Zigzag normalisation for associative n-categories

ACM Reference Format:
. 2022. Zigzag normalisation for associative n-categories. In *Proceedings of 37th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2022).* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/1122445.1122456

Abstract

The theory of associative n-categories has recently been proposed as a strictly associative and unital approach to higher category theory. As a foundation for a proof assistant, this is potentially attractive, since it has the potential to allow simple formal proofs of complex high-dimensional algebraic phenomena. However, the theory relies on an implicit term normalisation procedure to recognize correct composites, with no recursive method available for computing it.

Here we describe a new approach to term normalisation in associative n-categories, based on the categorical zigzag construction. This radically simplifies the theory, and yields a recursive algorithm for normalisation, which we prove is correct. Our use of categorical lifting properties allows us to give efficient proofs of our results. Our normalisation algorithm forms a core component of a proof assistant, and we illustrate our scheme with worked examples.

1 Introduction

1.1 Overview

Motivation. The flexibility of weak higher categories has enabled their wide use across many areas of mathematics, computer science, and physics. The most well-known include the homotopy type theory programme on univalent foundations for mathematics [2, 23, 25], motivated by the intensional groupoid model for Martin-Löf type theory [11]; Lurie's outline proof [15] of the cobordism hypothesis of Baez and Dolan [3], and the associated new perspective it brought for topological quantum field theory [1, 21]; and the higher topos theory programme [16], with broad implications for both logic and geometry, which develops ideas going back to Grothendieck [9]. In computer science, other applications include rewriting [10, 13, 18], quantum computation [12, 19], and concurrency [4, 7].

LICS 2022, Israel, 2022

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

In a weak higher category, equations hold only up to higher coherence data, which itself satisfies further equations up to coherence data, and so on ad infinitum, yielding a bureaucratic syntax in which conceptually simple proofs can become long-winded. Traditionally, this has been the price that must be paid for proof-relevance. Strict models [14, Section 1.4] discard this coherence data, but at the cost of expressivity, since not every weak higher category is equivalent to a strict one.

Associative n-categories (ANCs) are a new *semistrict* model that aims to strike a balance between these extremes: having enough strictness for practical use, while retaining sufficient weakness to remain conjecturally equivalent to the fully general case [5, 6, 20]. However, the original theory of ANCs cannot be directly implemented, in particular lacking an algorithm for *term normalisation*, a key part of the theory which allows recognition of valid composites.

A new theory of normalisation is therefore required, one which is well-adapted to the data structures of a potential proof assistant, and with respect to which an recursive algorithm for normalisation can be provided. We develop this new theory here, and describe its role within an implementation, with the goal of making higher category theory more accessible for the working computer scientist.

Diagrams. Associative n-categories have the striking feature of being inherently geometrical, with terms in the theory having a direct geometrical representation. Every term has a dimension, and the terms of dimension n are called *n-diagrams*. A 0-diagram is a point, a 1-diagram is a sequence of points arranged on a line, and a 2-diagram is a combinatorial version of a planar string diagram [22]. In the general case, an n-diagram can be interpreted as a combinatorial "n-dimensional string diagram".

At LICS 2019 a simple inductive term model for these n-diagrams was presented, called *zigzags*, which we make further use of here. An example of a 2-diagram is shown on the left in Figure 1, with its underlying zigzag structure shown on the right, the natural numbers giving the dimension of the component at each point. The zigzag representation makes the combinatorial structure explicit, but we will generally prefer the cleaner visual style of the left image.

Normalisation. Associative n-categories are strictly associative and strictly unital in all dimensions, two attractive properties which remove considerable bureaucracy from proof construction. However, the theory gains these properties in very different ways. The strict associativity is explicit: given composable 1-morphisms f, g, h, the composites[1] $(f \cdot g) \cdot h$ and $f \cdot (g \cdot h)$ are syntactically identical, and can

[1] Here and throughout we use forward composition notation.

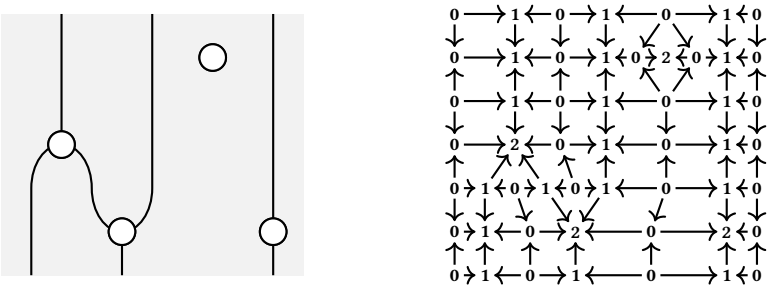
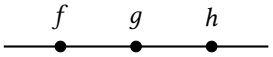


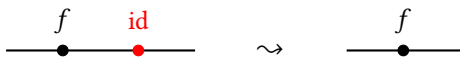
Figure 1. *Representing a 2-dimensional string diagram as an iterated zigzag of natural numbers.*

be drawn as the following 1-diagram:



(1)

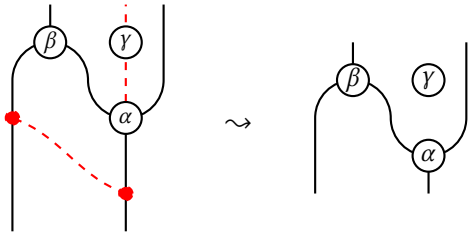
In contrast, the composite $f \cdot \mathrm{id}$ is not syntactically identical to f. Instead, there is a nontrivial *diagram normalisation* process $f \cdot \mathrm{id} \rightsquigarrow f$:



(2)

This normalisation process removes identity structures, yielding a "strictly unital" form for the composite, in this case f itself. To give the user the experience of interacting with a strictly unital theory, the proof assistant performs normalisation silently after every user interaction, ensuring the user sees only the normal form.

