

## Rewriting Structured Cospans

Daniel Cicala

This paper develops a framework for describing how composite systems can mutate according to a set of local rules. The key idea is to combine structured cospans with double pushout rewriting.

Structured cospans describe open systems and their composition — a key class of examples is systems that admit a graph or network based description, including electric circuits, social networks, and the link structure of the internet. They were recently defined by Baez and Courser. Rewriting describes rules for mutating one object into another; double pushout rewriting is a well-known framework for rewriting that is especially suited to the case of graphs.

While both structured cospans and double pushout rewriting can be done in more general settings, this paper specialises to the common setting of a topos. In this specialized setting, an L-structured cospan is a cospan  $L_a \longrightarrow x \longleftarrow L_b$  in some topos  $X$ , where  $L: X \longrightarrow A$  is a geometric morphism. The first result of the paper (Theorems 3.5 and 3.6) is that these L-structured cospans form the objects of a topos, and moreover that the construction of this topos is functorial in the choice of  $L$ .

In double pushout rewriting, a rewrite rule is a span  $l \longleftarrow k \longrightarrow r$  of monomorphisms, and a grammar is a set of rewrite rules (in a fixed topos). Further rewrite rules can be derived using a double pushout construction, and the rewrite relation comprises pairs  $(a,b)$  of objects such that there is a finite sequence of rewrite rules from  $a$  to  $b$ .

The author generalizes the notion of rewrite rule to open rewrite rule (“structured cospan rewrite rule”): a span of structured cospans such that the maps between the apexes are monomorphisms, and the maps between the feet are isomorphisms. These open rewrite rules form the 2-cells of a double category  $LStrCsp$ . In  $LStrCsp$ , horizontal composition of 2-cells corresponds to the gluing of two open rewrite rules, while vertical concatenation corresponds to an enacting rewrite rules in sequence.

After this set-up, the paper proceeds through three key theorems.

First, a structured cospan grammar is a geometric morphism  $L$  together with a set  $P$  of structured cospan rewrite rules. The construction of  $LStrCsp$  gives rise to a definition of the language double category associated with a grammar: the sub-double category of  $LStrCsp$  generated by the rewrite rules that can be derived from  $P$  using double pushout rewriting. This construction and its functoriality is Theorem 4.5.

Second, it is a result of Ehrig, Pfender, and Schneider that any graph rewrite rule is equivalent to the one in which the apex of the span is replaced with its underlying discrete graph. Theorem 5.6 provides conditions under which generalizes; in particular, it occurs for rewriting in any topos with a discrete comonad, such that for every rule  $l \longleftarrow k \longrightarrow r$  in the grammar the subobject lattice of  $k$  has all meets.

Finally, the author uses these two results to generalize a result of Gadducci and Heckel regarding characterizing the rewrite relation for a graph grammar. In particular, Theorem 6.2 states that under the conditions of Theorem 5.6, some pair  $(g, h)$  is in the rewrite relation for a grammar iff there exists a 2-cell of a certain form in a certain language double category (as defined in Theorem 4.5).

\*\*\*General comments\*\*\*

Overall the paper provides an interesting, novel perspective on a topical problem, and is well-written, with an especially thorough, accessible introduction and an enlightening running example.

That said, I'm left with a few questions.

First, what are some examples of morphisms of grammars and structured cospan grammars? While Sections 2 and 4 develop categories of grammars, I'm left unsure exactly what motivates this or how to understand the choice of morphisms. Moreover, while an analogy is built between these sections, it's not fully clear how Theorem 4.5 relates to Theorem 2.2: the relationship between the two different constructions of language categories is not discussed.

Compounding this, I'm not clear on what problem the language double category is designed to solve. The morphisms of the language category of Theorem 2.2 are in one-to-one correspondence with rewrites inside a double pushout formalism: they are a sequence of derived rules leading from the domain to the codomain. The 2-cells of the language double category of Theorem 4.5 are vertical and horizontal composites of derived rules and identity 2-cells within a certain double category; and so, subject to this composition rule, two different ways of rewriting one structured cospan to another may be represented by the same morphism. As such, it lies between a free language category in the ("fully operational") style of Theorem 2.2, and the ("fully denotational") rewriting relation. As far as I can tell, the vertical composition of 2-cells (open rewrite rules) by pullback has limited interpretation in terms of the rewriting system. Why not again use a similar free construction?

