Coherent invertibility in associative n-categories

Anonymous Author(s)

ABSTRACT

The theory of associative n-categories has been employed to give a combinatorial model of directed higher categories amenable to computer implementation, presented as n-dimensional string diagrams. However, the theory lacks a notion of inverse, which would require the existence of cancellation moves along with an infinite collection of higher-dimensional coherence data.

We generalise the theory to support invertibility by equipping generators with 'framing' data, allowing for a natural representation of the inverse. Using this 'framed zigzags' approach, we show that the required cancellation moves and coherence data arise uniformly via a colimit mechanism. We use tools from enriched category theory to justify correctness of our constructions, and exhibit a proof assistant which implements our theory.

CCS CONCEPTS

• **Theory of computation → Categorical semantics**; *Rewrite systems*; *Automated reasoning*; • **Mathematics of computing → Algebraic topology**.

ACM Reference Format:
Anonymous Author(s). 2024. Coherent invertibility in associative n-categories. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Coherent equivalences. In a category, a morphism $f\colon x \to y$ is invertible if it admits an inverse $f^{-1}\colon y \to x$, such that the composition in either order is the identity morphism:

$$f^{-1} \circ f = \mathrm{id}_x, \qquad f \circ f^{-1} = \mathrm{id}_y.$$

In this case, we say that f and f^{-1} are an inverse pair, witnessing the isomorphism $x \cong y$. In a higher category, this structure naturally generalises to that of an *equivalence*, where the equalities above are replaced by directed 2-cells that perform pair cancellations or introductions:

$$\begin{aligned} f^{-1} \circ f &\Rightarrow \mathrm{id}_x, & f \circ f^{-1} &\Rightarrow \mathrm{id}_y, \\ f^{-1} \circ f &\Leftarrow \mathrm{id}_x, & f \circ f^{-1} &\Leftarrow \mathrm{id}_y. \end{aligned}$$

We may require these 2-cells to themselves form equivalences as inverse pairs. This yields a coinductive definition of equivalence, which generates structure in all dimensions [18].

The resulting notion of equivalence has poor algebraic properties. To see why, consider the following 2-cell, obtained as a composite of the 2-cells described above:

$$f = \underline{\mathrm{id}_y} \circ f \Rightarrow (f \circ f^{-1}) \circ f = f \circ \underline{(f^{-1} \circ f)} \Rightarrow f \circ \mathrm{id}_x = f.$$

Here we underline the redex applying in each case, and assume for simplicity our theory is definitionally unital and associative. This 2-cell is called the *snake composite*, and has type $f \Rightarrow f$. We might expect it to be equivalent to id_f; this must certainly be the case if we desire the free ∞-category generated by an equivalence $f\colon x \to y$ to itself be equivalent to the free ∞-category generated by a single object, where parallel morphisms are always equivalent. However, under the coinductive definition of equivalence presented before, this will not be the case; we say that this notion of equivalence is *incoherent*.

For a coherent theory of equivalences, the snake composite must therefore be equivalent to the identity (via a coherent equivalence). But this is not the end of the story: there is an infinite sequence of such phenomena, known as *catastrophes* [6] from their study in manifold theory, with one qualitatively new example arising in each dimension[1]. Any computer algebra system for higher categories must have a solution to this problem in order to implement a satisfactory theory of equivalences.

String diagrams. This paper presents a solution to the problem of equivalences for the theory of string diagrams. We work with the formalism of *associative n-categories* [8], which has been developed into a proof assistant *homotopy.io* [17, 9, 22]. This proof assistant encodes string diagrams combinatorially using a simple inductive structure called a *zigzag*. This proof assistant allows the construction of composite higher morphisms in a finitely-generated higher category, using a geometrical string diagram interface. Given the difficulty of encoding coherent equivalences as discussed above, all the generators of the theory are directed.

The main technique for introducing nontrivial structure in the proof assistant is *contraction*, which uses a colimit operation to simplify some part of a diagram [17]. The simplest nontrivial example looks like Figure 1, where we collapse a diagram in which two 2-cells are composed at different heights. This contraction procedure can be applied both in the top dimension as demonstrated here, and also recursively within subdiagrams, to allow the construction of complex proofs.

In this article, we show how the contraction technology can be modified to allow the definition of invertible generators and manipulation of coherent inverses. As a result, the proof assistant *homotopy.io* gains the ability to work with these, and hence becomes a language for finitely-presented higher groupoids. This new capability will be enabled by introduction of *framing information* at the level of zigzags.

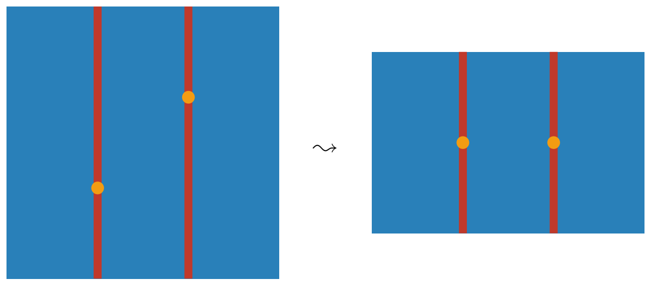
[1] The first few have enigmatic names: the snake, swallowtail, butterfly and wigwam.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference'17, July 2017, Washington, DC, USA

© 2024 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



$$
\begin{array}{ccccccccc}
x & \to & f & \leftarrow & x & \to & f & \leftarrow & x \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
x & \to & f & \leftarrow & x & \to & e & \leftarrow & x \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
x & \to & f & \leftarrow & x & \to & f & \leftarrow & x \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
x & \to & e & \leftarrow & x & \to & f & \leftarrow & x \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
x & \to & f & \leftarrow & x & \to & f & \leftarrow & x
\end{array}
\rightsquigarrow
\begin{array}{ccccccccc}
x & \to & f & \leftarrow & x & \to & f & \leftarrow & x \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
x & \to & e & \leftarrow & x & \to & e & \leftarrow & x \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
x & \to & f & \leftarrow & x & \to & f & \leftarrow & x
\end{array}
$$

Figure 1: Contraction example

To understand this, we consider a 1-cell $e \colon x \longrightarrow x$. In the zigzag formalism, this is encoded combinatorially as the following cospan:

$$x \longrightarrow e \longleftarrow x.$$

Suppose we would like e to be invertible. Then how can we encode e^{-1}? One approach may be to introduce 'e^{-1}' as a formal token, and allow the following zigzag to represent the inverse 1-cell:

$$x \longrightarrow e^{-1} \longleftarrow x.$$

However, this is unsatisfactory: we know that an infinite sequence of higher morphisms will be required to ensure the inverse is coherent — including the snake and its associated higher catastrophes — but their structure is chaotic, and we do not know of a good way to introduce a corresponding syntax of higher tokens.

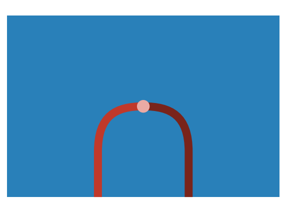
Our solution is to introduce for each arrow $x \longrightarrow e$ a *framing label*, as a decoration on the arrow. This allows us to distinguish e and e^{-1} as follows:

$$x \xrightarrow{1} e \xleftarrow{2} x \qquad\qquad x \xrightarrow{2} e \xleftarrow{1} x$$

The labels 1 and 2 are formal tokens to distinguish the framings; the choice of label does not have any special meaning. We call this extended structure a *framed zigzag*. This has the attractive property that the framed zigzag representations of e and e^{-1} are mirror-images.

Using this idea, we consider applying the contraction principle to the composite $e \circ e^{-1}$. Contraction is computed by colimit, which is easily computed over our base category with objects $\{x, e\}$ and morphisms $x \xrightarrow{1} e$ and $x \xrightarrow{2} e$. We obtain the following result:

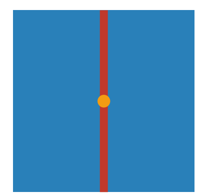
$$
\begin{tikzcd}
 & & x \arrow[d, "1"] \arrow[dll, equal] \arrow[drr, equal] & & \\
x \arrow[rr, "1"] \arrow[d, equal] & & e & & x \arrow[ll, "1"'] \arrow[d, equal] \\
x \arrow[r, "1"'] \arrow[urr, "1"] & e \arrow[ur, equal] & y \arrow[l, "2"] \arrow[r, "2"'] \arrow[u, "2"'] & e \arrow[ul, equal] & x \arrow[l, "1"] \arrow[ull, "1"']
\end{tikzcd}
$$



We observe that the contraction procedure performs the cancellation of e with e^{-1} as intended. Above, we append two triangles which close off the geometry, yielding an upper boundary labelled by x. Overall this is the correct geometrical representation of the cancellation move $e \circ e^{-1} \Rightarrow \mathrm{id}_x$, depicted in string diagrams on the right.

Posetal enrichment. We extend the same principle to generators of higher dimension. For example, a 2-cell $f \colon e \Rightarrow e$ would be represented as follows, where we give both the framed zigzag representation as well as the geometrical rendering produced by the proof assistant:

$$
\begin{tikzcd}
x \arrow[r, "1"] \arrow[d] \arrow[dr, phantom, "6"] & f \arrow[d, "7"] \arrow[dr, phantom, "8"] & x \arrow[l, "2"'] \arrow[d] \\
x \arrow[r, "9"] \arrow[dr, phantom, "3"] & e \arrow[dr, phantom, "5"] & x \arrow[l, "10"'] \\
x \arrow[r, "1"'] \arrow[u] & f \arrow[u, "4"'] & x \arrow[l, "2"] \arrow[u]
\end{tikzcd}
$$



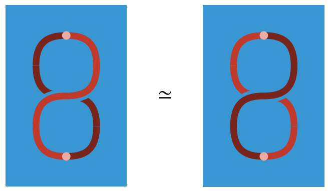
Here we see a further new aspect of our construction: where a triangle appears in the neighbourhood of a generator, we give it a 2-cell filler. This 2-cell structure will be posetal, and so we can handle it with the technology of **Pos**-enriched 1-categories. For any 2-cell generator, its neighbourhood structure is therefore encoded by a particular choice of finite **Pos**-enriched category.

As the dimension goes up, this **Pos**-enriched 1-categorical framework remains constant. Therefore, even for an n-dimensional generator X where $n > 2$, its neighbourhood is completely defined by a choice of **Pos**-category, and the same will be true for all composite geometries. In this way we are able to encode higher categorical diagrams of arbitrary dimension, with a formal foundation that stays at the level of **Pos**-categories. Geometrically, this suggests our geometries are encoded by 2-coskeletal simplicial sets. Precedent for this comes from the work of Nanda [15], who uses **Pos**-categories as a rich setting for developing discrete Morse theory in a categorical setting. As a consequence of this **Pos**-enrichment, contraction must now be handled with oplax conical colimits.

Implementation. These ideas have been implemented, and are available in a pre-release version of the proof assistant *homotopy.io*, available at the location https://beta.homotopy.io. Here, a generator can be given invertible structure via its settings dialog, accessed by clicking the cog icon.

To demonstrate the application of invertible generators, we illustrate in Appendix A a proof of the following result, along with a link to a video representing the proof as a smooth movie of 3-dimensional structures [12].

Theorem 1.1. *In the higher category generated by an object x and a 2-cell $f : \mathrm{id}_x \Rightarrow \mathrm{id}_x$, the following 'Figure-8' string diagrams* *represent equivalent 3-cells:*



The Figure-8 diagrams are themselves composites of the invertible structure, along with a braiding move. This result plays an important role in the homotopy groups of spheres, as follows. The free higher groupoid on an object x and a 2-cell $f\colon \mathrm{id}_x \to \mathrm{id}_x$ generates the homotopy type of the 2-sphere, and the homotopy classes of 3-cells in any given hom-set are in bijection with the elements of the homotopy group $\pi_3(S^2)$. Naively it appears there are four such Figure-8 diagrams, coming in two inverse pairs, and therefore we might expect 2 independent homotopy generators in this class, yielding $\pi_3(S^2) \simeq \mathbb{Z}_2$. However, due to the non-obvious equivalence established in the theorem, in fact these generators are homotopy equivalent, and hence $\pi_3(S^2) \simeq \mathbb{Z}$.

It is in this sense that our tool, homotopy.io, is described as a *proof assistant*, facilitating the formalisation and study of these theories.