In low dimensions, normalisation seems to be a simple process. In dimension 1, a composite is given by a string of tokens, and normalisation simply removes any identity tokens, as shown above in expression (2). In dimension 2, a composite can be understood as a string diagram in a weakly unital monoidal category, where some strands are explicitly labelled by the unit object, indicated here by dotted red wires. In this case the normalisation process removes these unit structures:



The situation is similar in dimension 3; once again, normalisation simply removes all identity structures. These examples quickly give us the impression that all identity structures are *redundant*, since normalisation produces an algebraically simpler version of the diagram that omits them.

However, in dimension 4 and above, more subtle behaviour arises. The geometrical structure of the diagram can cause certain identity structures to become "locked", in a way that prevents them being removed by normalisation. If such an identity were removed, the resulting diagram would be algebraically ill-defined. We call these *essential identities*, to contrast with the redundant identities we visualised above. We will see in Section 5 why these essential identities arise. This makes a normalisation algorithm in the general case non-obvious, since the naive strategy of "removing all identity structures" cannot succeed.

Type Checking. Given an n-diagram, a question of central importance is whether it *type-checks* with respect to some given signature Σ; that is, whether it correctly encodes an n-morphism in the free ANC generated by Σ. The existing theory of ANCs gives a simple answer to this question: break the diagram into atomic "pieces", and for each piece, check that it normalizes to give an element of the signature. Normalisation therefore plays a critical role in this aspect of the theory, and is essential for any implementation.

More details of the type-checking scheme are given in Section 7, where we also illustrate our normalisation algorithm in detail, using two substantial examples of real interest in higher category theory: the 3-dimensional braiding, and the 5-dimensional syllepsis.

Our Contribution. We introduce a new mathematical foundation for term normalisation in associative n-categories, focusing on the categorical properties of *degeneracy maps* in categories of n-diagrams. Defined in terms of a simple categorical lifting property, these degeneracy maps can be interpreted as "injecting identity structure" into a n-diagram; a degeneracy map $f : D \to D'$ therefore serves as a witness that D and D' are similar, except D contains fewer redundant identities. Proposition 4.2 shows that for any n-diagram D, and any pair of degeneracy maps $f : A \to D$ and $g : B \to D$, the pullback $A \times_D B$ exists, with the resulting map $A \times_D B \to D$ again a degeneracy map, which can be considered the "joint resolution" of f, g. In this way, the normalisation of D can be characterized as the joint resolution of all degeneracy maps into D. Since there only finitely many up to isomorphism, this is well-defined.

We then show how normalisation can be computed. Given n-diagrams D and $A_1, \ldots, A_n$, equipped with n-diagram maps $f_i : A_i \to D$, our central observation is that we can give a recursive algorithm for the *relative normalisation* of D with

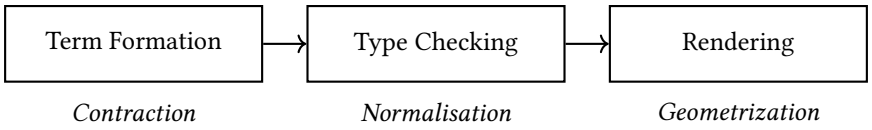
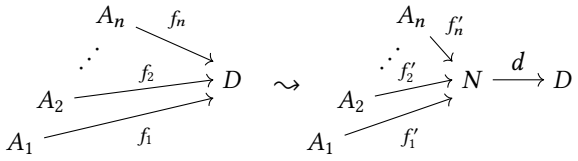


Figure 2. *An simplified overview of the proof assistant's processing pipeline.*

respect to the f_i, written N, as follows:



Here d is a degeneracy map, and we have $f'_i \cdot d = f_i$ for all $i \in I$, where $I = \{1, \ldots, n\}$.[2] Recalling the standard categorical definition of a *sink* as an object equipped with a family of incoming morphisms, the relative normalisation provides a universal factorization of the sink $(D, \{f_i : A_i \to D\}_{i \in I})$ into a composite of the sink $(N, \{f'_i : A_i \to N\}_{i \in I})$ with the morphism d.

We then compute the *absolute normalisation* of a diagram D as the relative normalisation of the sink $(D, \emptyset)$ consisting of D equipped with the empty collection of morphisms.

Implementation. We have implemented our results, and they form a central part of a proof assistant for associative n-categories. Implemented as a client-side web application, it is hosted anonymously for the purposes of this submission at the following URL:

http://95.179.192.207

The tool was launched in January 2019, and has since been loaded 12,000 times by over 4,000 users. It allows direct construction and manipulation of higher-categorical composites by a click-and-drag mechanic. In Section 7 we provide links to proof objects, which illustrate some of our results.

We give a simplified overview of the proof assistant's processing pipeline in Figure 2, which we summarize as follows.

– *Term Formation* is performed primarily via the contraction mechanism, which allows part of an n-diagram to be homotopically reduced. The theoretical foundation for this technique was presented at LICS 2019 [20].

– *Type Checking* verifies that the term generated by the user interaction step is valid. The major component of this type checker is an implementation of the recursive sink normalisation algorithm that we describe in this paper.

– *Rendering* takes place via a geometrization process that extracts a cubical mesh from the term representation, which is then processed and sent to the video card for rendering. This component will be described in a future paper.

[2]We emphasize that the object N and morphism $d : N \to D$ depend on the choice of morphisms f_i, although we suppress this in the notation.

Since this is a theoretical venue we will not describe further details of the implementation.

1.2 Related work

The theory of associative n-categories was originally developed by Dorn, Douglas and Vicary [6], and has been described in the thesis of Dorn [5] in terms of bundles of singular n-cubes. We present a new approach following the zigzag construction of Reutter and Vicary [20], giving us access to a simple inductive structure on terms. The theory of normalisation developed here makes heavy use of categorical lifting properties (cartesian and cocartesian maps), simple categorical "power tools" which allow an efficient formal development. This also allows us to give a recursive algorithm for normalisation, which is not achieved in the singular n-cubes approach.

While we believe our theory is in principle equivalent to that proposed by Dorn, Douglas and Vicary, we make our constructions from first principles, giving a self-contained development. Our approach also has the advantage of allowing a far more concise presentation.

1.3 Acknowledgements

The authors are grateful to Eric Finster, Christoph Dorn and Christopher Douglas for useful discussions.

