turn given by

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

and

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

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

Denote the structured cospans

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

in StrCsp_L by ℓ and

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

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

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

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

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

Even though StrCsp_L is actually a topos in our context, the general theory of structured cospans is not beholden to toposity, meaning that morphisms $\mathrm{StrCsp}_L \to \mathrm{StrCsp}_{L'}$ are not geometric morphisms. Instead, we define a **structured cospan functor** to be a pair of finitely continuous and cocontinuous functors $F\colon \mathrm{X} \to \mathrm{X}'$ and $G\colon \mathrm{A} \to \mathrm{A}'$ such that $FL = L'F$ and $GR = R'F$. Structured cospan categories and their morphisms form a category, but we leave it unnamed.

2.2. **Structured cospans as arrows.** We now turn to capturing the compositional structure that truly motivates the invention of structured cospans. To do this, we shift perspectives of structured cospans as objects in StrCsp_L to structured cospans as morphisms.

Definition 2.6. Denote by Cospan_L the category that has the same objects as A and structured cospans $La + Lb \to x$ as arrows of type $a \to b$.

Note that composition is defined by pushout:

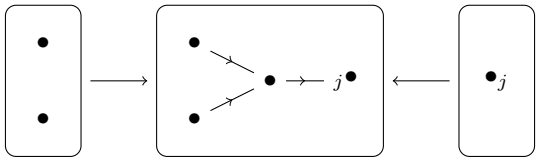
$$La \to x \leftarrow Lb \quad ; \quad Lb \to y \leftarrow Lc \quad \overset{\circ}{\mapsto} \quad La \to x +_{Lb} y \leftarrow Lc$$

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

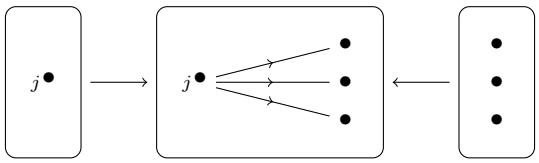
$$La + Lb \to x \quad \text{and} \quad Lb + Lc \to y$$

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

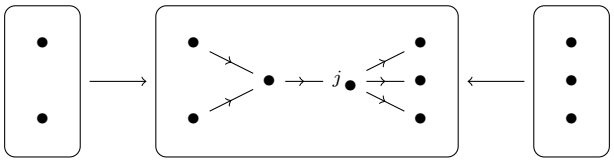
Example 2.7. The open graph



can be composed with the open graph



to obtain



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

2.3. **A double category of structured cospans.** Using double categories allows us to combine into a single instrument the competing perspectives of structured cospans as objects and as morphisms. For a precise definition of a symmetric monoidal double category, we point to Shulman [17], though for the sake of completeness, we list the key pieces. A double category $\mathbb{C}$ consists of a pair of categories $(\mathrm{C}_0, \mathrm{C}_1)$ with some additional data that binds them together. The categories C_0 and C_1 are assemble together into a double category as follows:

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

$$\begin{array}{ccc} c & \xrightarrow{m} & d \\ {\scriptstyle f}\downarrow & \Downarrow\ \theta & \downarrow{\scriptstyle g} \\ c' & \xrightarrow[n]{} & d' \end{array} \qquad \begin{array}{c} c, c', d, d' \in \mathrm{ob}(\mathrm{C}_0) \\ f, g \in \mathrm{arr}(\mathrm{C}_0) \\ m, n \in \mathrm{ob}(\mathrm{C}_1) \\ \theta \in \mathrm{arr}(\mathrm{C}_1) \end{array}$$

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

The vertical arrows compose as they do in C_0 and there is a structure map for composing horizontal arrows. The squares can compose both horizontally and vertically.

Observe that the horizontal arrows play two roles: as objects in their origin category and arrows in the double category. This reflects the content of the categories StrCsp_L and Cospan_L.

The first double category that we define is discussed in the related work by Baez and Courser [1, Cor. 3.9]. There, they prove that it actually is a double category, so we content ourselves to simply provide the definition.

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

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

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

Actually, $\mathbb{S}\mathrm{trCsp}_L$ is a symmetric monoidal double category. Since all involved categories are topoi, we can lift their cocartesian structure to serve as a tensor on the double category. However, we have no need for this structure, so we say no more about it.