1.1 Related work

This work is part of a larger programme to develop a proof assistant for higher category theory in the model of associative n-categories [8], homotopy.io.In particular, it builds on the zigzag construction and contraction operation described in Reutter and Vicary [17]. Other important constructions in the tool are given by Heidemann et al. [9], Sarti and Vicary [20] and Tataru and Vicary [22].

The predecessor tool *Globular* [1] directly encoded the axioms of a quasistrict globular 4-category, and as such is fundamentally limited to dimension 4. It included an *incoherent* notion of inverse, with the coinductive structure of cancellations, but without the snake equation and higher coherence moves.

Existing type-theoretic models of higher groupoids [3, 4], including homotopy type theory [23], handle invertible structures in a coherent way. The proof assistant CaTT [2] can handle both directed and coherent undirected inverses, although that point is not made very clearly in the literature. The novelty of our approach is to use new techniques that extend support of invertible generators to a geometrical string-diagram setting.

1.2 Outline

The outline of our contribution is as follows. Section 2 introduces the necessary background on enriched categories, and in particular **Pos**-categories in detail, which are categories enriched in the category of posets and monotone maps. Section 3 introduces the notion of *framed zigzags*, which are a generalisation of the zigzags of Reutter and Vicary [17] modelling associative n-categories to an appropriately enriched setting. Section 4 shows that framed zigzags give rise to a notion of coherent invertibility in associative n-categories.

1.3 Notation

All (enriched) categories will be strict and small, which means that their objects form a set and can be compared for equality. $f[X]$ denotes the application of a function f to a subset X of its domain pointwise. Δ_0 denotes the category of (possibly empty) finite linear orders and monotone maps, also known as the augmented simplex category. Δ_x denotes the constant functor at x.

2 ENRICHED CATEGORIES

We assume familiarity with basic enriched category theory, so we will elide some of the details necessary for a fully-rigorous presentation. The reader is referred to Kelly [14] and Riehl [19, §3] for a detailed introduction to the general theory. Familiarity with monoidal category theory or higher category theory will also be helpful, but we will not cover it here. Nonetheless, we begin this section with a review of the main concepts used in the theory of categories enriched in a symmetric monoidal category $\mathcal{V}$ ($\mathcal{V}$-categories). For our purposes, $\mathcal{V}$ will always be cartesian closed, and complete and cocomplete, which is a relatively simple fragment of the general theory of enriched categories.

Definition 2.1 ($\mathcal{V}$-category). A $\mathcal{V}$-category C is a category enriched over $\mathcal{V}$; it has a class of objects $x \in C$, and, for any pair of objects x and y, an object $C(x, y)$ of the category $\mathcal{V}$. For every object x, there is a distinguished $\mathcal{V}$-morphism $\mathrm{id}_x\colon 1 \to C(x, x)$ called the *identity on* x. For any triple of objects x, y, z, there is a $\mathcal{V}$-morphism $\circ_{x,y,z}\colon C(y, z) \times C(x, y) \to C(x, z)$ called *composition*. Composition is associative and unital with respect to these identities, in the sense that the coherence diagrams expressing these must commute.

Definition 2.2 ($\mathcal{V}$-functor). A $\mathcal{V}$-functor $F\colon C \to \mathcal{D}$ between $\mathcal{V}$-categories is given by a mapping of objects $x \in C$ to $Fx \in \mathcal{D}$, together with $\mathcal{V}$-morphisms $F_{x,y}\colon C(x, y) \to \mathcal{D}(Fx, Fy)$ which preserve identities and compositions in the sense of Definition 2.1.

*Example 2.3 ($\mathcal{V}$ = **Set**).* Enriching in the category of sets and functions, $\mathcal{V}$-categories are ordinary (locally small) categories.

*Example 2.4 ($\mathcal{V}$ = **Vect**).* Enriching in the category of vector spaces and linear maps, $\mathcal{V}$-categories are linear categories.

*Example 2.5 ($\mathcal{V}$ = **Ab**).* Enriching in the category of abelian groups and group homomorphisms, $\mathcal{V}$-categories are preadditive categories. Preadditive categories generalise rings in the same way that ordinary categories generalise monoids: every one-object preadditive category is precisely the data of a ring (with identities).

*Example 2.6 ($\mathcal{V}$ = **Bool**).* Enriching in the category determined by the poset $\bot \to \top$, $\mathcal{V}$-categories are preorders.

*Example 2.7 ($\mathcal{V}$ = **Cat**).* Enriching in the category of (small) categories and functors, $\mathcal{V}$-categories are 2-categories. These are the natural setting for '2-dimensional category theory', generalising ordinary category theory; in addition to having objects and morphisms, there are also 2-cells between parallel morphisms. A modern reference is Johnson and Yau [13].

Lemma 2.8. *There is a category $\mathcal{V}$-**Cat** whose objects are $\mathcal{V}$-categories and morphisms are $\mathcal{V}$-functors.*

Lemma 2.9 (Change of enriching base [19, Lemma 3.4.3]). *Given a lax monoidal functor $F\colon \mathcal{V} \to \mathcal{W}$ and a $\mathcal{V}$-category C, there is a $\mathcal{W}$-category F_*C with the same objects as C, with Hom objects given by*

$$F_*C(x, y) \coloneqq FC(x, y).$$

Definition 2.10 (Underlying category). The underlying category of a $\mathcal{V}$-category C is the category C_0 with the same objects as C, and Hom sets given by

$$C_0(x, y) \coloneqq \mathcal{V}(1, C(x, y)).$$

This is equivalent to base change along the functor $\mathcal{V}(1, -)\colon \mathcal{V} \to$ **Set**.

2.1 Pos-categories

Our most important instance is $\mathcal{V} = \mathbf{Pos}$, so we will spend some time explicating what constitutes a **Pos**-category[2].

*Definition 2.11 (**Pos**-category).* A **Pos**-category is a category enriched over **Pos**, the category of posets and monotone maps.

Explicitly, this means that if C is a **Pos**-category, then C has the structure of a category, and additionally for each pair of objects $x, y \in C$, the Hom $C(x, y)$ is partially ordered, and composition must respect this partial ordering: if $f \le f' \in C(x, y)$, and $g \le g' \in C(y, z)$, then in $C(x, z)$, $g \circ f \le g' \circ f'$, and similarly for precomposition also.

One can understand this concept by analogy with 2-categories: posets are trivial kinds of categories (one in which every Hom set is thin), so enrichment in these yields a simplified notion of higher category, sitting somewhere between the theory of categories and the theory of 2-categories — a **Pos**-category is roughly a 2-category where all the 2-cell structure is thin[3], and the specification of monotonicity conditions with respect to Hom posets is equivalently given by 2-cell filler structure and its preservation. Because of this, we can apply known results from 2-category theory to **Pos**-categories in a simplified form. **Pos**-categories are a middle ground allowing for the development of some ideas which are 2-categorical in nature, but in a simplified setting which mostly has the flavour of ordinary category theory with some additional monotonicity conditions in some places.

Another intuition for **Pos**-categories is that they are informally the minimum amount of structure on top of a category necessary to have a sensible notion of *op/laxly commutative diagram*: a diagram in which some of the 2D faces do not represent equality between morphisms, but rather an ordering relation. In particular, this allows for the richer notion of *oplax* conical colimits, which we will make use of heavily in the following sections.

It is for these reasons that we use the symbol $\Rightarrow$ for the partial order relation on Hom posets in a **Pos**-category from here onwards, and describe the witnesses to an ordering $f \Rightarrow g$ in a Hom poset as a 2-cell filler.

*Definition 2.12 (**Pos**-functor).* A **Pos**-functor is a mapping between **Pos**-categories; for $F\colon C \to \mathcal{D}$, each object $x \in C$ is mapped to $Fx \in \mathcal{D}$, and for each $x, y \in C$ there is a monotone map $C(x, y) \to \mathcal{D}(Fx, Fy)$ which preserves identity and composition. That is, $F(\mathrm{id}_x) = \mathrm{id}_{Fx}$, and $F(g \circ f) = F(g) \circ F(f)$.

Lemma 2.13. *There is a category* **Pos-Cat** *whose objects are* **Pos***-categories and morphisms are* **Pos***-functors.*

Lemma 2.14. *There is an adjunction of categories between* **Cat** *and* **Pos-Cat**.

Proof. Every **Pos**-category C has an underlying category C_0, which has the same objects, and each Hom set $C_0(x, y)$ given by $\mathbf{Pos}\,(1, C(x, y))$, where 1 is the trivial one-element poset. The monotone maps from the singleton to $C(x, y)$ are in bijection with the elements of $C(x, y)$, so we can think of the category C_0 as merely C and forgetting that each Hom has a partial order.

Moreover, the functor $\mathbf{Pos}(1, -)\colon\ \mathbf{Pos} \to \mathbf{Set}$ admits a left adjoint, which is the functor which equips a set with the trivial partial order, where all elements are pairwise incomparable. This means that every category can be considered as a **Pos**-category, by equipping each Hom set with the trivial partial order, and every functor can be considered as a **Pos**-functor between **Pos**-categories obtained this way which is vacuously monotone (the action on 2-cell fillers is trivial). □

Corollary 2.15 (Δ **Pos**-category). *There is a* **Pos***-category* Δ *of finite linear orders and monotone maps, with trivial 2-cell fillers given by* $f \Rightarrow g$ *exactly when* $f = g$.

We have an important special case of **Pos**-categories which we will make use of in the algorithmic sections of the paper.

*Definition 2.16 (id-max **Pos**-category).* A **Pos**-category C is *id-max* when for each $c \in C$, id_c is the top element of the poset $C(c, c)$.

2.2 Completions of posets

Definition 2.17 (Meet). A poset P has meets if for every subset $Q \subseteq P$, there is a greatest lower bound of Q, $\bigwedge Q \in P$. Explicitly, $\bigwedge Q$ has the following properties:

- $\forall q \in Q.\ \bigwedge Q \le q$,
- for $p \in P$, if $\forall q \in Q. p \le q$, then $p \le \bigwedge Q$.

In this case we say that P is a *meet-semilattice*.

Dually, a poset P has joins if for every subset $Q \subseteq P$, there is a least upper bound of Q, $\bigvee Q \in P$.

*Definition 2.18 ($\bigwedge$-**Lat**).* The category $\bigwedge$-**Lat** is the category of meet-semilattices and meet-semilattice homomorphisms. That is, its objects are posets which have all meets, and morphisms are monotone maps preserving them.

Lemma 2.19. $\bigwedge$-**Lat** *is a cartesian closed subcategory of* **Pos**.

Proof. The set $1 \coloneqq \{*\}$ with the trivial ordering is the terminal object of $\bigwedge$-**Lat**: it is a meet-semilattice, with every meet given by $*$, and for any other poset P the unique map $P \to \{*\}$ is trivially monotone and meet-preserving. $P \times Q$ is a meet-semilattice if and only if P and Q are, and this gives the categorical product in $\bigwedge$-**Lat**. □

[2] Some authors call these *2-posets*; our choice in terminology emphasises our focus on the theory of enriched categories.

[3] We call these *2-cell fillers*.

Definition 2.20 (Free join-completion). The free join-completion of a poset P, $F(P)$, is the poset of downwards-closed subsets of P, ordered by inclusion. The join of a set of downwards-closed subsets of P, X, is given by $\bigvee X \coloneqq \bigcup X$.

Lemma 2.21. *$F(P)$ also has all meets.*

Proof. $F(P)$ is a join-semilattice. The **Bool**-enriched adjoint functor theorem says that every join-semilattice is also a meet-semilattice. Meets in $F(P)$ are given by intersections of downwards-closed subsets. □

Lemma 2.22. *There is a monotone map $\downarrow(-)\colon P \to F(P)$, sending each element $p \in P$ to the* principal downwards-closed subset $\downarrow p \coloneqq \{q \in P \mid q \leq p\}$*. Moreover, this map preserves and reflects meets in P.*

Proof. This is the **Bool**-enriched version of the fact that the Yoneda embedding sends a category to its free cocompletion, which is presheaf topos, in a way which preserves and reflects limits. Posets are skeletal **Bool**-categories, monotone maps are **Bool**-functors, and meets and joins are **Bool**-enriched limits and colimits respectively. $F(P)$ is the **Bool**-enriched presheaf topos associated to the **Bool**-category P, and it has all meets and joins analogously to how presheaf topoi have all limits and colimits. □

Lemma 2.23. *F extends to a functor $F\colon$ **Pos** $\to \bigwedge$-**Lat** which sends monotone maps to meet-semilattice homomorphisms.*

Proof. The action of taking **Bool**-presheaves is functorial: each monotone map $f\colon P \to Q$ induces a monotone map $f^*\colon F(Q) \to F(P)$ by precomposition. In this instance, we have an adjoint triple $f_! \dashv f^* \dashv f_*$ in the **Bool**-enriched sense. f_*, as a right adjoint, preserves **Bool**-enriched limits (i.e. meets), and is given explicitly as

$$f_*(X) \coloneqq \bigcap \{\downarrow f(x) : x \in X\},$$

for a downwards-closed subset $X \subseteq P$. So define $Ff \coloneqq f_*$. □

Remark 1. This is in contrast to the usual free-forgetful adjunction between posets and join-semilattices, which would send f to $f_!$ given by union instead of intersection.