1.4 Notation

For any $n \geq 0$ we denote by $\underline{n}$ the finite total order $\{0, \ldots, n-1\}$. We write Δ_+ for the category where objects are these finite total orders, and morphisms are order-preserving maps. We also write $\Delta_=$ for the subcategory of non-empty total orders, with maps that preserve the initial and final elements. For an order-preserving map $f : \underline{n} \to \underline{m}$ and some $i \in \underline{m}$, we write $f^{-1}(i)$ for its preimage as a subset of $\underline{n}$. The terminal category is denoted as 1.

2 The zigzag construction

We begin by recalling the theory of categorical zigzags due to Reutter and Vicary [20]. Our presentation is in fact a mild generalization, permitting non-identity boundary maps, as we make clear below. The main object of study is the zigzag, defined as follows.

Definition 2.1. In a category $\mathbf{C}$, a *zigzag* X is a diagram of the following form, for some integer length $n \geq 0$:

$$X(\mathrm{r}0) \longrightarrow X(\mathrm{s}0) \longleftarrow X(\mathrm{r}1) \longrightarrow \cdots \longleftarrow X(\mathrm{r}n)$$

The objects of the form $X(\mathrm{r}\,i)$ are called the *regular objects*, and the objects of the form $X(\mathrm{s}\,i)$ the *singular objects*.

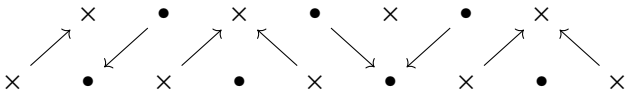


Figure 3. *A monotone map $f : \underline{3} \to \underline{4}$ in Δ_+ going down the page, interleaved with the map $(Rf)^{\mathrm{op}} : \underline{5} \to \underline{4}$ in $\Delta_=$ going up the page. The elements in Δ_+ correspond to the gaps between elements in $\Delta_=$, so each map determines the other.*

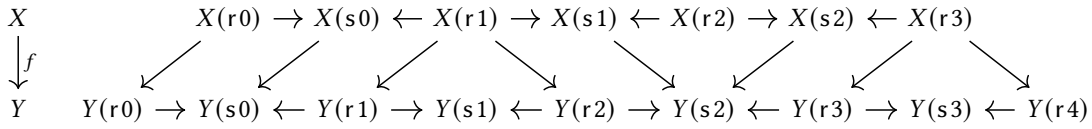


Figure 4. *A map $f : X \to Y$ of zigzags, with singular map $f^{\mathrm{s}} : \underline{3} \to \underline{4}$ shown in Figure 3.*

A zigzag can also be thought of as a sequence of cospans, with adjacent cospans sharing a base object.

To define maps between zigzags, we first require an auxilliary observation. There is an equivalence $R : \Delta_+ \to \Delta_=^{\mathrm{op}}$, originally described by Wraith [26], which sends an object $\underline{n}$ to $\underline{n+1}$, and a map $f : \underline{n} \to \underline{m}$ to the opposite of the map $\underline{m+1} \to \underline{n+1}$ defined as follows:

$$i \mapsto \min(\{j \in \underline{n} \mid f(j) \geq i\} \cup \{n\}).$$

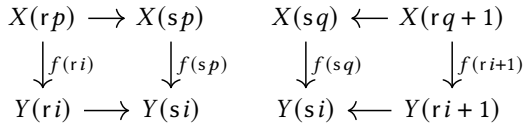
While the formula may appear non-obvious, the idea is straightforward, and we illustrate it in Figure 3, showing how the monotones f and $(Rf)^{\mathrm{op}}$ are interleaved in a simple way.

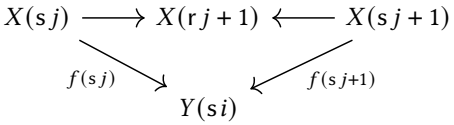
Definition 2.2. In a category $\mathbf{C}$, given zigzags X, Y of length n, m respectively, a *zigzag map* $f : X \to Y$ consists of a *singular map* $f^{\mathrm{s}} : \underline{n} \to \underline{m}$ in Δ_+, along with an implied *regular map* $f^{\mathrm{r}} := (Rf^{\mathrm{s}})^{\mathrm{op}} : \underline{m+1} \to \underline{n+1}$, together with the following additional structure:

1. for every $0 \leq i \leq m$ a map in $\mathbf{C}$:
$$f(\mathrm{r}\,i) : X(\mathrm{r}\,f^{\mathrm{r}}(i)) \to Y(\mathrm{r}\,i)$$
2. for every $0 \leq j < m$ a map in $\mathbf{C}$:
$$f(\mathrm{s}\,j) : X(\mathrm{s}\,j) \to Y(\mathrm{s}\,f^{\mathrm{s}}(j))$$

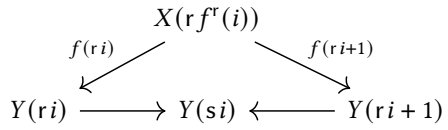
The maps $f(\mathrm{r}\,i)$ are called the *regular slices*, and the maps $f(\mathrm{s}\,j)$ are the *singular slices*. These maps must satisfy the following equations, for every $0 \leq i < m$:

1. If $(f^{\mathrm{s}})^{-1}(i)$ is nonempty, with initial value p and final value q, then the following diagrams must commute, for all j with $p \leq j < q$:





2. If $(f^{\mathrm{s}})^{-1}(i)$ is empty, this diagram must commute:



The notion of zigzag map is geometrically natural, and best understood via example. We illustrate it in Figure 4. The regular slices $f(\mathrm{r}\,i)$ and singular slices $f(\mathrm{s}\,j)$ are drawn vertically, with the zigzag structure of X drawn above, and of Y drawn below. As a result we naturally obtain a categorical diagram comprising 7 squares, and the equational part of the zigzag map definition simply requires these to commute. In this example the singular monotone $f^{\mathrm{s}} : \underline{3} \to \underline{4}$ is defined by $f^{\mathrm{s}}(0) = 1$, $f^{\mathrm{s}}(1) = 2$ and $f^{\mathrm{s}}(2) = 2$, while the regular monotone $f^{\mathrm{r}} : \underline{5} \to \underline{4}$ acts as $f^{\mathrm{r}}(0) = 0$, $f^{\mathrm{r}}(1) = 1$, $f^{\mathrm{r}}(2) = 1$, $f^{\mathrm{r}}(3) = 3$ and $f^{\mathrm{r}}(4) = 3$.