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

3. Inductive Rewriting of Structured Cospans

In this final section, we recall the basics of graph rewriting and the theorem that first provided its inductive viewpoint. We then extend this to rewriting in a wider class of objects using the structured cospan formalism.

3.1. **Graph rewriting.** Graph rewriting is a well-studied field with many suitable references [11, 14]. To be self-contained, we present enough of the topic to facilitate a connection with our main result.

We start our discussion of graph rewriting with the notion of a **production**: a span of graphs

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

with monic legs. Together with a **matching morphism**, that is a mono $\ell \rightarrowtail g$, this production behaves by removing from g the copy of ℓ and replacing it with r. The removal of ℓ from g is formalized as the **pushout complement**. This is a graph d that fits into a pushout diagram

$$\begin{array}{ccc} \ell & \longrightarrow & k \\ {\scriptstyle m}\downarrow & \text{p.o.} & \downarrow \\ g & \longrightarrow & d \end{array}$$

Certainly, d need not exist but when it does and $k \to \ell$ is monic, then d is unique up to isomorphism.

If P is a set of productions, often called a **graph grammar**, there is an induced relation $\rightsquigarrow$ on the objects of the category RGraph. This is defined by $g \rightsquigarrow h$

whenever there is a production $k \to \ell \times r$ in P, a match $\ell \to g$, and a pushout complement d that fit into the double pushout diagram

$$\begin{array}{ccccc} \ell & \longleftarrow & k & \longrightarrow & r \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ g & \longleftarrow & d & \longrightarrow & h \end{array}$$

The content of this diagram is an identification of a subgraph of shape ℓ inside of g, its removal and subsequent replacement in a coherent way with r, finally resulting in the graph h. If $g \rightsquigarrow h$, we say that h is a **direct derivation** of g. The term "direct" refers to the derivation happening in a single step. In general, $\rightsquigarrow$ is not transitive, meaning that this relation does not tell us about (multi-step) derivations. It is also not reflexive, which is preferable for a nicer theory. Therefore, it is customary to work with the so-called **rewrite relation**, that is the reflexive and transitive closure $\rightsquigarrow^*$. Indeed, to any graph grammar P, there is an induced rewrite relation.

In term rewriting, there are two different viewpoints. The *operational* viewpoint can be concisely described as performing substitution. That is, we rewrite a term by substituting subterms with other terms according to a rule. This is the viewpoint of the graph rewriting discussed above. On the other hand, the *inductive* viewpoint consists of building a class of rewritings from a collection of basic ones. This is the viewpoint that Gadducci and Heckel first brought to graph rewriting first in [13] and we introduce to adhesive rewriting in Theorem 3.11.

To prove their result, Gadducci and Heckel made use of the following well-known fact: the graph grammars $k_\alpha \to \ell_\alpha \times r_\alpha$ and $(k_\alpha)_\flat \to \ell_\alpha \times r_\alpha$, where $(k_\alpha)_\flat$ is the discrete graph underlying k_α, induce the same rewrite relation [11, Prop. 3.3]. Lemma 3.10 generalizes this result into our context. Gadducci and Heckel leveraged this fact to construct a computad using a graph grammar P. The underlying category of this computad is Cospan_L where L is the discrete graph functor from Example 2.3. We additionally add a 2-cell

$$L\emptyset \;\; \overset{\hat{\ell}}{\underset{\hat{r}}{\Downarrow \gamma}} \;\; Lk$$

into the computad for every production $k \to \ell \times r$ in P. By $\hat{\ell}$, we mean the open graph $L\emptyset + Lk \to \ell$ and the analogous open graph for $\hat{r}$. Note the use of the discrete graph underlying k. Here, we translated Gadducci and Heckel's work into the language of structured cospans. This computad then freely generates a 2-category which we call G, the 1-arrows of which are open graphs. The usual closed graphs are exactly the objects of the hom-category $\mathrm{G}(L\emptyset, L\emptyset)$. This hom-category captures the rewriting relation as we state here [**?**, Thm. 23].