Proposition 2.24. *F is a lax monoidal functor.*

Proof. We need

(1) a meet-preserving monotone map $\{*\} \to F(\{*\})$, which is given by $\downarrow *$;
(2) a meet-preserving monotone map $F(P) \times F(Q) \to F(P \times Q)$, which is given by

$$(X \subseteq P, Y \subseteq Q) \mapsto X \times Y.$$

The necessary coherence conditions are easy to check. □

Corollary 2.25. *Every **Pos**-category C gives rise to a $\bigwedge$-**Lat**-category $F_*(C)$ in which every Hom poset has all meets, via base change along F.*

Lemma 2.22 says that $F_*(C)$ is like a 'completion' of C which has formally added all meets to each Hom poset, preserving and reflecting any that already existed. The forgetful functor $\bigwedge$-**Lat** $\to$ **Pos** also induces a base change, which allows us to treat any $\bigwedge$-**Lat**-category as a **Pos**-category. This will be useful later to define an enriched category of *framed zigzags*.

Corollary 2.26. *If the **Pos**-category C is id-max, then so is the $\bigwedge$-**Lat**-category F_*C.*

Proof. The meet of an empty set is the top element, and F_* preserves meets in Hom posets. □

2.3 Oplax conical colimits

We now make use of this higher structure to define oplax conical colimits, which are a generalisation of colimits.

Throughout this section, let $F, G\colon C \rightrightarrows \mathcal{D}$ be a **Pos**-functors.

Definition 2.27 (Lax natural transformation). A *lax natural transformation* $\alpha\colon F \Rightarrow G$ is

- a family of $\mathcal{D}$-morphisms $\alpha_x\colon Fx \to Gx$ for each $x \in C$,
- for each morphism $f\colon x \to y$ in C, a 2-cell filler $Gf \circ \alpha_x \Rightarrow \alpha_y \circ Ff$ in $\mathcal{D}$:

$$\begin{tikzcd} Fx \arrow[r, "Ff"] \arrow[d, "\alpha_x"'] \arrow[dr, phantom, "\Longrightarrow"] & Fy \arrow[d, "\alpha_y"] \\ Gx \arrow[r, "Gf"'] & Gy \end{tikzcd}$$

Definition 2.28 (Oplax cocone). An oplax cocone over some $d \in \mathcal{D}$ is a lax natural transformation $\alpha\colon F \Rightarrow \Delta_d$. That is, a family of $\mathcal{D}$-morphisms $\alpha_x\colon Fx \to d$ such that for each morphism $f\colon x \to y$ in C, there is a 2-cell filler $\alpha_x \Rightarrow \alpha_y \circ Ff$:

$$\begin{tikzcd} Fx \arrow[rr, "Ff"] \arrow[dr, "\alpha_x"'] & \arrow[d, phantom, "\Rightarrow"] & Fy \arrow[dl, "\alpha_y"] \\ & d & \end{tikzcd}$$

Oplax cocones over a diagram F with a fixed tip d are partially-ordered pointwise, and form a poset **OplaxCocone**(F, d).

Definition 2.29 (Oplax conical colimit). The oplax conical colimit of F, $(d, \iota\colon F \Rightarrow \Delta_d)$, is a universal oplax cocone: for any other oplax cocone $\alpha\colon F \Rightarrow \Delta_{d'}$, there is a unique morphism $u\colon d \to d'$ such that the following commutes in $\mathcal{D}$:

$$\begin{tikzcd} & d' \\ Fx \arrow[ur, "\alpha_x"] \arrow[r, "\iota_x"'] & d \arrow[u, "u"'] \end{tikzcd}$$

Also, for any other oplax cocone with the same tip, $\alpha'\colon F \Rightarrow \Delta_{d'}$, which factors through d via $u'\colon d' \to d$, as above, such that for all x in C, $\alpha'_x \Rightarrow \alpha_x$, it must be the case that $u' \Rightarrow u$ and vice versa. In other words, the universal property of the oplax conical colimit is equivalently expressed as a natural order isomorphism of posets:

$$\mathcal{D}(\text{oplax colim}\, F, d') \cong \textbf{OplaxCocone}(F, d').$$

We denote the oplax conical colimit d of F by oplax colim F.

Remark 2. In the case that all oplax cocones are cocones, for example when the **Pos**-category $\mathcal{D}$ admits only trivial 2-cell fillers, oplax conical colimit coincides with conical colimit.

Definition 2.30 (Local oplax cocone). A *local oplax cocone* for F is an oplax cocone α with tip Fc for some $c \in C$, where each component α_x is given by Ff for some C-morphism f.

These form a subposet

$$\mathbf{OplaxCocone}_F(F, Fc) \subseteq \mathbf{OplaxCocone}(F, Fc).$$

This notion is helpful to resolve some of the pathologies that oplax conical colimits admit in comparison to colimits in ordinary category theory (see Appendix B).

Proposition 2.31. *Let (Fx, λ) be a local oplax cocone of F, and additionally suppose that*

(1) *$\mathcal{D}$ is id-max;*
(2) *for $c \in C$, $\mathcal{D}(Fx, Fc) \neq \emptyset \iff Fc = Fx$.*

Then (Fx, λ) is the unique local oplax cocone, and also it is maximal in $\mathbf{OplaxCocone}(F, Fx)$.

Proof. Let (Fx, α) be an oplax cocone over F. Because $\lambda_c = Ff$ for some C-morphism $f : c \to x$, by virtue of α being an oplax cocone, obtain a 2-cell filler $\alpha_c \Rightarrow \alpha_x \circ \lambda_c$. As $\mathcal{D}$ is *id*-max, $\alpha_x \Rightarrow \mathrm{id}_{Fx}$, so $\alpha_c \Rightarrow \lambda_c$. Hence, λ is maximal.

Any local oplax cocone over F must have tip Fx, or else there is no component for x. Let (Fx, λ') be another local oplax cocone. Any local oplax cocone is an oplax cocone, so deduce that $\lambda' \leq \lambda$. But the argument works in reverse because λ' is a local oplax cocone, so $\lambda \leq \lambda'$, and hence $\lambda = \lambda'$. ◻

2.4 Computing colimits by collapse

In ordinary category theory, when calculating the colimit of some functor, there is a case when the colimit is easily computable from just the data of the diagram.

Lemma 2.32. *Given a diagram $F : J \to C$ where J has a terminal object x, then F has a colimit given by Fx, with legs of the colimit arising as images of the unique J-morphisms $j \to x$ for each $j \in J$.*

The abstract justification for this is that the functor $x : \mathbf{1} \to J$ which chooses x is *final*, as below, and moreover can be obtained structurally by categorical machinery.

Definition 2.33 (Final functor). A functor $G : I \to J$ is final if for all functors $F : J \to C$, the colimit of FG exists whenever the colimit of F does, and the canonical C-morphism $\operatorname{colim} FG \to \operatorname{colim} F$ is an isomorphism.

Example 2.34. The inclusion

$$\bullet \leftarrow \bullet \rightarrow \bullet \qquad \hookrightarrow \qquad \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet$$

is a final functor. This symbolises that when a pushout diagram has 'extra legs', the overall colimit is unaffected.

Final functors are an important tool for computing colimits, representing when a colimit computation over some diagram can be replaced by a (usually simpler) colimit computation over a different diagram by restriction along that functor, and in this section we develop an analogue for oplax conical colimits in a **Pos**-category.

For ordinary categories, it is clear that whenever some diagram contains an identity morphism $x = x$, then the colimit computation can be simplified by replacing that diagram with a smaller one containing only one x. This is not true in the case of oplax conical colimits in **Pos**-categories, as the following example shows.

Example 2.35. Let C be the **Pos**-category determined by $C(x, y) := \{f \Rightarrow h \Leftarrow g\}$, and consider two diagrams $G : I \to C$ and $F : J \to C$, given below:

$$\begin{tikzcd}
a \arrow[d] \arrow[dr] & & & x \arrow[d, equal] \arrow[dr, "f"] & \\
b \arrow[r] \arrow[ur, phantom, "\Swarrow"] \arrow[dr, phantom, "\Nwarrow"] & c & \overset{F}{\longmapsto} & x \arrow[r, "h"] \arrow[ur, phantom, "\Swarrow"] \arrow[dr, phantom, "\Nwarrow"] & y \\
d \arrow[u] \arrow[ur] & & & x \arrow[u, equal] \arrow[ur, "g"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
u \arrow[r, bend left=50, ""{name=A}] \arrow[r, ""{name=B}] \arrow[r, bend right=50, ""{name=C}] & v & \overset{G}{\longmapsto} & x \arrow[r, bend left=50, "f"] \arrow[r, "h"] \arrow[r, bend right=50, "g"'] & y.
\end{tikzcd}$$

It may appear that F and G are essentially the same diagram, given the identity morphisms in the image of F, and one might expect them to have equivalent oplax conical colimits. However, while G admits an oplax conical colimit (y, ι), with $\iota_a = f$, $\iota_b = h$, $\iota_c = \mathrm{id}_y$, and $\iota_d = g$, the diagram F does not even admit any oplax cocones. Such an oplax cocone would necessarily consist of a choice of a single morphism below all others in $C(x, y)$, which does not exist.

The issue is that the presence of non-trivial 2-cell fillers obstruct the ability to remove identity morphisms in the image of G.

Definition 2.36 (Collapse). Given some diagram of **Pos**-categories, $D : J \to C$, define the sub-**Pos**-category $Q_D \hookrightarrow J \times J$ to have

objects pairs (j_0, j_1) such that

$$Dj_0 = Dj_1, \tag{1}$$
$$\textstyle\bigcup_{j \in J} D_{j,j_0}[J(j, j_0)] = \bigcup_{j \in J} D_{j,j_1}[J(j, j_1)], \tag{2}$$
$$\textstyle\bigcup_{j \in J} D_{j_0,j}[J(j_0, j)] = \bigcup_{j \in J} D_{j_1,j}[J(j_1, j)]; \tag{3}$$

morphisms $f_0 \times f_1 \to (j_0, j_1) \to (j_0', j_1')$ such that $Df_0 = Df_1$;
2-cell fillers $\alpha : (f_0 \times f_1) \Rightarrow (g_0 \times g_1)$ such that $D\alpha$ is trivial.

The *collapse of J with respect to D* is the following **Pos**-coequaliser:

$$\nabla(J) := \operatorname{coeq}(Q_D \hookrightarrow J \times J \underset{\pi_2}{\overset{\pi_1}{\rightrightarrows}} J);$$

Its universal property ensures that there is a unique **Pos**-functor, the *collapse of D*, $\nabla(D)$:

$$\begin{tikzcd}
Q_D \arrow[r, hook] & J \times J \arrow[r, shift left, "\pi_1"] \arrow[r, shift right, "\pi_0"'] & J \arrow[r, "q"] \arrow[dr, "D"'] & \nabla(J) \arrow[d, dashed, "\nabla(D)"] \\
& & & C.
\end{tikzcd}$$

The indexing **Pos**-category $\nabla(J)$ is a quotient of J under D, identifying indistinguishable objects along identities in the image of D which are not in the neighbourhood of any non-trivial 2-cell fillers; Equations (2) and (3) say that j_0 and j_1 are *indistinguishable* if the sets of morphisms with Dj_0 and Dj_1 as the co/domain are the same. In other words, no morphism in the image of D is able to distinguish between them by pre/postcomposition.