Our approach slightly generalizes the original zigzag definition of Reutter and Vicary [20], where the regular slices of a zigzag map were required to be identities. This extra generality will be critical for our results, since normalisation can change the boundary of a diagram.

Categories of Zigzags. Zigzag maps can be composed in a natural way. Given $f : X \to Y$ and $g : Y \to W$, we define the following:

$$(g \circ f)(\mathrm{s}\,j) = g(\mathrm{s}\,f^{\mathrm{s}}(j)) \circ f(\mathrm{s}\,j)$$
$$(g \circ f)(\mathrm{r}\,i) = g(\mathrm{r}\,i) \circ f(\mathrm{r}\,g^{\mathrm{r}}(i))$$

In terms of the representation used in Figure 4, this corresponds to stacking one diagram above the other. This is easily seen to be associative and unital, and hence for any category $\mathbf{C}$, we obtain a *zigzag category* $Z(\mathbf{C})$ of zigzags and zigzag maps. This construction is functorial in $\mathbf{C}$.

The category $Z(1)$ of zigzags in the terminal category is isomorphic to Δ_+. For every category $\mathbf{C}$ the unique functor $\mathbf{C} \to 1$ thus induces a functor $\pi : Z(\mathbf{C}) \to \Delta_+$, which is natural in $\mathbf{C}$. In the next section we will make heavy use of the theory of cartesian and cocartesian lifts of maps in Δ_+ to

$Z(\mathbf{C})$, as they will allow us to characterise universal zigzag maps of a particular shape.

Since for any category $\mathbf{C}$ the zigzag construction $Z(\mathbf{C})$ is a category itself, the construction can be iterated. We write $Z^n(\mathbf{C})$ for the *n-fold zigzag category* on $\mathbf{C}$.

Diagrams from Zigzags. We define an *n-diagram* to be an object of the category $Z^n(\mathbb{N})$, where $\mathbb{N}$ denotes the poset of natural numbers $0 < 1 < \ldots$. We consider such objects of $Z^n(\mathbb{N})$ as combinatorial encodings of n-dimensional string diagrams. To motivate this definition, we refer back to Figure 1; using the machinery we have developed, we can now see that this represents an object of $Z^2(\mathbb{N})$, giving a combinatorial foundation for the string diagram that appears onthe left-hand side of the figure.

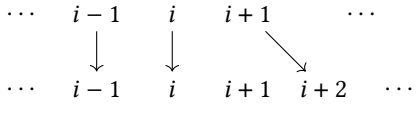
The natural numbers at each point encode the dimension of the algebraic generator that exists at that location in the diagram. In a real string diagram, we would ordinarily give further information, labelling the points with the name of a generator. However, for the purposes of normalisation, only the dimensions of the generators are relevant, and so this simpler notation suffices for our purposes.

To represent a meaningful string diagram, an n-diagram must also satisfy *type-checking* conditions, for which normalisation plays a critical role. We describe this in Section 7.

3 Degeneracy Maps

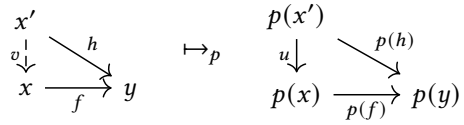
The zigzag construction admits a notion of *degeneracy map* that insert "identity regions" into a diagram. A degeneracy map $X \to Y$ then serves as a witness that the diagrams X and Y are similar, with the only difference being that X contains fewer redundant identities. In this section we define degeneracy maps and study their properties. Significant use is made of cartesian and co-cartesian liftings.

Cartesian Liftings. For some $\underline{n}$ in Δ_+ the ith *face map* is the unique injective map $d_i : \underline{n} \to \underline{n+1}$ that omits $i \in \underline{n+1}$ from its image. We can illustrate this as follows:



We will use these face maps to construct zigzag maps with an important categorical lifting property, as follows.

Definition 3.1. Let $p : \mathbf{C} \to \mathbf{D}$ be a functor. A map $f : x \to y$ in $\mathbf{C}$ is *p-cartesian* if for every map $h : x' \to y$ in $\mathbf{C}$ and $u : p(x') \to p(y)$ such that $p(h) = p(f) \circ u$, there exists a unique $v : x' \to x$ in $\mathbf{C}$ such that $h = f \circ v$ and $u = p(v)$:

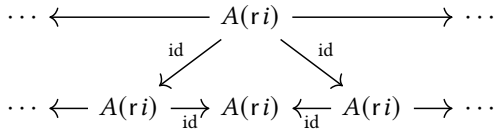


The *p-vertical* maps are those maps $f : x \to y$ such that $p(f) = \mathrm{id}$. A map is *p-cocartesian* if it is p^{op}-cartesian.

This is widely used property in categorical algebra, with a simple intuition: one imagines that f, h are paths in a space, with a projection p to some subspace, such that whenever h factors through f in the projection, the factorisation can be lifted to the original space.

Cartesian maps are a standard concept from the theory of fibred categories [8, 24]. The functors that we consider in this paper will not be categorical fibrations or opfibrations, but nevertheless, certain maps will still satisfy the universal property of cartesian or cocartesian maps, as we now show.