Relatedly, on page 32 the author writes "we argue that the rewrite relation is not the morally correct semantics to study in the case of structured cospans". What is? Is the author suggesting that the double category provides the correct semantics, presumably because it allows the composition of open systems/structured cospans as well? Notions of what is 'morally correct' aside, for what purpose should we be interested in the construction of Theorem 4.5? If the main point is to provide a setting for Theorem 6.2, that's certainly acceptable, but if the ambition is more general it would be nice to hear about it.

Second, and most substantially, I'm unable to follow lines 7 and 8 of the proof of Theorem 5.6 on page 36. Since we want to use the hypothesis that  $\text{Sub}(k)$  has all meets, my guess is that we're working in the poset  $\text{Sub}(k)$ . However, since all we know about  $d$  is that there's a morphism  $k \rightarrow d$ , I don't understand how we can work with  $d$  as an element of  $\text{Sub}(k)$ . Perhaps we instead

want to be working in  $\text{Sub}(\mathcal{d})$ ? In this case, I'm not sure why  $w$  is well-defined, and indeed how best to understand  $k$  and  $\bigwedge k$  as elements of  $\text{Sub}(\mathcal{d})$ , nor why  $w \bigwedge k = \bigwedge k$ .

Third, while I'm sympathetic to the idea behind the converse argument on pages 41-42 in the proof of Theorem 6.2, I would like to see more detail. To proceed by structural induction, my understanding is that we should have some recursive way to construct all squares of the form specified by the theorem. We then want to prove the property holds for the base cases, and is preserved by the recursive steps. In this proof, the author implicitly suggests that all such squares arise in essentially two ways, which seems believable, but I would like to see an explicit justification.

Finally, it would be nice to see an indication of the utility of the generalizations given in Theorems 5.6 and 6.2, if only by listing some examples of toposes with discrete comonads and whose subobject lattices have all meets, in which rewriting systems have an interesting application or interpretation. (The ZX-calculus is briefly mentioned as an example, but it would be nice to hear of another.)

Nonetheless, I thank the author for an enjoyable read, and subject to clarification of the proofs of Theorems 5.6 and 6.2, and review of the typos and comments listed below, I would be happy to recommend this paper for publication.

\*\*\*Line-by-line comments\*\*\*

Abstract, line 5: "structure cospans"  $\longrightarrow$  "structured cospans"

page 6, line -10: I think it's easier to parse at this point if the comma category is written with parenthesis around the first argument, as in: " $(A \times A) \downarrow \Delta R$ "

p.11, l.10: I think it would be helpful to quickly note that  $0$  is the chosen notation for initial object.

P.14, Definition 2.1: Is this definition novel here? What informs the definition of morphism of grammar? I see for Theorem 2.2 we need morphisms to preserve pushouts, but what is the importance of preserving pullbacks?

p.17, Theorem 2.2: Last sentence in theorem statement is missing a period. Moreover, I think a brief remark on the proof would be good, even just to acknowledge that it's quite straightforward.

p.19, first paragraph: most arrows in this paragraph appear to be reversed, drawing spans rather than cospans

p.19, Definition 3.2: What is an isomorphism of structured cospans? If the concern is that it's not important for the main narrative of the paper, then at least a reference would be nice. That said,

enough for the definition seems to later be given in Definition 3.4. It might be useful to combine both definitions by speaking about the double category of structured cospans directly, and then specialising to these two cases.

p.19, Example 3.3: “A graph is open when equipped with two subsets of nodes...” — even if it feels morally true, this seems misleadingly imprecise to me, since a structured cospan over the Graph-Set adjunction consists of a pair of functions from a set into the underlying set of a graph. This is not the same thing as a pair of subsets of the nodes.