Proposition 2.37. *Suppose that C is id-max. D admits an oplax conical colimit if and only if $\nabla(D)$ does.*

Proof. Suppose that $\nabla(D)$ admits an oplax conical colimit; it is clear to see that this induces an oplax conical colimit for D by duplicating legs.

Conversely, suppose that D admits an oplax conical colimit (c, ι). We will show that C being *id*-max ensures that every leg of ι at x and y, for $f : x \to y$ in J identified in $\nabla(J)$, is equal. Firstly,

$Dx = Dy$ and $Df = \mathrm{id}_{Dx}$, so ι_x and ι_y have the same type; as Dx and Dy are indistinguishable, we can obtain a valid oplax cocone

$$\iota_j^+ := \begin{cases} \iota_y & \text{if } j = x, \\ \iota_j & \text{otherwise;} \end{cases}$$

as ι is an oplax conical colimit, ι^+ factors through ι via some morphism u^+:

$$\begin{tikzcd} & c \\ Dx \arrow[ur, "\iota_x^+ = \iota_y"] \arrow[r, "\iota_x"'] & c. \arrow[u, "u^+"'] \end{tikzcd}$$

Because ι uniquely factors through itself via id_c, and $\iota \leq \iota^+$ pointwise, it must be the case that $\mathrm{id}_c \Rightarrow u^+$. However, C is id-max, so deduce $u^+ = \mathrm{id}_c$ and therefore $\iota_x = \iota_y$. □

3 FRAMED ZIGZAG CONSTRUCTION

The iterated zigzag category $\mathrm{Zig}_0^n(C)$ represents the combinatorial space of n-dimensional string diagrams which can be built from an algebraic signature encoded by a category C. In this way, objects of this category represent n-cells of the free globular n-category over C. These are referred to as *typed* zigzag categories, with respect to C, as opposed to untyped zigzag categories where $C = \mathbf{1}$. As we are interested in making elements of a signature invertible, our discussion is implicitly focused on typed zigzags.

In previous developments, C was a trivial kind of category, namely a poset of generators ordered by dimension inclusions[4] [17, Definition 16], or even just the poset $\mathbb{N}$ of natural numbers [9]. Our key innovation is to extend C into **Pos**-category, which is not thin, whereby its morphisms encode framings, capturing a much richer notion of typing.

3.1 Framed zigzags and framed zigzag maps

We take the standard zigzag construction [17, §2] and extend it to **Pos**-categories to define *framed zigzags*.

Definition 3.1 (Framed zigzag **Pos***-category).* Let C be a **Pos**-category such that every Hom poset is locally finitely meet-complete, with these meets preserved by composition.

The **Pos**-category $\mathrm{Zig}\,(C)$ is given by the following data:

objects *framed zigzags* — a C-labelled iterated cospan, i.e. objects and morphisms of C of the form $r_0 \to s_0 \leftarrow r_1 \to \ldots \leftarrow r_n$ for $n \geq 0$ and $r_i, s_i \in C$ — the tips of the cospans (s_i) are called *singular levels*, and the feet (r_i) are called *regular levels*;

morphisms *framed zigzag maps* — a C-labelled monotone map between singular levels, combined with a C-labelled dual monotone map between regular levels[5], such that there exist 2-cell fillers which make the resulting diagram valid in C;[6]

2-cell fillers given pointwise by the 2-cell fillers of C; i.e. for two zigzag maps $f, g\colon Z \rightrightarrows Z'$, $f \Rightarrow g$ exactly when the underlying monotone maps of f and g are equal and each C-morphism component of f is below its corresponding C-morphism labelling in g.

In preparation for the description of composition of framed zigzag maps, note that there are two observations from the standard theory of zigzags which should continue to hold in our extension.

(1) Every zigzag category $\mathrm{Zig}_0\,(C)$ is canonically equipped with a projection $\pi\colon \mathrm{Zig}_0\,(C) \to \Delta_0$[7], so $\mathrm{Zig}_0\,(C)$ can naturally be considered an object of $\mathbf{Cat}/\Delta_0$.

(2) There is an adjunction of Hom sets $\mathbf{Cat}/\Delta_0(J, \mathrm{Zig}_0\,(C)) \cong \mathbf{Cat}(E(J), C)$ — every diagram in a zigzag category $\mathrm{Zig}_0\,(C)$ indexed over J is equivalently determined by a diagram in C indexed over a larger category $E(J)$ [17, Definition 23]: this larger diagram is called the *explosion* of the smaller one, and essentially consists of replacing every object in $\mathrm{Zig}_0\,(C)$ (a zigzag) with its corresponding C-labelled iterated cospan, and each zigzag map with its underlying sequence of C-labelled maps in components. An example of this, ignoring 2-cell fillers, is illustrated in Figure 2.

This suggests that composition should respect explosion of diagrams, and that composition in $\mathrm{Zig}\,(C)$ should ultimately be given in terms of composition in C.

Definition 3.2 (Composition of framed zigzag maps). For two composable framed zigzag maps in $\mathrm{Zig}\,(C)$, $p\colon Z \to Z'$, $q\colon Z' \to Z''$, their composite framed zigzag map $q \circ p\colon Z \to Z''$ is given by the following procedure:

(1) $q \circ p$ determines a planar diagram in C by explosion, where the vertical morphisms below (resp. above) are the C-morphism components of p (resp. q). An example of this is given in Figure 2.

(2) The underlying singular monotone maps of p and q determine by composition a monotone map which shall form the underlying singular monotone map of $q \circ p$; this determines the shape of $q \circ p$ entirely.

(3) To equip to this a labelling of C-morphisms, for a regular/singular level l of Z to a regular/singular level l'', take the meet of all morphisms $l \to l''$ in this planar diagram which exist because C is locally finitely meet-complete and the 2-cell filler structure of framed zigzag maps ensures connectedness. Case analysis on l and l'' yields three possibilities:
   (a) regular-to-regular components, from a regular level of Z to a regular level of Z'';
   (b) singular-to-singular components, from a singular level of Z to a singular level of Z'';
   (c) regular-to-singular components, from a regular level of Z to a singular level of Z''.

   Only in the last case is the meet not trivially determined, due to lacking a minimal path arising from shape alone.

Example 3.3. Figure 2 shows an example exploded diagram in C on the right, as in step 1. Step 2 composes the underlying monotone maps of p and q, obtaining the shape of Figure 3. Finally, in step 3,

---

[4] This easily refines to boundary inclusion.

[5] Given by the equivalence $\Delta_0 \cong \Delta_=^{op}$ [24]. Informally, when two zigzags are drawn in a planar fashion with a monotone map between singular levels, this induces a unique way to draw arrows monotonically in the opposite direction between regular levels and vice versa.

[6] These 2-cell fillers induce 'composite' regular-to-singular C-morphism components which only commute *oplaxly*. The mnemonic is that 2-cell fillers point from the short path to the long path.

[7] In the case where C is cocomplete, π is an opfibration [17, Proposition 36].

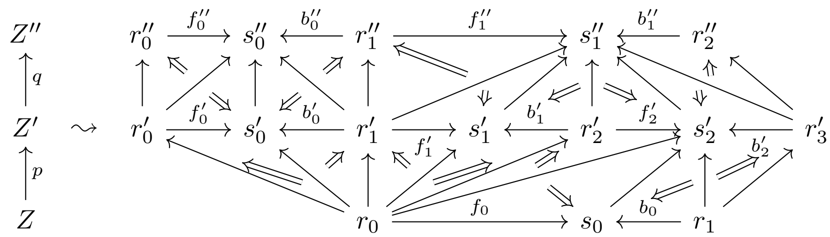


Figure 2: Exploded diagram in C of $Z \xrightarrow{p} Z' \xrightarrow{q} Z''$. p and q determine C-morphism labels on the vertical morphisms, which have been omitted for clarity.

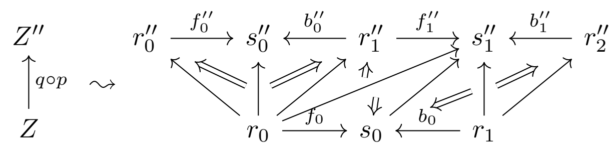


Figure 3: Exploded diagram in C of $Z \xrightarrow{q \circ p} Z''$.

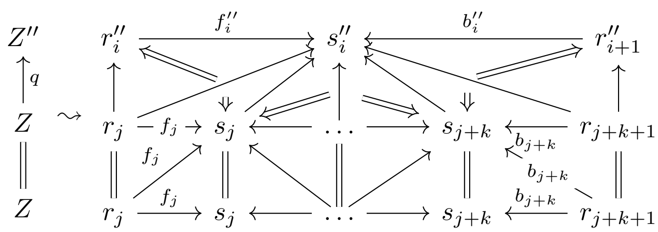


Figure 4: Exploded diagram in C of $q \circ \mathrm{id}_Z$

the C-morphism labelling is obtained on all the vertical morphisms of Figure 3 by taking the meet of all corresponding paths from Figure 2; the only instances where this meet is not determined by shape alone are for the paths $r_0 \to \{r_1', s_1', r_2', s_2'\} \to s_1''$ and $r_1 \to \{s_2', r_3'\} \to s_2''$.

The identity morphism for some framed zigzag Z in Zig (C) is generated by the identity monotone map on singular levels of Z labelled by identity morphisms with only trivial 2-cell fillers, as in the bottom half of Figure 4.

Lemma 3.4. *Composition of framed zigzag maps is unital.*

Proof. Assume that p is the identity. Every regular-to-regular or singular-to-singular component of the composite is the composition of an identity morphism with a component of q, and therefore is equal to the corresponding component of q. For a regular-to-singular component, this is derived from trivial structure of the 2-cell fillers of p. For example, consider the paths $r_j \to s_i''$ in Figure 4. Because p has trivial 2-cell fillers, this induces the path $r_j = r_j \to s_i''$ as the minimal one, which agrees with the corresponding component of q.

Similar reasoning holds for when q is the identity. □

Lemma 3.5. *Composition of framed zigzag maps is associative.*

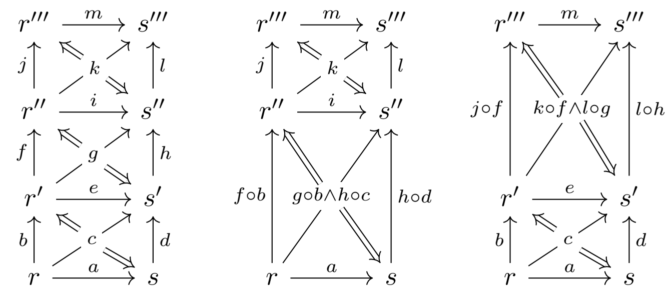


Figure 5: Fragment of $Z \to Z' \to Z'' \to Z'''$, associated to the left and right

Proof. Associativity will arise due to C being a **Pos**-category which has meets in Hom posets preserved by composition, and that this meet when considered as a binary operation is itself associative. For example, consider the fragment in Figure 5: the component $r \to s'''$ obtained by associating composition on the left is $k \circ f \circ b \wedge l \circ (g \circ b \wedge h \circ c)$, whereas on the right it is $(k \circ f \wedge l \circ g) \circ b \wedge l \circ h \circ c$. Both of these expressions are equal to $k \circ f \circ b \wedge l \circ g \circ b \wedge l \circ h \circ c$. □

Lemma 3.6. *Composition of framed zigzag maps is monotone.*

Proof. Let

$$\begin{tikzcd} Z \arrow[r, bend left, "p'"] \arrow[r, bend right, "p"'] \arrow[r, phantom, "\Uparrow"] & Z' \arrow[r, bend left, "q'"] \arrow[r, bend right, "q"'] \arrow[r, phantom, "\Uparrow"] & Z'' \end{tikzcd}$$

be framed zigzag maps. q and q' have the same shape as they are comparable, and each component of q' is above its corresponding component of q. Therefore, $q \circ p \Rightarrow q' \circ p$, as both are given with respect to these components. Similarly, $q \circ p \Rightarrow q \circ p'$. □

Combining all of these, we can ascertain that Zig (C) is a **Pos**-category.

Theorem 3.7. *Zig (C) is a* **Pos***-category.*

3.1.1 Iterated framed zigzags. Recall that this gadget is supposed to represent the space of 1D string diagrams of a signature encoded by C, and that $\mathrm{Zig}^n(C) := \mathrm{Zig}\left(\mathrm{Zig}^{n-1}(C)\right)$, where $\mathrm{Zig}^0(C) := C$, is supposed to represent the space of n-dimensional string diagrams. In order for this n-fold iterated construction to be well-defined, we must show the following.