Lemma 3.2. *In a category* $\mathbf{C}$*, for any zigzag* A *of length* n *and* $i \in \underline{n+1}$*, the following zigzag map out of* A *is* π*-cocartesian over the face map* d_i*:*



Here the singular map is the i*th face map, and we insert an itentity cospan in the target, with the slice maps also being identities*

Proof. Let $f : A \to A'$ be a map of this form. Let $h : A \to B$ be a map in $Z(\mathbf{C})$ and $u : \pi(A') \to \pi(B)$ a map in Δ_+ such that $u \circ \pi(f) = \pi(h)$. The shape of the diagram defining a lift $v : A' \to B$ of u is completely determined. Since the slices of f and the cospan inserted in A' are identities, the slices of v are completely determined by the requirement that $v \circ f = h$. □

Generating Degeneracies. In the category Δ_+, the face maps generate all the monomorphisms. Since π-cocartesian maps are unique up to unique vertical isomorphism [24], this lemma therefore implies that any π-cocartesian map over a monomorphism in Δ_+ inserts levels consisting of isomorphisms. We call those maps the *simple degeneracy maps*.

Diagrams of higher dimension admit more ways to insert identities: not only can we insert a 1-dimensional identity slice into a 2-dimensional diagram, but we can degenerate each of the 1-dimensional subslices. We call these zigzag maps f with $\pi(f) = \mathrm{id}$ that have degeneracy maps as slices the *parallel degeneracy maps*. We represent higher-dimensional diagrams by iterating the zigzag construction, so we can define general degeneracy maps recursively, as follows.

Definition 3.3. *Degeneracy maps* in $Z^n(\mathbf{C})$ are generated under composition by the following classes:

1. *simple degeneracy maps*, the π-cocartesian maps over the monomorphisms in Δ_+;
2. *parallel degeneracy maps*, the π-vertical maps in which every slice map is a degeneracy map in $Z^{n-1}(\mathbf{C})$.

To simplify inductive arguments we define degeneracy maps in $Z^0(\mathbf{C})$ to be the isomorphisms.

Lemma 3.4. *Isomorphisms in* $Z^n(\mathbf{C})$ *are degeneracy maps.*

Proof. Let $f : A \to B$ be an isomorphism in $Z^n(\mathbf{C})$. Since π preserves isomorphisms and Δ_+ is skeletal, f is π-vertical. Since the slice maps of an isomorphism in $Z^n(\mathbf{C})$ need to be isomorphisms in $Z^{n-1}(\mathbf{C})$, by induction they are degeneracy maps as well. Hence f is a parallel degeneracy map. □

Lemma 3.5. *Let $f : A \to B$ be a degeneracy map in $Z^n(\mathbf{C})$. Then f factors uniquely (up to isomorphism) into a simple degeneracy map followed by a parallel degeneracy map.*

Proof. Since the maps in Definition 3.3 are sent by π to either a monomorphism or an identity map, we have that $\pi(f)$ is a monomorphism as well. By Lemma 3.2 the map $f_1 : A \to A'$ of the shape $\pi(f)$ which inserts identity levels is π-cocartesian, so there exists a unique π-vertical map $f_2 : A' \to B$ such that $f = f_2 \circ f_1$. Since the slice maps of the maps in Definition 3.3 are either isomorphisms or degeneracy maps in $Z^n(\mathbf{C})$, the slice maps of f must be degeneracy maps. But the slice maps of f_1 are identities, so the slices of f_2 must be degeneracy maps as well. □

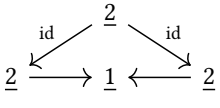
Degeneracy Maps as Subobjects. Any given degeneracy map $f : X \to Y$ can be interpreted as a witness that X as a subobject of Y, which omits some identity regions. This intuition is reflected in the theory as follows.

Lemma 3.6. *Degeneracy maps in $Z^n(\mathbf{C})$ are monomorphisms.*

Proof. Since monomorphisms are closed under composition it suffices to prove the claim for the generating maps of Definition 3.3:

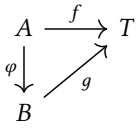
- Let $f : X \to Y$ be a simple degeneracy map, and let $g_1, g_2 : Z \to X$ be maps such that $f \circ g_1 = f \circ g_2$. Since $\pi(f)$ is a monomorphism we have $\pi(g_1) = \pi(g_2)$. The slices of f are isomorphisms, hence g_1 and g_2 must have equal slices, and so $g_1 = g_2$. Thus f is a monomorphism.
- Now let $f : X \to Y$ be a parallel degeneracy map and $g_1, g_2 : Z \to X$ a pair of maps such that $f \circ g_1 = f \circ g_2$. Since $\pi(f) = \mathrm{id}$ we have $\pi(g_1) = \pi(g_2)$. The slices of f are degeneracies, so by induction they are monomorphisms. Hence g_1 and g_2 must have equal slices and so $g_1 = g_2$. Thus f is a monomorphism as well. □

Example 3.7. We note that the converse of the previous lemma does not hold: not every monomorphism is a degeneracy map. The following map of zigzags is a monomorphism in $Z(\Delta_+) = Z^2(1)$, but not a degeneracy map:



Just like monomorphisms, degeneracy maps satisfy the following closure property, which we establish with a simple inductive argument.

Lemma 3.8. *For any commutative triangle in $Z^n(\mathbf{C})$ as follows, if f, g are degeneracy maps, so is φ:*



Proof. For $n = 0$ this follows since isomorphisms satisfy 2-out-of-3. For $n > 0$ we have that $\pi(f)$ and $\pi(g)$ are monomorphisms and hence $\pi(\varphi)$ must be a monomorphism as well. Consider the diagram obtained by gluing together the defining diagrams f, g and φ. Then every slice map of φ is contained in a commutative triangle of the form above, so by induction is a degeneracy map. □

By Lemma 3.6 a degeneracy map $X \to T$ is a monomorphism, and thus represents a subobject of T. Two monomorphisms $f : X \to T$ and $g : X \to T$ represent the same subjobject of T when there exists an isomorphism $\varphi : X \to Y$ such that $g \circ \varphi = f$. Since by Lemma 3.4 all isomorphisms are degeneracy maps and degeneracy maps are closed under composition, if one monomorphism representing some subobject of T is a degeneracy map, then all of them are. This allows us to define the subposet $\mathrm{Deg}(T) \subseteq \mathrm{Sub}(T)$ of degeneracies into T.