p.21, Definition 3.4: See remarks about Definition 3.2.

p.21, 1.12 (displayed math): I think it would be more consistent with earlier notation (and less confusing) to call the feet of the cospans “ $a, b, c, \dots$ ” and the apexes “ $R_x, R_y$ ”, since the feet are in  $A$  and the apexes in  $X$ .

p.21, 1.-1: The last word “Topos” on this page should be in sans serif, rather than italics.

p.22, 1.2: I suggest writing  $F, F'$  for the adjunction between  $X$  and  $X'$ , and  $G, G'$  for the adjunction between  $A, A'$ , since this sets up the notation for Definition 3.7 nicely. I think I would have found it easier to read that way.

p.22, 1.9: “L-structured cospans”  $\longrightarrow$  “L-structured cospan”.

p.22, lines 14,15 (displayed math): I think the  $F$ s should be  $G$ s in these lines, and vice versa.

p.23, 1.4: There is an unwanted space between  $\{\}_{{}_L}$  and  $\text{StrCsp}$ .

p.23, Definition 3.7 (1.8): I think it'd be nice to briefly say why this is the right notion of morphism. (Similarly for 2-morphism — does this 2-category become useful for something?)

p.23, 1.-5: “ $G \rightarrowtail G$ ”  $\longrightarrow$  “ $G \rightarrowtail G$ ”

p.23, 1.-2: “therefore, support  $a$ ”  $\longrightarrow$  “therefore, supports  $a$ ”

p.25, last paragraph: this sentence claims that structured company grammars and their morphisms form a subcategory of  $\text{Gram}$ . Further to my comment on Definition 3.7, I think it'd be nice to briefly say somewhere why this is a subcategory (ie. why the induced morphisms between the toposes preserve pushouts and pullbacks).

p.26, 1.-4: “legs of the span”  $\longrightarrow$  “legs of the cospan”

p.26, 1.-3: this should be a cospan not a span.

p.27, 1.1,2: similar issues regarding spans/cospans.

p.28, l.-3: “Rule 7”  $\longrightarrow$  “Rule (7)” seems more in line with your convention on labelling displayed math, and also hints better to the reader to look for displayed math (7), and not some list of 7 rules somewhere.

p.30, ll.-3, -1: The apostrophes are not well placed in these two lines of displayed math, two are misplaced, one is missing, one is not necessary.

p.32, ll.13, 14: The notation  $\$ \flat P \$$  here should be replaced by  $\$ P \_ \flat \$$ .

p.35, Proposition 5.4 and proof:

Many  $\$ T \$$ s here should be sans serif.

In the first line of the proof,  $\$ T \searrow t \$$  should be  $\$ T \searrow k \$$ . In the second sentence of the proof, “hence meets” would be more precise if it specified “hence meets in  $\text{Sub}(k)$ ”; the “it” that is the subject of the sentence until this point is the slice topos.

More materially, the last sentence is not precise. Any join in  $\text{Sub}(k)$  is not a coproduct in the slice category. For example, working in  $\text{Set}$ , the join of the top element with itself in  $\text{Sub}(1)$  is not the coproduct of 1 with itself in  $\text{Set}$ . Nonetheless, if the slice topos over  $k$  has coproducts, then a join can be constructed by factorizing the map from the coproduct to  $k$ .

p.36, l.-4: I think the second “ $g \rightsquigarrow h$ ” should not have the  $*$ .

p.38, l.-4: Again, I think it would be helpful here to remark 0 is the chosen notation for the initial object.

p.38, l.-3: “decomposition we use are”  $\longrightarrow$  “decomposition we use is”/“decompositions... are”.

p.40, l.1: “ $r_1$ ”  $\longrightarrow$  “ $r_{_1}$ ”. Also, while it’s ultimately clear what this construction should be, I think this diagram could be improved. It seems a little too abbreviated to me, so the pattern the ellipses are meant to continue is not really established. I understand there are space constraints though.