PROPOSITION 3.8. *Every Hom poset in Zig (C) is locally finitely meet-complete, and these meets are preserved by composition.*

PROOF. Every poset is partitioned into connected components by the symmetric closure of its order relation. Each connected component $[p]$ of Zig (C) (Z, Z') corresponds to framed zigzag maps with a fixed shape, where every p is component-wise connected in the appropriate Hom poset of C. In other words, they exist in a fibre over the projection $\pi\colon \mathrm{Zig}(C) \to \Delta$, living over some $\alpha \in \Delta$; moreover, in the case that C is itself a framed zigzag category already, every component is also component-wise connected recursively. The meet of any finite subset of $[p]$ is given by component-wise meets in C, which exist by assumption. That composition preserves these meets is also inherited from C. □

3.2 The case for C being an arbitrary Pos-category

Definition 3.2 requires C to have locally finitely meet-complete Hom posets in order for composition to be well-defined. However, in practice, we are interested in the case where C is an arbitrary **Pos**-category. This is rectified by freely passing from C to the $\bigwedge$-**Lat**-category $F_*(C)$ as in Corollary 2.25, seen as a **Pos**-category, via base change. Such a **Pos**-category is guaranteed to have meet-complete Hom posets, so satisfies the hypotheses on Definition 3.1; moreover, because the process of moving from C to $F_*(C)$ preserves existing meets in Hom posets, we can interpret the result of composition of framed zigzag maps as follows: if it is some 'formal' meet of morphisms in any component that does not reflect back to a C-morphism, then it is considered a failure and does not represent a 'meaningful' framed zigzag map. We would expect that when Zig (C) models some theory of string diagrams, then these failures never occur.

The reason this completion needs to be done is analogous to the problem of the ordinary zigzag category $\mathrm{Zig}_0(C)$ being 'too big': informally, we want to think of $\mathrm{Zig}_0(C)$ as the space of combinatorial encodings of 1D string diagrams of a signature encoded by C, but it contains more objects which do not have any meaningful interpretation in this sense. For example, if C encodes the algebraic signature consisting of two 0-cells x and y and a single 1-cell $f\colon x \to y$, then $\mathrm{Zig}_0(C)$ contains the object $x \to f \leftarrow y$ which represents f as a string diagram, but it also has objects like $x \to f \leftarrow f$ which can only be considered ill-typed.

The idea is that framed zigzag maps with components given by meets which exist only formally under this machinery are also devoid of meaning under this interpretation, and that because $F_*(C)$ always exists we can pretend that Definition 3.1 works even for the case where C lacks some meets in its Hom posets, as will be the case in Section 4.

3.3 Colimits in C

A prominent idea in the theory of zigzag categories is that all complex homotopical structure can be built from a primitive *contraction* operation [17, §3]. The contraction operation is essentially a sequence of colimits, which in the theory of framed zigzags are replaced by oplax conical colimits. It will come to pass that an oplax conical colimit in a zigzag **Pos**-category Zig (C) is ultimately determined by oplax conical colimits in C, and this will be reflected in our recursive collapse-colimit algorithm: the base case of this algorithm is the computation of an oplax conical colimit in a **Pos**-category C which is to be thought of as a 'thick' poset encoding an algebraic signature of higher string diagrams.

This section is dedicated to describing this base case and justifying its correctness.

In the original theory, this base case takes as input a poset; a poset is a somewhat trivialised type of category — one which is both skeletal and thin. In order to explain what is meant by 'thick' poset, we observe how **Pos**-enrichment allows for a somewhat subtle un-trivialisation of the notion of a poset.

A generalised analogue of thin category (i.e. a preorder) for our enriched categories is desired. Thinness in a category is the property that every Hom set is either empty or the singleton; in other words, a thin category is a **Bool**-enriched category.

LEMMA 3.9. *There is a free-forgetful adjunction* $\mathbf{Pos} \underset{U}{\overset{F}{\rightleftarrows}} \mathbf{Bool}$.

The functor F sends a poset to true if and only if it is non-empty, and it is cartesian (and hence strong and lax monoidal). Moreover, because this functor is strong monoidal, this is a monoidal adjunction which induces the following.

LEMMA 3.10 ([7, §4.4]). *There is a 2-adjunction* $\mathbf{Pos\text{-}Cat} \underset{U_*}{\overset{F_*}{\rightleftarrows}} \mathbf{Bool\text{-}Cat}$.

In particular, this means that every category enriched in **Pos** admits a **Bool**-category by base change along F, which we call the *underlying preorder*, and this process preserves enriched colimits because F is a left adjoint.

Definition 3.11 (Underlying preorder). Let C be a **Pos**-category. The *underlying preorder* of C is the preorder $C_{\lesssim}$ of objects of C, ordered by $c \lesssim c'$ if and only if $C(c, c') \neq \emptyset$.

Definition 3.12 (Local-colimit). Given some diagram $D\colon J \to C$, for *id*-max C, the *local-colimit* is the following procedure:

(1) D induces a sub-**Pos**-category of C by its full image, which admits an underlying preorder. If this preorder does not admit a unique maximal element, then fail, otherwise obtain Dx, for some $x \in J$, with the property that for all $j \in J$, $C(Dx, Dj) \neq \emptyset \iff j = x$.
(2) If the meet of every $D_{j,x}[J(j,x)] \subseteq C(Dj, Dx)$, for $j \in J$, does not exist in the image of D then fail. Otherwise, obtain some morphism $f\colon j \to x$ of J for which $\lambda_j := Df$ forms a leg making Dx into the tip of an oplax cocone λ.

PROPOSITION 3.13. *If the procedure of Definition 3.12 succeeds, the result is the unique local oplax cocone and it is maximal in* **OplaxCocone**(D, Dx).

PROOF. First, we show that λ is an oplax cocone. For any $f\colon j \to j'$ in J, we have that for all $g\colon j' \to x$, $\lambda_j \Rightarrow Dg \circ Df$ because $D(g \circ f) \in D_{j,x}[J(j,x)]$ and λ_j is the meet of this set. Moreover, $\lambda_{j'} = Dg$ for some g, hence $\lambda_j \Rightarrow \lambda_{j'} \circ Df$. It is a local oplax cocone by construction. Uniqueness and maximality follow from Proposition 2.31. □

PROPOSITION 3.14. *If there is a unique local oplax cocone for D, then the procedure of Definition 3.12 succeeds and finds it.*

Proof. Suppose that such a local oplax cocone λ exists, with tip Dx for some $x \in J$. Then, for any $x \neq j \in J$, it must be the case that $D_{x,j}[J(x,j)] = \emptyset$; otherwise, there would be a morphism $Df: Dx \to Dj$, for which another local oplax cocone of tip Dj can be built by composition with components of λ. Therefore, step 1 succeeds, finding x.

Because λ is an oplax cocone, for any $f: j \to x$ in J, $\lambda_j \Rightarrow \lambda_x \circ Df$. $\mathcal{D}$ is id-max, so $\lambda_x \Rightarrow id$, hence $\lambda_j \Rightarrow Df$. As λ is local, $\lambda_j = Df$ for some f, therefore the meet of $D_{j,x}[J(j,x)]$ exists, so step 2 succeeds. □

Combining these, we obtain the following.

Theorem 3.15 (Correctness of Definition 3.12). *D admits a unique local oplax cocone if and only if the procedure of Definition 3.12 succeeds.*

Now, according to Definition 2.29, this local oplax cocone is the oplax conical colimit exactly when there is an order isomorphism

$$C(Dx, c) \cong \mathbf{OplaxCocone}(D, c)$$

for all $c \in C$, natural in c. That is, it suffices to check that each $f: Dx \to c$ monotonically determines a new oplax cocone by composition with λ, i.e. $\alpha_j = f \circ \lambda_j$.

In particular, if there is only one C-morphism with domain Dx, the identity, then λ is automatically the oplax conical colimit.

In order to make our computation smaller, we can leverage the fact that C is id-max and that we are interested in oplax conical colimits.

Definition 3.16 (Collapse-colimit base case). Given some diagram $D: J \to C$, for id-max C, the *collapse-colimit base case* is the following procedure:

(1) Simplify D by collapse (Definition 2.36) to obtain a smaller diagram $\nabla(D)$.
(2) Perform the local-colimit procedure of Definition 3.12 on $\nabla(D)$.

Correctness of this simplification follows from Proposition 2.37.

3.4 Colimits in Zig (C)

An oplax conical colimit of Zig (C) is built by lifting oplax conical colimits in C and gluing them together. The technical machinery of this procedure is determined by the natural projection π: Zig $(C) \to \Delta$, which is an opfibration in the case of C being cocomplete in an appropriately enriched setting. This mirrors the procedure of Reutter and Vicary [17, §3], and determines the recursive case of the collapse-colimit algorithm.

In this section, we briefly justify that correctness is retained in our enriched setting. A more detailed explanation is given in Appendix C, but the interested reader should refer to Reutter and Vicary [17, §3] for all the details; in our setting, the results are analogous, replacing 'zigzag category' with 'framed zigzag **Pos**-category' and 'colimit' with 'oplax conical colimit'.

Firstly, we establish that the singular projection functor [17, Definition 10] works in our enriched setting.

Definition 3.17 (Singular projection **Pos***-functor).* There is a **Pos**-functor π: Zig $(C) \to \Delta$ called the *singular projection* which sends a framed zigzag Z to $[n] \in \Delta$, where n is the number of singular levels of Z, and a framed zigzag map $p: Z \to Z'$ to its underlying singular monotone map. It is trivially monotone.

Remark 3. Definition 3.17 can also be obtained as the image of the terminal **Pos**-functor $C \to \mathbf{1}$ by viewing Zig $(-)$ as a **Pos**-functor **Pos**-**Cat** $\to$ **Pos**-**Cat**, along the equivalence Zig $(\mathbf{1}) \cong \Delta$.

Remark 4. Oplax conical colimits and conical colimits in Δ coincide. This is because Δ has only trivial 2-cell fillers. Similarly, oplax conical limits and conical limits in $\Delta^{=}$ coincide.

The constructions still apply, because π is a **Pos**-opfibration when C is **Pos**-cocomplete (and when C is not **Pos**-cocomplete, we can take the free **Pos**-completion along the enriched dual Yoneda embedding), meaning that oplax conical colimits can be lifted along π under suitable assumptions.

Definition 3.18 (Collapse-colimit recursive case). Given some connected diagram $D: J \to$ Zig (C), the *collapse-colimit recursive case* is the following procedure:

(1) build the oplax conical colimit of the composite diagram $\pi \circ D$; if this does not exist then fail, otherwise we obtain some object $[n] \in \Delta$ equipped with an oplax cocone of monotone maps, one for each $j \in J$, which determines the singular projection of the result; along the isomorphism $\Delta \cong \Delta_{=}^{\mathrm{op}}$ we also obtain an oplax conical colimit $[n+1] \in \Delta_{=}^{\mathrm{op}}$ which determines the regular projection of the result; together, these fix the *shape* of the result framed zigzag and component maps into it;
(2) recalling that $D: J \to$ Zig (C) is equivalent to a larger diagram $E(D): E(J) \to C$ by explosion, obtain a sequence of diagrams
   (a) for $s_i \in [n]$, $E(D)_{s_i}: E(J)_{s_i} \to C$ dictated by the reverse-reachability closure of s_i with respect to $E(J)$;
   (b) subdiagrams of this for $r_i, r_{i+1} \in [n+1]$, $E(D)_{r_i}$ and $E(D)_{r_{i+1}}$, analogously;
   (c) recursively compute the collapse-colimit for $E(D)_{r_i}$, $E(D)_{s_i}$, and $E(D)_{r_{i+1}}$, failing if any of these fail;
   (d) the oplax conical colimits obtained from this determines the fragment of the result framed zigzag corresponding to singular height i and its adjacent regular heights, with the legs of the oplax cocones determining contiguous fragments of component maps for each framed zigzag Dj for $j \in J$;
(3) glue together these fragments to form both the framed zigzag which is the oplax conical colimit of D, and framed zigzag maps into it forming an oplax cocone over D; the fact that these fragments are compatible on the boundary and hence can be glued results from Proposition C.4.