Definition 3.9. Let $\mathbf{C}$ be a category and $n \geq 1$. For $T \in Z^n(\mathbf{C})$ let $\mathrm{Deg}(T)$ be the subposet of $\mathrm{Sub}(T)$ consisting of those subobjects of T represented by a degeneracy map into T.

4 Diagram normalisation

The normalisation of an object T of $Z^n(\mathbf{C})$ for some $n \geq 0$ is the smallest element of $\mathrm{Deg}(T)$, if it exists, in which all redundant identities have been removed. In this section we show that normalisations exist, and describe them as the meet of all of the elements of $\mathrm{Deg}(T)$. Meets in $\mathrm{Sub}(T)$ are intersections of subobjects, which are calculated by taking the pullback of representatives.

Lemma 4.1. *A morphism of zigzags $A \to B$ in $Z(\mathbf{C})$ over the ith face map $d_i : \underline{n} \to \underline{n+1}$ in Δ_+ is π-cartesian if and only if the morphisms in Figure 5 indicated by $\cong$ are isomorphisms in $\mathbf{C}$, and the square indicated by $\lrcorner$ is a pullback square in $\mathbf{C}$.*

Proof. Let $f : A \to B$ be a map of this form. Let $h : A' \to B$ be a map of $Z(\mathbf{C})$ and $u : \pi(A') \to \pi(A)$ a map of Δ_+ such that $\pi(f) \circ u = \pi(h)$. We need to construct a lift $v : A' \to A$ of u that satisfies $f \circ v = h$. The slice $v(\mathsf{r}\, i)$ is uniquely determined by the universal property of the pullback square in the defining diagram of f. The other slices of v are determined since the slices of f to the left and right of the pullback square are isomorphisms. The converse follows by essential uniqueness of π-cartesian maps. □

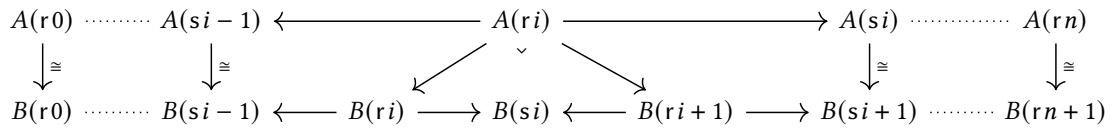


Figure 5. *A π-cartesian map of zigzags over the ith face map $d_i : \underline{n} \to \underline{n+1}$.*

In particular, a square consisting of isomorphisms is a pullback square, so all simple degeneracy maps are π-cartesian maps. Using the machinery of cartesian lifts and the following proposition, we can prove that $\mathrm{Deg}(T)$ is closed under intersections. These intersections are a rare instance of limits which exist in $Z^n(\mathbf{C})$, independently of the existence of limits in $\mathbf{C}$.

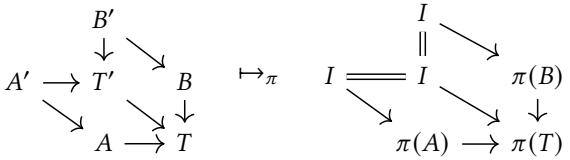
Proposition 4.2. *For any category $\mathbf{C}$, for any two degeneracy maps $f : X \to T$ and $g : Y \to T$ in $Z^n(\mathbf{C})$, their pullback exists, and the projections are also degeneracy maps. In particular $\mathrm{Deg}(T)$ is closed under intersection of subobjects.*

Proof. For $n = 0$, the pullback of a pair of isomorphisms exists and the projections are isomorphisms again. We now proceed to the case $n > 0$ by induction. There is a full embedding of $Z^n(\mathbf{C})$ into $Z([Z^{n-1}(\mathbf{C})^{\mathrm{op}}, \mathrm{Set}])$ induced by the Yoneda embedding. We first calculate the pullback there and afterwards show that it consists of representable objects with degeneracy maps as projectors from the pullback. The claim then follows because full embeddings reflect limits.

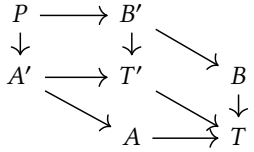
Since f and g are degeneracy maps, the induced maps $\pi(f)$ and $\pi(g)$ are monomorphisms. Pullbacks of monomorphisms exist in Δ_+ and are monomorphisms themselves, so we have a pullback square in which every map is a monomorphism:

$$\begin{array}{ccc} I & \longrightarrow & \pi(B) \\ \downarrow & & \downarrow \\ \pi(A) & \longrightarrow & \pi(T) \end{array}$$

Since the presheaf category $[Z^{n-1}(\mathbf{C})^{\mathrm{op}}, \mathrm{Set}]$ is complete, the pullbacks necessary to apply Lemma 4.1 exist, and so there are π-cartesian lifts of the maps from I as well as unique π-vertical maps between them that make the following diagram commute:



Then the pullback of the cospan $A' \to T' \leftarrow B'$ in the fibre over I is the pullback of the original cospan $A \to T \leftarrow B$:



The fibres of π are diagram categories in which limits are determined pointwise. We thus have that together with Lemma 4.1 the objects of P are limits of diagrams of the following form, where every horizontal map is a degeneracy map, and at least one of the outer columns consists of isomorphisms:

$$\begin{array}{ccccc} A_0 & \longrightarrow & T_0 & \longleftarrow & B_0 \\ \downarrow & & \downarrow & & \downarrow \\ A_1 & \longrightarrow & T_1 & \longleftarrow & B_1 \\ \uparrow & & \uparrow & & \uparrow \\ A_2 & \longrightarrow & T_2 & \longleftarrow & B_2 \end{array}$$

Without loss of generality we can assume that it is the leftmost column and the isomorphisms are identities. By induction, the pullbacks P_0, P_1 and P_2 of the rows are representable, and the projection maps are degeneracy maps, so we get a diagram as follows, in which every row is a degeneracy map:

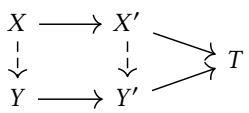
$$\begin{array}{ccc} P_0 & \longrightarrow & A_0 \\ \downarrow & & \| \\ P_1 & \longrightarrow & A_1 \\ \uparrow & & \| \\ P_2 & \longrightarrow & A_2 \end{array}$$

By Lemma 3.8 the vertical maps are degeneracy maps as well and their pullback is representable by induction. Hence every object in the pullback in $Z([Z^{n-1}(\mathbf{C})^{\mathrm{op}}, \mathrm{Set}])$ is representable and the slice maps of the projections are are degeneracy maps as required. □

While this lemma proves that $\mathrm{Deg}(T)$ is closed under binary and hence finite intersections, we need the intersection of all elements in $\mathrm{Deg}(T)$. Since degeneracy maps are uniquely determined up to isomorphism by their action on the shape of a diagram, and any shape only has a finite number of identity regions to be removed, finite intersections will be enough.