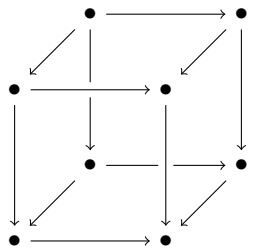
Theorem 3.1. *Let P be a graph grammar. Then $g \rightsquigarrow^* h$ if and only if there is a 2-arrow from $L\emptyset + L\emptyset \to g$ to $L\emptyset + L\emptyset \to h$ in $\mathrm{G}(L\emptyset, L\emptyset)$.*

Having shared a viewpoint on inductive graph rewriting, we now turn to apply these ideas to inductive adhesive rewriting using structured cospans.

3.2. **Adhesive rewriting.** There have been a number attempts at axiomatizing graph rewriting. Adhesive categories, introduced by Lack and Sobociński [15], is

a particularly elegant solution and the one we use presently. After recalling the basics of DPO rewriting in adhesive categories, we use it to introduce the rewriting theory of structured cospans.

Adhesivity is an exactness condition that ensures certain pushouts behave as they do in Set. This is captured by saying that a pushout satisfies the **Van Kampen condition**: given a commuting cube



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

Definition 3.2. A category with pullbacks is **adhesive** if pushouts along monics exist and are Van Kampen.

An adhesive category is a place where rewrite systems retain important properties of DPO graph rewriting, namely concurrency. The definitions above—production, matching morphism, pushout complement, grammar, direct derivation, derivation, and rewrite relation—all have their adhesive rewriting counterparts. The primary difference is that they occur in a generic adhesive category.

Rewriting in RGraph had the benefit of being able to describe a production's behavior by referring to edges and nodes. While we lose this ability having abstracted to adhesive categories, the intuition of removing a subobject and coherently replacing it with another is helpful.

It is still true that pushout complements, if they exist, over a mono are unique up to isomorphism [15, Lem. 15]

For improved bookkeeping, we pair a grammar with the adhesive category housing the productions. Specifically, a production is a pair (C, P) where C is an adhesive category and P is a set of productions in C. This pairing allows us to define the following category.

Definition 3.3. There is a category Gram having

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

As in graph rewriting, any grammar (C, P) induces a relation on the objects of C defined by $g \rightsquigarrow h$ whenever there is a production $d \to g \times h$ fitting into the bottom row of a double pushout diagram

$$\begin{array}{ccccc} \ell & \longleftarrow & k & \longrightarrow & r \\ {\scriptstyle m}\downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ g & \longleftarrow & d & \longrightarrow & h \end{array}$$

such that the top row is a production in P and m is monic. In fact, this relation arises from a functor as shown in the following lemma.

Lemma 3.4. *There is an idempotent functor* $D\colon \mathrm{Gram} \to \mathrm{Gram}$*, called the* ***direct derivation functor****, defined on objects by setting* $D(\mathrm{C}, P)$ *to be the grammar in* C *consisting of all productions* $h \to g \times d$ *that witness the relation* $g \rightsquigarrow h$ *with respect to* (C, P)*. On arrows,* $DF\colon D(\mathrm{C}, P) \to D(\mathrm{D}, Q)$ *is defined exactly as* F*. Moreover, the identity on* Gram *is a subfunctor of* D*.*

Proof. That $D(\mathrm{C}, P)$ actually gives a grammar follows from the fact that pushouts respect monics in an adhesive category [15, Lem. 12].

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

$$\begin{array}{ccccc} \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet \end{array}$$

is a pushout.

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

As in the graph rewriting case, we are actually interested in the rewriting relation $\rightsquigarrow^*$ defined to be the reflexive transitive closure of $\rightsquigarrow$. A useful fact that we use in our main result is that the rewriting relation plays well with coproducts.

Lemma 3.5. *If* $x \rightsquigarrow^* y$ *and* $x' \rightsquigarrow^* y'$*, then* $x + x' \rightsquigarrow^* y + y'$

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

$$\begin{array}{ccccccccc} \ell_1 & \longleftarrow & k_1 & \longrightarrow & r_1 & \quad \ell_2 & \longleftarrow & k_2 & \longrightarrow & r_2 & \quad\cdots\quad & \ell_n & \longleftarrow & k_n & \longrightarrow & r_n \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \searrow & \swarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow & & \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ x & \longleftarrow & d_1 & \longrightarrow & w_1 & \longleftarrow & d_2 & \longrightarrow & w_2 & & & w_{n-1} & \leftarrow & d_n & \longrightarrow & y \end{array}$$