Remark 5. Reutter and Vicary [17] make the additional assumption of *globularity*: that every regular-to-regular component of a zigzag map is an identity morphism. This assumption makes sense in the context of modelling higher string diagrams, as every such one will be globular in this sense, but is not required for the constructions to work. Here we have presented the algorithm in a way which does not require this assumption, generalised to framed zigzags.

Remark 6. This generalises the composition of framed zigzag maps (Definition 3.2), which can be seen as this computation over the diagram $q \circ p$ for two composable framed zigzag maps p and q.

Theorem 3.19 (Correctness of Definition 3.18 [17, Theorem 33]). *For a* **Pos***-category C which admits a terminal object, a connected diagram $D\colon J \to Zig(C)$ admits an oplax conical colimit if and only if the procedure of Definition 3.18 succeeds, i.e. the oplax conical colimits in steps 1 and 2c exist and the procedure constructs them.*

3.5 Unframing

The framed zigzag construction should be a conservative extension of the original theory: the admissible operations in the framed theory, restricting to not using any invertibility, are exactly the ones that were present in the unframed theory of Reutter and Vicary [17].

Definition 3.20 (Unframing). Let $\mathrm{Zig}^n(C)$ be a framed zigzag **Pos**-category. The *unframing* of $\mathrm{Zig}^n(C)$ is the zigzag category determined recursively by:

$n = 0$ the underlying preorder $C_{\lesssim}$;
$n > 0$ $\mathrm{Zig}_0\left(\text{unframing of } \mathrm{Zig}^{n-1}(C)\right)$.

Effectively, the framing is discarded in the base category. This extends to a **Pos**-functor $\mathrm{Zig}^n(C) \to \mathrm{Zig}_0^n(C_{\lesssim})$, seeing the latter as the free **Pos**-category on the category $\mathrm{Zig}_0^n(C_{\lesssim})$ with trivial 2-cell fillers (Lemma 2.14), allowing us to talk about the unframing of framed zigzag maps and diagrams in a framed zigzag **Pos**-category.

Proposition 3.21. *Let $D\colon J \to Zig^n(C)$ be a diagram in a framed zigzag* **Pos***-category which admits an oplax conical colimit. Then the unframing of D admits a colimit, obtained by unframing the components of the oplax conical colimit of D.*

Proof. The unframing **Pos**-functor is a left adjoint, and hence preserves weighted **Pos**-colimits: for the case that $n = 0$, this is the adjunction of Lemma 3.10; for $n > 0$, this is the adjunction of Lemma 2.14. □

Example 3.22 (Unframing does not reflect colimits). Consider the case where C is the **Pos**-category with two objects x and f determined by $C(x, f) := \{1, 2\}$ with 1 and 2 incomparable. $C_{\lesssim}$ is the poset given by $x \leq f$. We have the following diagrams in C and $C_{\lesssim}$ respectively:

$$x \xrightarrow{1} f \xleftarrow{2} x \xrightarrow{1} f \xleftarrow{2} x \quad \overset{\text{unframe}}{\rightsquigarrow} \quad x \to f \leftarrow x \to f \leftarrow x;$$

the former does not admit an oplax conical colimit, because there is no meet of $\{1, 2\}$ in $C(x, f)$ and hence no leg for the middle x, but the latter does: f with legs $x \to f$ and id_f as appropriate.

This example corresponds to an 'ill-typed' contraction [17, Example 32] in the unframed theory, which is a valid contraction that is later rejected by the type-checking procedure described by Heidemann et al. [9], and demonstrates how our theory is more 'type-aware'.

4 COHERENT INVERTIBILITY

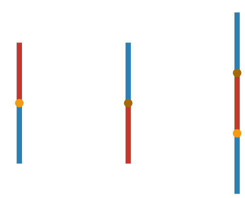
Here we demonstrate some examples of framed zigzag **Pos**-categories which model the setting for n-dimensional string diagrams, with respect to some algebraic signature, which admit coherently invertible cells. For lack of a better scheme, morphisms in the base **Pos**-category (which represent framing) are simply indexed by positive natural numbers, but the reader is reminded that they are supposed to signify *direction* in the context of an n-dimensional string diagram.

Each example is constructed inside homotopy.io, and can be easily reproduced by the interested reader[8]. Further guidance on the use of the system is available [16, 11].

4.1 Invertible 1-cell

The algebraic signature to be modelled consists of two 0-cells x and y, and a single 1-cell $f\colon x \to y$ which admits an inverse $f^{-1}\colon y \to x$.

The base **Pos**-category in this case is the **Pos**-category C given by the diagram $x \xrightarrow{1} f \xleftarrow{2} y$. 1-dimensional string diagrams over this signature are encoded by objects of $\mathrm{Zig}(C)$ (C trivially satisfies the hypotheses in Definition 3.1) such as $x \xrightarrow{1} f \xleftarrow{2} y$, $y \xrightarrow{2} f \xleftarrow{1} x$, and $x \xrightarrow{1} f \xleftarrow{2} y \xrightarrow{2} f \xleftarrow{1} x$, which respectively represent f, f^{-1} and the composite $f^{-1} \circ f$ as string diagrams:



Here the coloured wires represent x and y, with f and f^{-1} being coloured points. In our string diagram convention, diagrams go from bottom-to-top.

The cancellation of $f^{-1} \circ f = \mathrm{id}_x$ seen directionally left-to-right as a 2-cell is represented by the 2-dimensional string diagram:

This is encoded by some object $\bullet \in \mathrm{Zig}^2(C)$, with projections in $\mathrm{Zig}(C)$ and C[9]:

$$\begin{tikzcd}
 & r_1 \arrow[d] & & & & x \arrow[d, "1"] & & \\
\bullet \arrow[r, rightsquigarrow] & s_0 \arrow[r, rightsquigarrow] & x \arrow[urr, equal] \arrow[rr, "1"] \arrow[d, equal] & & & f & & x \arrow[ull, equal] \arrow[ll, "1"'] \arrow[d, equal] \\
 & r_0 \arrow[u] & x \arrow[r, "1"'] \arrow[urrr, "1"] & f \arrow[urr, equal] & & y \arrow[l, "2"] \arrow[r, "2"'] \arrow[u, "2"] & f \arrow[ul, equal] & x. \arrow[l, "1"] \arrow[ulll, "1"']
\end{tikzcd}$$

In particular, this diagram is obtained by contracting the 1-dimensional string diagram $f^{-1} \circ f$, and observe that f is the oplax conical colimit of the directed diagram

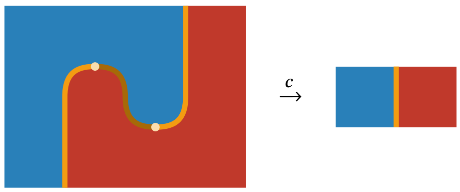
$$x \xrightarrow{1} f \xleftarrow{2} y \xrightarrow{2} f \xleftarrow{1} x,$$

as given by Definition 3.12, and this determines the framed zigzag map $r_0 \to s_0$ above.

[8] In order for the system to allow the inverse of an n-cell for $n > 1$ to be attached, that cell must be marked 'invertible' inside the signature.

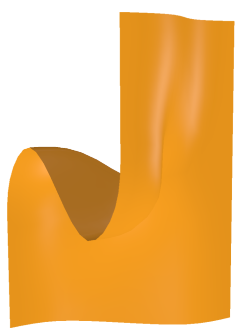
[9] All 2-cell fillers are trivial, so have been elided in the diagram.

An example witness to f^{-1} being a *coherent* inverse to f is given by the framed zigzag map of the following form:



Crucially, such a framed zigzag map is obtained by contraction, and we conjecture that all such coherences in every dimension are also obtainable this way.

This can also be seen as part of a 3-cell, which is the 3-dimensional string diagram:



The front face of this picture is encoded by the domain of c, and the back face by its codomain. Scanning continuously from front-to-back, c itself encodes the homotopical deformation of 'straightening the snake'. The 3-cell c is itself coherently invertible, and will admit an infinite family of coherences in the same way as f.

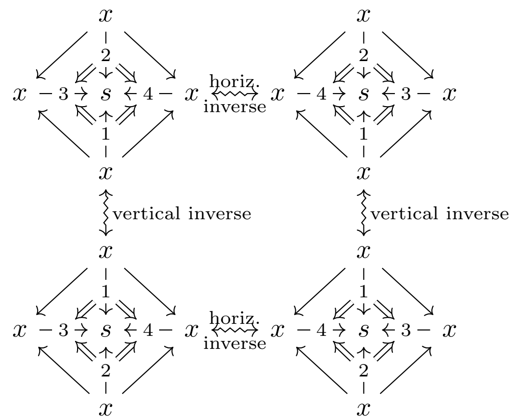
4.2 Invertible 2-cell scalar

In this example, we model the algebraic signature consisting of a single 0-cell x, and a single 2-cell scalar $s\colon \mathrm{id}_x \to \mathrm{id}_x$ which admits vertical and horizontal inverses of the same type.

The base **Pos**-category in this case is the **Pos**-category C given by two objects x and s, with the only non-identity morphisms being

$$C(x,s) \coloneqq \begin{array}{c} 3 \\ | \\ 1 \end{array} \times \begin{array}{c} 4 \\ | \\ 2 \end{array}$$

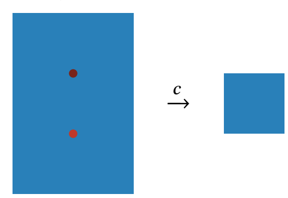
Observe that $C(x,s)$ is not meet-complete (e.g. there is no meet of $\{1,2\}$), so technically we rely on Section 3.2 to obtain an iterated framed zigzag **Pos**-category. The objects of $\mathrm{Zig}^2(C)$, drawn as diagrams in C, which represent s and its inverses are:



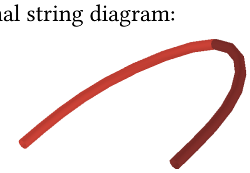
The top-left diagram represents s itself, with its vertical inverse s^{-1} being represented by its reflection in the horizontal axis below, and its horizontal inverse being represented by its reflection in the vertical axis to the right.

Even though as diagrams in C these all appear to contain the same data, as objects of $\mathrm{Zig}^2(C)$ they are distinct. This example also illustrates the necessity of **Pos**-enrichment and non-trivial 2-cell fillers: without that, commutativity of all the triangles would be imposed, which in turn determines $1 = 2 = 3 = 4$ and removes the ability for s to be distinguished from its inverses in $\mathrm{Zig}^2(C)$.

The cancellation of $s^{-1} \circ s = \mathrm{id}_{\mathrm{id}_x}$ is again given by the framed zigzag map generated by contraction:



or as a 3-dimensional string diagram:



That this inverse is coherent says, among other things, that the 3D snake that can be constructed by extending this also admits a homotopy obtained by contraction which pulls it straight.

In Appendix A we use an invertible scalar to illustrate a formal proof of Theorem 1.1, as stated in the introduction.

REFERENCES

[1] Krzysztof Bar, Aleks Kissinger and Jamie Vicary. 2018. Globular: an online proof assistant for higher-dimensional rewriting. *Logical Methods in Computer Science*, Volume 14, Issue 1, (22nd Jan. 2018). DOI: 10.23638/LMCS-14(1:8)2018.

[2] Thibaut Benjamin, Eric Finster and Samuel Mimram. 2021. Globular weak ω-categories as models of a type theory. (8th June 2021). arXiv: 2106.04475 [cs, math]. Retrieved 27th Jan. 2024 from http://arxiv.org/abs/2106.04475. preprint.

[3] Benno van den Berg and Richard Garner. 2011. Types are weak omega-groupoids. *Proceedings of the London Mathematical Society*, 102, 2, (Feb. 2011), 370–394. arXiv: 0812.0298 [math]. DOI: 10.1112/plms/pdq026.

[4] Guillaume Brunerie. 2016. *On the Homotopy Groups of Spheres in Homotopy Type Theory*. PhD thesis. University of Nice Sophia Antipolis, (19th June 2016). Retrieved 23rd Jan. 2024 from http://arxiv.org/abs/1606.05916 arXiv: 1606.05916 [cs, math].

[5] Mitchell Buckley. 2014. Fibred 2-categories and bicategories. *Journal of Pure and Applied Algebra*, 218, 6, (1st June 2014), 1034–1074. DOI: 10.1016/j.jpaa.2013.11.002.