Lemma 4.3. *For any two degeneracy maps $f : X \to T$ and $g : Y \to T$ in $Z^n(\mathbf{C})$ that are sent to the same map of untyped diagrams in $Z^n(1)$, there exists an isomorphism $\varphi : X \to Y$ such that $f = g \circ \varphi$. In particular $\mathrm{Deg}(T)$ is finite.*

Proof. For $n = 0$ the degeneracy maps are isomorphisms. For $n \geq 1$ we can factor the degeneracy maps into top-level degeneracy maps followed by a parallel degeneracy map.



We now construct isomorphisms $X' \to Y'$ and $X \to Y$ that fit into this diagram as follows. By induction there exist isomorphisms between the slices of X' and Y' that commute with the cospans to form an isomorphism $X' \to Y'$ since the slice maps of $Y' \to T$ are monomorphisms by Lemma 3.6. Now by Lemma 4.1 the maps $X \to X'$ and $Y \to Y'$ are π-cartesian. Since they are sent to the same map in $Z(1)$ by π, there exists an isomorphism $X \to Y$ that makes the diagram commute. □

We can thus conclude that any object in an iterated zigzag category admits a normalisation.

Proposition 4.4. *For any $T \in Z^n(\mathbf{C})$ the poset $\mathrm{Deg}(T)$ has a smallest element.*

Proof. $\mathrm{Deg}(T)$ is finite by Lemma 4.3, so the binary intersections of Proposition 4.2 suffice to construct the intersection of all elements of $\mathrm{Deg}(T)$, yielding the smallest element. □

Let $T \in Z^n(\mathbf{C})$, and suppose $n : N \to T$ is a degeneracy map representing the smallest element of $\mathrm{Deg}(T)$. Then n is called a *normalising map*, and N is the *normalisation* of T.

5 Computing Normalisation

Essential Identities. In order to be implemented as part of a type checking algorithm, we need a way to compute the normalisation of a diagram. As a first attempt, we might consider a naive recursive algorithm, where we normalise all the singular and regular objects of a zigzag, levelwise.

However, this cannot work. To see why, consider the diagram in Figure 6, in which the top zigzag T and bottom zigzag B are normalised, but the middle zigzag M is not. If we normalised the middle row, we would obtain a new zigzag M' with length 0, and then the updated zigzag map $q' : T \to M'$ from the top row to the middle row would require a singular map of type of $\underline{2} \to \underline{0}$ in Δ_+. But there are no such functions, as $\underline{0}$ is empty.

As a result, the identity cospan contained in M is not redundant, but *essential* for the geometry of the entire structure, and cannot be removed. The entire structure shown in Figure 6 is therefore *already normalised*, despite the existence of the identity cospan in the middle row.

What makes the normalisation algorithm nontrivial is that it must correctly detect these essential identities, leaving them in place, while removing the redundant identities. Note that the original definition of normalisation via Proposition 4.4 handles this subtlety automatically in some sense, since an essential identity cannot be factorized out. But for our normalisation algorithm, we must handle it explicitly.

Sink Normalisation. To solve this problem, instead of normalising each part of a diagram in isolation, we keep track of that part's "environment", in the form of a *sink* of incoming maps $(T, \{f_i : A_i \to T\}_{i \in I})$, where I is an indexing set, which we will often omit when it is clear from context.

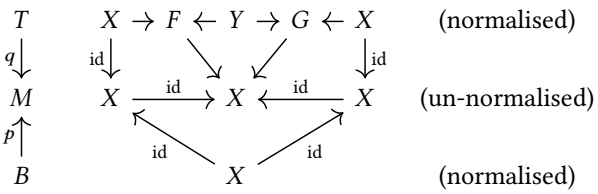


Figure 6. *A normalised diagram with an un-normalised level.*

In the case of Figure 6, we would ask for a normalisation of the middle zigzag M in the context of the sink

$$(M, \{p : B \to M,\ q : T \to M\})$$

with two incoming zigzag maps.

Proposition 4.4 showed that for any $T \in Z^n(\mathbf{C})$, the poset $\mathrm{Deg}(T)$ has a smallest element $N \to T$, which we defined as the normalisation of T. In the following, we prove a relative version of this proposition, yielding a notion of normalisation *relative* to a sink. Given a sink $\mathcal{S} = (T, \{f_i : A_i \to T\})$, let $\mathrm{Deg}_{\mathcal{S}}(T)$ denote a subposet of $\mathrm{Deg}(T)$, containing those degeneracy maps $n : N \hookrightarrow T$ through which the sink $\mathcal{S}$ factors, i.e. such that there exists $(N, \{g_i : A_i \to N\})$ such that $f_i = n \circ g_i$. Intuitively, the idea is that N is a subobject of T that arises by discarding only those redundant identities which are *not* in the image of an element of the sink f_i.