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

$$\begin{array}{ccccccccc} \ell'_1 & \longleftarrow & k'_1 & \longrightarrow & r'_1 & \quad \ell'_2 & \longleftarrow & k'_2 & \longrightarrow & r'_2 & \quad\cdots\quad & \ell'_n & \longleftarrow & k'_m & \longrightarrow & r'_m \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \searrow & \swarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow & & \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ x' & \longleftarrow & d'_1 & \longrightarrow & w'_1 & \longleftarrow & d'_2 & \longrightarrow & w'_2 & & & w'_{m-1} & \leftarrow & d'_m & \longrightarrow & y' \end{array}$$

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. □

We can obtain $\rightsquigarrow^*$ functorially by viewing the productions as arrows in a span category.

Lemma 3.6. *There is a functor* $S\colon \mathrm{Gram} \to \mathrm{Cat}$*, defined on objects by setting* $S(\mathrm{C}, P)$ *to be the subcategory of* $\mathrm{Span}(\mathrm{C})$ *generated by the productions in* P *and on arrows by setting* $SF\colon S(\mathrm{C}, P) \to S(\mathrm{D}, Q)$ *to be the functor given by* $SFc = Fc$ *and*

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

For expositional purposes, we will denote the composite functor SD by Lang and call $\mathrm{Lang}(\mathrm{C}, P)$ the **language** of (C, P). The language is another characterization of the rewrite relation.

Theorem 3.7. *$g \rightsquigarrow^* h$ with respect to a grammar (C, P) if and only if there is an arrow $g \to h$ in $\mathrm{Lang}(\mathrm{C}, P)$.*

Proof. If $g \rightsquigarrow^* h$, then there is a zig-zag from g to h comprised of productions in $D(\mathrm{C}, P)$ and identity spans. This zig-zag gives a sequence of composable arrows in $\mathrm{Lang}(\mathrm{C}, P)$. Conversely, any arrow $g \to h$ in $\mathrm{Lang}(\mathrm{C}, P)$ is a composite of generating arrows. □

3.3. **Rewriting structured cospans.** Having built up the necessary infrastructure, we now turn to rewriting structured cospans. To do this we need structured cospans to form an adhesive category. Theorem 2.4 together with the fact that topoi are adhesive [16] gives the following result.

Theorem 3.8. *The category StrCsp_L is adhesive for any geometric morphism $L \dashv R$.*

This ensures that we are within reasonable constraints to define rewriting on structured cospans. The language $\mathrm{Lang}(\mathrm{StrCsp}_L, P)$ is a subcategory of $\mathrm{Span}(\mathrm{StrCsp}_L)$ generated by spans of structured cospans, that is commuting diagrams of form

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

This is an arrow from $La + La' \to x$ to $Lc + Lc' \to z$. In this span category, we can compose productions, but not structured cospans. To accommodate both compositions, we use the theory of double categories.

To use double categories, we need to restrict the class of grammars that we work with. Fortunately, the resulting language is structurally richer.

Definition 3.9. The category of **structured cospan grammars** StrCspGram is a subcategory of Gram. The objects are (StrCsp_L, P) where P consists of productions of the form

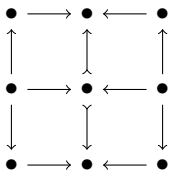
$$\begin{array}{ccccc} La & \longrightarrow & x & \longleftarrow & La' \\ \cong\uparrow & & \uparrow & & \uparrow\cong \\ Lb & \longrightarrow & y & \longleftarrow & Lb' \\ \cong\downarrow & & \downarrow & & \downarrow\cong \\ Lc & \longrightarrow & z & \longleftarrow & Lc' \end{array}$$

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

Following our work in adhesive rewriting, we define a language for the structure cospan grammars. However, instead of a category-valued semantics, we provide a double category semantics. Specifically, the language functor is the composite of two

functors: $D\colon \mathrm{StrCspGram} \to \mathrm{StrCspGram}$, which is a restriction of the associated functor defined in Section 3.1 and the functor $S\colon \mathrm{StrCspGram} \to \mathrm{DblCat}$.

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



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

double check this

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

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

For the following theorem, we denote by $(\mathrm{C}, P_\flat)$ the **discrete adhesive grammar** underlying the adhesive grammar (C, P). The discrete adhesive grammar consists of productions $LRK \to k \to \ell \times r$ for each $k \to \ell \times r$ in P.