[6] 2024. Catastrophe theory. In *Wikipedia*. (6th Jan. 2024). Retrieved 27th Jan. 2024 from https://en.wikipedia.org/w/index.php?title=Catastrophe_theory&oldid=1193972222.

[7] G S H Cruttwell. 2008. *Normed Spaces and the Change of Base for Enriched Categories*. PhD thesis. Dalhousie University, Halifax, Nova Scotia, (Dec. 2008). 152 pp. https://www.reluctantm.com/gcruttw/publications/thesis4.pdf.

[8] Christoph Dorn. 2018. *Associative n-categories*. University of Oxford, (26th Dec. 2018). Retrieved 14th Oct. 2019 from http://arxiv.org/abs/1812.10586 arXiv: 1812.10586.

[9] Lukas Heidemann, David Reutter and Jamie Vicary. 2022. Zigzag normalisation for associative n-categories. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*. 37th Annual ACM/IEEE Symposium on Logic in Computer Science. ACM, Haifa Israel, (18th May 2022). arXiv: 2205.08952 [cs, math]. DOI: 10.1145/3531130.3533352.

[10] Claudio Hermida. 1999. Some properties of Fib as a fibred 2-category. *Journal of Pure and Applied Algebra*, 134, 1, (5th Jan. 1999), 83–109. DOI: 10.1016/S0022-4049(97)00129-1.

[11] [n. d.] Homotopy-rs/TUTORIAL.md at master · homotopy-io/homotopy-rs. Retrieved 18th Jan. 2024 from https://github.com/homotopy-io/homotopy-rs/blob/master/TUTORIAL.md.

[12] 2024. Homotopy.io figure-8. (2024). https://www.youtube.com/watch?v=7kW5rI-mmeU.

[13] Niles Johnson and Donald Yau. 2020. 2-Dimensional Categories. (17th June 2020). arXiv: 2002.06055 [math]. Retrieved 23rd Dec. 2023 from http://arxiv.org/abs/2002.06055. preprint.

[14] G. M. Kelly. 1982. *Basic Concepts of Enriched Category Theory*. Number 64 in *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge ; New York. 245 pp. ISBN: 978-0-521-28702-9.

[15] Vidit Nanda. 2019. Discrete Morse theory and localization. *Journal of Pure and Applied Algebra*, 223, 2, (Feb. 2019), 459–488. arXiv: 1510.01907 [math]. DOI: 10.1016/j.jpaa.2018.04.001.

[16] nLab authors. [n. d.] Homotopy.io. homotopy.io. https://ncatlab.org/nlab/show/homotopy.io.

[17] David Reutter and Jamie Vicary. 2019. High-level methods for homotopy construction in associative n-categories. In 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS). IEEE, Vancouver, Canada, (11th Feb. 2019). arXiv: 1902.03831. DOI: 10.1109/LICS.2019.8785895.

[18] Alex Rice. 2020. Coinductive Invertibility in Higher Categories. (17th Oct. 2020). arXiv: 2008.10307 [math]. Retrieved 27th Jan. 2024 from http://arxiv.org/abs/2008.10307. preprint.

[19] Emily Riehl. 2014. *Categorical Homotopy Theory*. (1st ed.). Cambridge University Press, (26th May 2014). ISBN: 978-1-107-04845-4 978-1-107-26145-7. DOI: 10.1017/CBO9781107261457.

[20] Chiara Sarti and Jamie Vicary. 2023. Posetal Diagrams for Logically-Structured Semistrict Higher Categories. *Electron. Proc. Theor. Comput. Sci.*, 397, (14th Dec. 2023), 246–259. arXiv: 2305.11637 [cs, math]. DOI: 10.4204/EPTCS.397.15.

[21] Ross Street. 1976. Limits indexed by category-valued 2-functors. *Journal of Pure and Applied Algebra*, 8, 2, (1st June 1976), 149–181. DOI: 10.1016/0022-4049(76)90013-X.

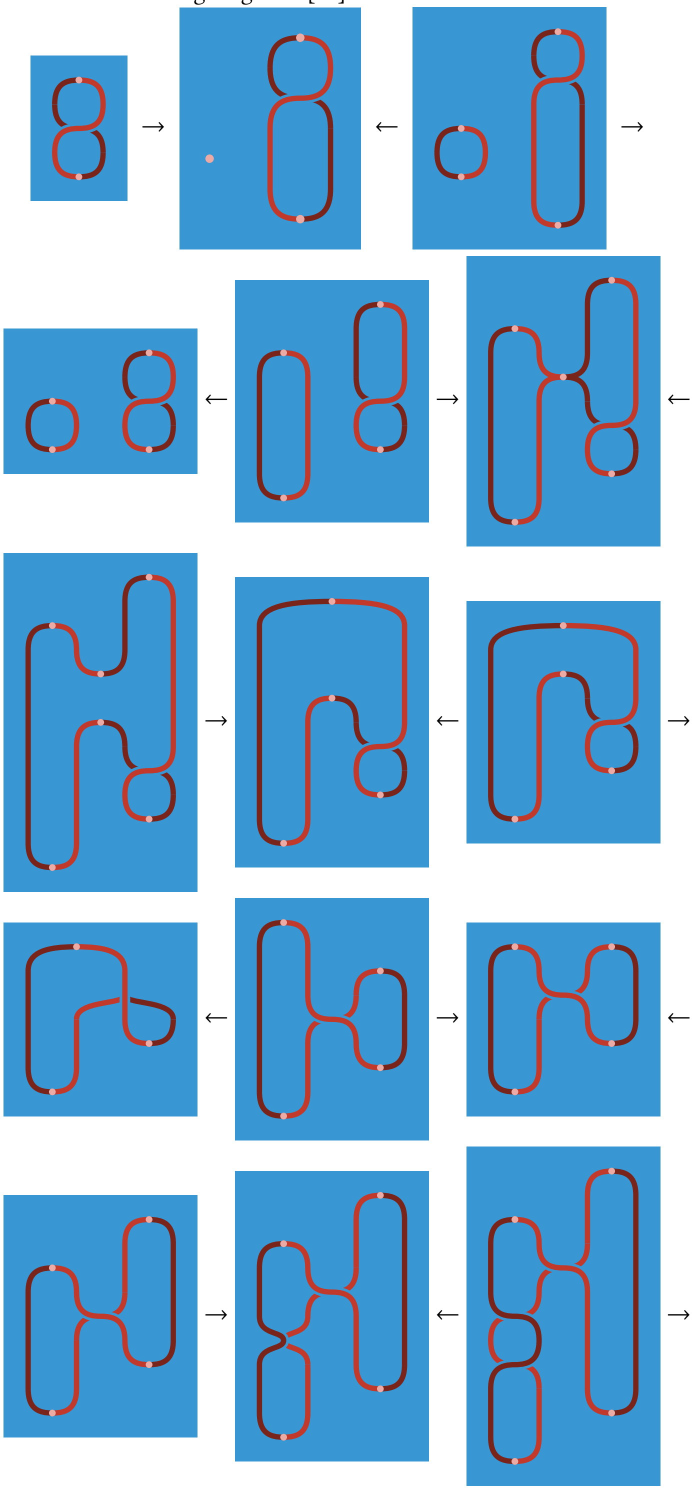
[22] Calin Tataru and Jamie Vicary. 2023. A layout algorithm for higher-dimensional string diagrams. (11th May 2023). arXiv: 2305.06938 [math]. Retrieved 31st Oct. 2023 from http://arxiv.org/abs/2305.06938. preprint.

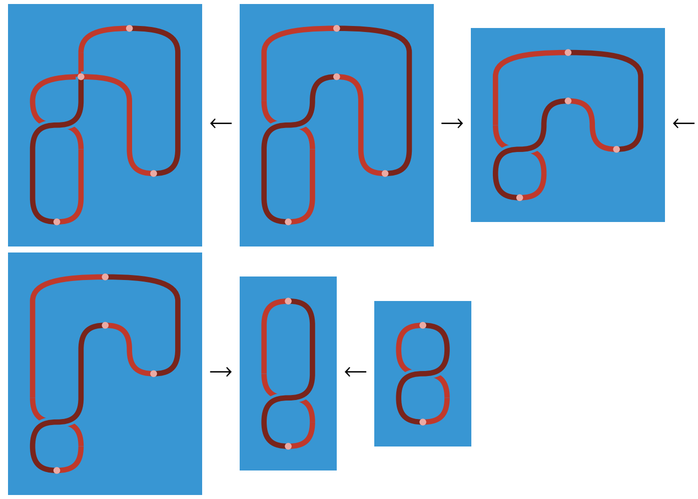
[23] Univalent Foundations Program. 2013. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study.

[24] Gavin C Wraith. 1993. Using the generic interval. *Cahiers de topologie et géométrie différentielle catégoriques*, 34, 4, 259–266.

A THE FIGURE-8 THEOREM

Here we illustrate a formalised proof of Theorem 1.1 as stated in the introduction. This proof can also be viewed as a movie of 3-dimensional string diagrams [12].





B PATHOLOGIES OF OPLAX CONICAL COLIMITS

In this section, we collect some examples where oplax conical colimits fail to exist in pathological ways.

Lemma 2.32 shows that whenever a diagram admits a terminal object in its indexing category, the colimit of this diagram exists; this is not true in the case of oplax conical colimit.

Example B.1. Let F be the diagram over the walking arrow (which has a terminal object) which chooses g in the **Pos**-category on the right:

$$\begin{tikzcd} \cdot \arrow[r] & \cdot & \overset{F}{\longmapsto} & x \arrow[r, bend left, "g"] \arrow[r, bend right, "f"'] \arrow[r, phantom, "\Uparrow"] & y. \end{tikzcd}$$

There are two oplax cocones over F, both with tip y, with legs given by f and g respectively; however, neither cocone factorises via the other, so no oplax conical colimit exists.

However, if F had chosen f instead, then the oplax conical colimit would exist because only one oplax cocone would exist.

Proposition 2.37 has a condition that collapse only preserves oplax conical colimits when the **Pos**-category is *id*-max. Here is an example showing that this condition is necessary.

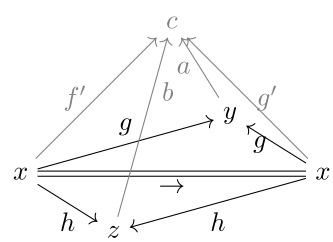
Example B.2. Let C be the category determined by

$$\begin{aligned} C(x,y) &:= \{f \Rightarrow g\}, & C(x,z) &:= \{h\}, \\ C(y,c) &:= \{a\}, & C(z,c) &:= \{b\}, \\ C(x,c) &:= \{f' \Rightarrow g' \Rightarrow h'\}, & C(c,c) &:= \{u^- \Rightarrow \mathrm{id}_c \Rightarrow u^+\}, \end{aligned}$$

with composition given by

$$\begin{aligned} a \circ f &= f', & a \circ g &= g', \\ b \circ h &= h', \\ u^+ \circ f' &= g', & u^- \circ f' &= f', \\ u^- \circ g' &= g', & u^+ \circ f' &= f'. \end{aligned}$$

The diagram



admits an oplax conical colimit c, however this is not preserved by its collapsed diagram $z \xleftarrow{h} x \xrightarrow{g} y$.

C HIGH-LEVEL METHODS REVISITED

We briskly develop the theory of **Pos**-opfibrations, based the theory of 2-fibrations [10, 5]. Fix a **Pos**-functor $p\colon \mathcal{E} \to \mathcal{B}$.

*Definition C.1 (**Pos**-opcartesian morphism).* A morphism $f\colon x \to y$ in $\mathcal{E}$ is *p-opcartesian* if for any $g\colon x \to z$ in $\mathcal{E}$ and $w\colon py \to pz$ in $\mathcal{B}$ such that $pg = w \circ pf$, there is a unique morphism $\hat{w}\colon y \to z$ such that $g = \hat{w} \circ f$. In pictures:

$$\begin{tikzcd} & \mathcal{E} \arrow[rrr, "p"] & & & \mathcal{B} & \\ & z & & & & pz \\ x \arrow[ur, "\forall g"] \arrow[r, "f"'] & y \arrow[u, dashed, "\exists!\hat{w}"'] & \longmapsto & & px \arrow[ur, "pg"] \arrow[r, "pf"'] & py. \arrow[u, "\forall w"'] \end{tikzcd}$$

$\hat{w}$ is the p-opcartesian lift of w.

*Definition C.2 (**Pos**-opcartesian 2-cell filler).* Let $f, g\colon x \rightrightarrows y$ in $\mathcal{E}$. A 2-cell filler $f \Rightarrow g$ is *p-opcartesian* if for all $h \Rightarrow f$ such that $pf \Rightarrow pg \Rightarrow ph$, $g \Rightarrow h$:

$$\begin{tikzcd} \mathcal{E}(x,y) \arrow[rrr, "p"] & & & \mathcal{B}(px,py) \\ & h & & ph \\ f \arrow[ur, Rightarrow, "\forall"] \arrow[r, Rightarrow] & g \arrow[u, Rightarrow, dashed, "\exists"'] & \longmapsto \quad pf \arrow[ur, Rightarrow] \arrow[r, Rightarrow] & pg. \arrow[u, Rightarrow, "\forall"'] \end{tikzcd}$$

*Definition C.3 (**Pos**-opfibration).* p is a **Pos**-opfibration when

(1) for each $e \in \mathcal{E}$, and $w\colon pe \to b$ in $\mathcal{B}$, there is a p-opcartesian lift $\hat{w}\colon e \to e'$:

$$\begin{tikzcd} e \arrow[r, "\hat{w}"] & e' & \mathcal{E} \arrow[d, "p"] \\ pe \arrow[r, "w"'] & b & \mathcal{B}, \end{tikzcd}$$

(2) for each $f\colon e \to e'$ in $\mathcal{E}$ such that there is a $u\colon pe \to pe'$ with $pf \Rightarrow u$, there is another morphism $g\colon e \to e'$ admitting a p-opcartesian 2-cell filler $f \Rightarrow g$, such that $pg = u$:

$$\begin{tikzcd} e \arrow[r, bend left, "g"] \arrow[r, bend right, "f"'] \arrow[r, phantom, "\Uparrow\exists"] & e' & \mathcal{E} \arrow[d, "p"] \\ pe \arrow[r, bend left, "u"] \arrow[r, bend right, "pf"'] \arrow[r, phantom, "\Uparrow\forall"] & pe' & \mathcal{B}, \end{tikzcd}$$

(3) the horizontal composition of p-opcartesian 2-cells is p-opcartesian.

Remark 7. The first part of Definition C.3 is the familiar lifting property of opfibrations from ordinary category theory; the second part says that the monotone map induced by p on locally each Hom poset is also an opfibration, treating those Hom posets as categories and the monotone map as a functor; the third part says that the property of being a p-opcartesian 2-cell filler is closed under whiskering by morphisms.

Proposition C.4. *If C is **Pos**-cocomplete, then $\pi\colon \mathit{Zig}\,(C) \to \mathbf{\Delta}$ is a **Pos**-opfibration.*

Proof. This will follow quite easily from the fact that the 2-cell filler structure of $\mathbf{\Delta}$ is trivial: each $\mathbf{\Delta}([m],[n])$ is a discrete poset where elements are comparable if and only if they are equal. Therefore, it suffices to show that given a framed zigzag Z of length m, and a monotone map $\alpha\colon [m] \to [n]$, α lifts to a framed zigzag map $\hat{\alpha}\colon Z \to Z'$ for some framed zigzag Z' of length n:

$$\begin{tikzcd} Z \arrow[r, "\hat{\alpha}"] & Z' & \mathrm{Zig}\,(\mathcal{C}) \arrow[d, "\pi"] \\ {[m]} \arrow[r, "\alpha"'] & {[n]} & \mathbf{\Delta}\,. \end{tikzcd}$$

Analogously to Reutter and Vicary [17, Proposition 36], Z' is constructed by 'collapsing' adjacent singular levels of Z by taking oplax conical colimits, and creating new singular levels by 'expanding' regular levels along identities, as dictated by α. □

Proposition C.5 ([10, §5]). *Let $D\colon J \to \mathcal{E}, p\colon \mathcal{E} \to \mathcal{B}$ be diagrams of **Pos**-categories, such that:*

(1) *p is a **Pos**-opfibration;*
(2) *the composite $p \circ D$ admits an oplax conical colimit;*
(3) *every fibre $p^{-1}(b)$ admits J-indexed oplax conical colimits, which are preserved by base change.*

Then D admits an oplax conical colimit, and it is preserved by p.

The idea of the proof is that enough structure is present to ensure that all the morphisms forming the legs of the oplax conical colimit in the composite diagram $p \circ D$ can be canonically lifted along the opfibration to form an oplax cocone over D, which will be the oplax conical colimit.

As with Reutter and Vicary [17], we argue separately for the soundness and completeness of the procedure of Definition 3.18. Soundness means that the result of the procedure is an oplax conical colimit, and completeness means that if the oplax conical colimit exists it can be obtained from the procedure.

C.0.1 Soundness.

Corollary C.6. *Let C be* **Pos***-cocomplete, and $D\colon J \to Zig(C)$ be a connected diagram such that the composite $\pi \circ D$ admits an oplax conical colimit. Then D admits an oplax conical colimit, which is preserved by π.*

Similarly to Reutter and Vicary [17], the **Pos**-categories C from which we start are seldom **Pos**-cocomplete, and so we leverage the *enriched* dual Yoneda embedding of $C \overset{y}{\hookrightarrow} \hat{C} \coloneqq [C, \mathbf{Pos}]^{\mathrm{op}}$ of a **Pos**-category into its free **Pos**-*completion*. This completion will admit all **Pos**-weighted colimits, and the embedding preserves and reflects these [14, §3.3], creating new 'formal' **Pos**-weighted colimits for those which are not already present in C. It also induces a fully faithful **Pos**-functor $\mathrm{Zig}(C) \overset{\mathrm{Zig}(y)}{\hookrightarrow} \mathrm{Zig}\left(\hat{C}\right)$. Note that oplax conical colimits are just **Pos**-weighted colimits of a specific weight (see Appendix D). Then, starting from $\hat{C}$, one can observe that the oplax conical colimit obtained from Corollary C.6 is in the image of this **Pos**-functor (analogous to Reutter and Vicary [17, Proposition 38]) and is reflected to an oplax conical colimit in C. In effect, the failure cases captured by our algorithm represent exactly when a 'formal' oplax conical colimit not already existing in C persists in the final result, and is analogous to the completion of Section 3.2. Note that $\hat{C}$ also admits a terminal object, as it is the free **Pos**-completion of C.

C.0.2 Completeness.

Lemma C.7. *If C has a terminal object, then $\pi\colon Zig(C) \to \Delta$ preserves connected oplax conical colimits.*

Proof. Recall that any oplax conical limit in $\Delta^{=}$ is a conical limit. The result follows as analogously to Reutter and Vicary [17, Proposition 39], using the fact that, for a **Pos**-functor $F\colon C \to \mathcal{D}$, the forgetful **Pos**-functor $F/d \to C$ preserves connected conical colimits. ◻

D OPLAX CONICAL COLIMITS ARE COLIMITS OF ANOTHER WEIGHT

In this section, we show that oplax conical colimits are weighted colimits for a certain weight. This means that they exist in any cocomplete **Pos**-category (which admit all weighted colimits). The original result is due to Street [21], who showed it in the case of 2-categories; we give a more explicit form.

First, we recall the notion of weighted colimit for **Pos**-categories.

*Definition D.1 (***Pos***-weighted colimit).* A weighted colimit of a diagram $F\colon J \to C$ weighted by $W\colon J^{\mathrm{op}} \to \mathbf{Pos}$, $\mathrm{colim}^W F$, is an object of C with **Pos**-representation

$$C\left(\mathrm{colim}^W F, c\right) \cong \left[J^{\mathrm{op}}, \mathbf{Pos}\right] \left(W, C\left(F-, c\right)\right).$$

Explicitly, this equips $\mathrm{colim}^W F$ with a family of morphisms $\left(\iota_{j,w}\colon Fj \to \mathrm{colim}^W F\right)_{j\in J, w\in Wj}$ such that for all $f\colon j \to j'$ and $w \in Wj'$:

$$\begin{tikzcd} & \mathrm{colim}^W F & \\ Fj \arrow[ur, "\iota_{j,Wf(w)}"] \arrow[rr, "Ff"'] & & Fj' \arrow[ul, "\iota_{j',w}"'] \end{tikzcd}$$

commutes, and this is universal: any other object with such a family of morphisms factors uniquely via $\mathrm{colim}^W F$. This factorisation and the mapping $Wj \to C(Fj, \mathrm{colim}^W F)$ are required to be monotone.

Remark 8. W is a generalisation of the shape of the cocone. A natural transformation $\alpha_j\colon Wj \to C(Fj, c)$ is a collection of maps which act as legs of this generalised cocone, varying monotonically over weights $w \in Wj$; the naturality property asserts that for any $f\colon j \to j'$, varying the weight determined by a specific leg $Fj' \to c$ by mapping along Wf is equal to precomposing that leg by Ff — in other words, the previous triangle commutes replacing c for $\mathrm{colim}^W F$. The order isomorphism between the poset of these natural transformations and the Hom poset $C(\mathrm{colim}^W F, c)$ then establishes the universal property: firstly that a morphism $\mathrm{colim}^W F \to c$ is in bijective correspondence with a particular generalised cocone of tip c, and that moreover this bijection should preserve and reflect orders[10]. That is, the order on factoring maps $C(\mathrm{colim}^W F, c)$ is completely determined by cocones, one above another exactly when the target cocones are ordered as such.

The ordinary (conical) colimit is obtained by taking W to be the constant functor at the terminal object of **Pos**.

Proposition D.2 (Oplax colimits are weighted colimits [21]). *Every oplax colimit of $F\colon J \to C$ is equivalently given by a colimit of F weighted by some $W\colon J \to$* **Pos***.*

Corollary D.3. *For* **Pos***-cocomplete C, C admits all oplax colimits.*

The explicit description of this transformation is given as follows.

Definition D.4 (Oplax conical weight). Let $F\colon J \to C$ be a diagram. The *oplax conical weight* of F is the contravariant **Pos**-functor $L_J\colon J^{\mathrm{op}} \to \mathbf{Pos}$ which sends j to the set of morphisms in J with domain j, ordered by $u \leq v$ whenever there exists some J-morphism which oplaxly extends u to v:

$$\begin{tikzcd} & y \\ j \arrow[ur, "v"] \arrow[r, "u"'] & x. \arrow[u, "\exists"'] \arrow[ul, phantom, "\Rightarrow"] \end{tikzcd}$$

A J-morphism $f\colon j \to j'$ is sent the monotone map of precomposition with f, and Hom posets inherit the ordering on J-morphisms.

Corollary D.5. *Every oplax conical colimit of $F\colon J \to C$ is equivalently given by a* **Pos***-colimit of F weighted by L_J.*

Example D.6 (Example B.1 revisited). Consider again the diagram F in Example B.1. Its oplax conical weight L_J is given by $0 \mapsto \{\mathrm{id}_0 \leq 0 \to 1\}$ and $1 \mapsto \{\mathrm{id}_1\}$, with the unique morphism $0 \to 1$ mapping contravariantly to the monotone map $\mathrm{id}_1 \mapsto 0 \to 1$. A

[10] Recall that the natural transformations α are ordered pointwise by components.

colimit of F weighted by L_J is a universal object y and a family of maps $\iota_{0,\mathrm{id}_0} \Rightarrow \iota_{0,0\to1} : x \to y$ and $\iota_{1,\mathrm{id}_1} : y \to y$ making

$$\begin{tikzcd} & y & \\ x \arrow[ur, "\iota_{0,0\to 1}"] \arrow[rr, "g"'] & & y \arrow[ul, "\iota_{1,\mathrm{id}_1}"'] \end{tikzcd}$$

commute. The only choice for ι_{1,id_1} is id_y, which in turn forces $\iota_{0,0\to1}$ to be g, but the only constraint on ι_{0,id_0} is that it is below g, and there is no universal choice between f and g; this yields two generalised cocones which do not factor through each other.

If F is replaced by the diagram that chooses f instead, which replaces f for g in the previous triangle, then $\iota_{0,0\to1}$ is forced to be f, eliminating the possibility that $\iota_{0,\mathrm{id}_0} = g$ — there is a unique generalised cocone in this case, which is trivially universal, and from this we would correctly deduce the existence of an oplax conical colimit with leg f.