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

Lemma 3.10. *Fix a geometric morphism $L \dashv R\colon \mathrm{X} \to \mathrm{A}$ 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 an adhesive grammar (X, P) is equal to rewriting relation for the discrete adhesive grammar $(\mathrm{X}, P_\flat)$*

Proof. For any derivation

$$\begin{array}{ccccc} \ell & \longleftarrow & k & \longrightarrow & r \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ g & \longleftarrow & d & \longrightarrow & h \end{array}$$

arising from P, there is a derivation

$$\begin{array}{ccccccccc} \ell & \longleftarrow & k & \leftarrow & LRk & \rightarrow & k & \longrightarrow & r \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ g & \longleftarrow & d & \longleftarrow & w & \longrightarrow & d & \longrightarrow & h \end{array}$$

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. □

Let us now connect rewriting structured cospans to rewriting in a topos. Fix a geometric morphism $L \dashv R \colon \mathrm{X} \to \mathrm{A}$ and grammar (X, P) satisfying the conditions of Lemma 3.10. Associate to (X, P) a structured cospan grammar (StrCsp_L, P') consisting of a production

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

for every production $LRk \to \ell \times r$ of $P_\flat$. The main theorem encodes the rewrite relation from (X, P) in the double category $\mathrm{Lang}(\mathrm{StrCsp}_L, P')$.

Theorem 3.11. *Fix a geometric morphism $L \dashv R \colon \mathrm{X} \to \mathrm{A}$ and grammar (X, P) satisfying the conditions of Lemma 3.10. Then $g \rightsquigarrow^* h$ in the rewriting relation for an grammar (X, P) if and only if there is a square*

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

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

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

$$\begin{array}{ccccc} \ell & \leftarrow & LRk & \to & r \\ \downarrow & \text{p.o.} & \downarrow & \text{p.o.} & \downarrow \\ g & \longleftarrow & d & \longrightarrow & h \end{array}$$

the desired square is the horizontal composition of

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

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

$$\begin{array}{ccccc} L0 & \longrightarrow & g & \longleftarrow & L0 \\ \uparrow & & \uparrow & & \uparrow \\ L0 & \longrightarrow & d & \longleftarrow & L0 \\ \downarrow & & \downarrow & & \downarrow \\ L0 & \longrightarrow & h & \longleftarrow & L0 \\ \uparrow & & \uparrow & & \uparrow \\ L0 & \longrightarrow & e & \longleftarrow & L0 \\ \downarrow & & \downarrow & & \downarrow \\ L0 & \longrightarrow & j & \longleftarrow & L0 \end{array}$$

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

$$\begin{array}{ccccc} L0 & \longleftarrow & w & \longrightarrow & L0 \\ \uparrow & & \uparrow & & \uparrow \\ L0 & \longleftarrow & x & \longrightarrow & L0 \\ \downarrow & & \downarrow & & \downarrow \\ L0 & \longleftarrow & y & \longrightarrow & L0 \end{array}$$

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

$$\begin{array}{ccccc} L0 & \longleftarrow & y & \longrightarrow & L0 \\ \uparrow & & \uparrow & & \uparrow \\ L0 & \longleftarrow & L0 & \longrightarrow & L0 \\ \downarrow & & \downarrow & & \downarrow \\ L0 & \longleftarrow & z & \longrightarrow & L0 \end{array}$$

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

$$\begin{array}{ccccc} L0 & \longleftarrow & w & \longrightarrow & L0 \\ \uparrow & & \uparrow & & \uparrow \\ L0 & \longleftarrow & L0 & \longrightarrow & L0 \\ \downarrow & & \downarrow & & \downarrow \\ L0 & \longleftarrow & z & \longrightarrow & L0 \end{array}$$

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

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

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

$$\begin{array}{ccccc} L0 & \leftarrow & w+\ell & \rightarrow & L0 \\ \uparrow & & \uparrow & & \uparrow \\ L0 & \rightarrow & x+LRk & \leftarrow & L0 \\ \downarrow & & \downarrow & & \downarrow \\ L0 & \leftarrow & y+r & \rightarrow & L0 \end{array}$$

But $w + \ell \rightsquigarrow^* y + r$ as seen in Lemma 3.5.

□

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.