Proposition 5.1. *Let $\mathcal{S} = (T, \{f_i : A_i \to T\})$ be a sink of maps in $Z^n(\mathbf{C})$. Then the subposet $\mathrm{Deg}_{\mathcal{S}}(T)$ of the finite poset $\mathrm{Deg}(T)$ is non-empty and closed under intersection, and therefore also has a smallest element.*

Proof. The identity $\mathrm{id} : T \to T$ is a degeneracy map through which every sink factors, and hence is an element of $\mathrm{Deg}_{\mathcal{S}}(T)$. The intersection of subobjects is the pullback of representatives, which exists for degeneracy maps due to Proposition 4.2. The factorisations through the intersections are then given by the universal property of the pullback. By the same argument as in the proof of Proposition 4.4, it follows that $\mathrm{Deg}_{\mathcal{S}}(T)$ has a smallest element. □

We call the initial element of $\mathrm{Deg}_{\mathcal{S}}(T)$ the *relative normalisation* of T with respect to the sink $\mathcal{S}$. If $\mathcal{S}$ is the empty sink, we have $\mathrm{Deg}_{\mathcal{S}}(T) = \mathrm{Deg}(T)$, and hence the relative normalisation of T with respect to the empty sink $(T, \emptyset)$ agrees with the normalisation of T. In this way, we see that full normalisation is a special case of relative normalisation. The reason to study relative normalisation is that it admits a recursive algorithm, as follows.

The Normalisation Algorithm. Given a sink in $Z^n(\mathbf{C})$, the following recursive algorithm computes its relative normalisation. We give a step-by-step illustration in Example 5.3.

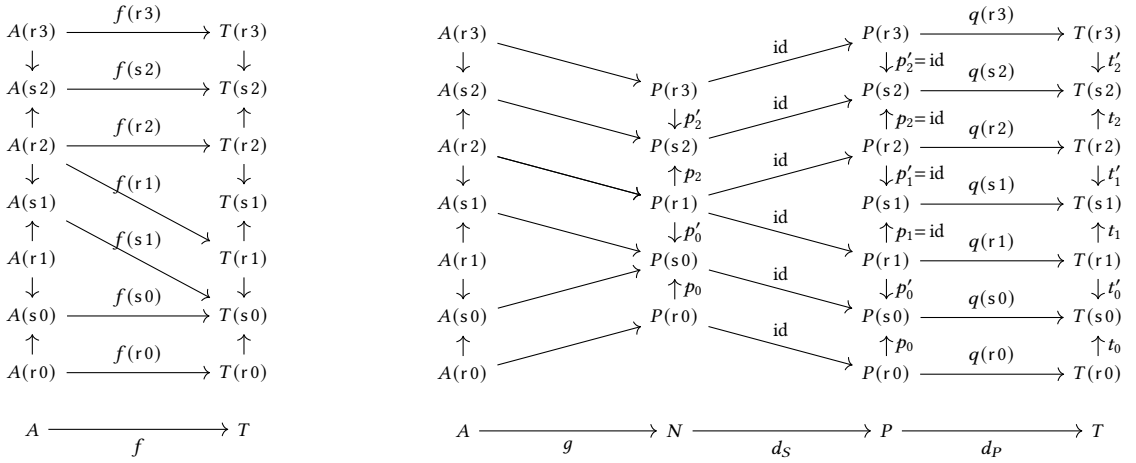


Figure 7. *An illustration of the normalisation algorithm.*

Construction 5.2 (Normalisation). Given a sink of maps $\mathcal{S} = (T, \{f_i : A_i \to T\}_{i \in I})$ in $Z^n(\mathbf{C})$ we define, by induction on $n \geq 0$, a degeneracy map $d : N \to T$ and factorisations $A_i \to N \to X$ of each map f_i. For $n = 0$, we set $d = \mathrm{id}$. For $n > 0$, recall from Lemma 3.5 that any degeneracy map $d : N \to T$ factors uniquely as a simple degeneracy map $d_S : N \to P$ followed by a parallel degeneracy map $d_P : P \to N$. Given the sink $\mathcal{S}$, we construct these maps d_S and d_P and the factorisations

$$A_i \to N \overset{d_S}{\to} P \overset{d_P}{\to} T$$

of $f_i : A_i \to T$ via the following scheme.

1. Let $\mathrm{r}h$ be a regular height of T. Consider the sink

$$(T(\mathrm{r}h), \{f_i(\mathrm{r}h) : A_i(\mathrm{r}f_i^{\mathrm{r}}(h)) \to T(\mathrm{r}h)\})$$

in $Z^{n-1}(\mathbf{C})$ consisting of the component of the zigzag maps $f_i : A_i \to T$ at the regular height $\mathrm{r}h$ (see Definition 2.2). Recursively apply relative normalisation to this sink to obtain an object $P(\mathrm{r}h)$ of $Z^{n-1}(\mathbf{C})$, a degeneracy map $P(\mathrm{r}h) \to T(\mathrm{r}h)$, and for every $i \in I$ a factorisation as follows:

$$\begin{array}{ccc} A_i(\mathrm{r}f_i^{\mathrm{r}}(h)) & & \\ \downarrow & \searrow & \\ P(\mathrm{r}h) & \dashrightarrow & T(\mathrm{r}h) \end{array} \tag{3}$$

2. For every singular height $\mathrm{s}h$ of T, consider the sink in $Z^{n-1}(\mathbf{C})$ which consists of the maps $f_i(\mathrm{s}t) : A_i(\mathrm{s}t) \to T(\mathrm{s}h)$ for every $i \in I$ and every $t \in (f_i^{\mathrm{s}})^{-1}(h)$, as well as the composite $P(\mathrm{r}h) \to T(\mathrm{r}h) \to T(\mathrm{s}h)$ and the composite $P(\mathrm{r}h+1) \to T(\mathrm{r}h+1) \to T(\mathrm{s}h)$. Recursively apply relative normalisation to this sink to obtain an object $P(\mathrm{s}h)$ in $Z^{n-1}(\mathbf{C})$, a degeneracy map $P(\mathrm{s}h) \to T(\mathrm{s}h)$, and the following factorisations, for every $i \in I$ and $t \in (f_i^{\mathrm{s}})^{-1}(h)$:

$$\begin{array}{ccc} A_i(\mathrm{s}t) & & \\ \downarrow & \overset{f_i(\mathrm{s}t)}{\searrow} & \\ P(\mathrm{s}h) & \dashrightarrow & T(\mathrm{s}h) \end{array} \tag{4}$$

$$\begin{array}{ccc} P(\mathrm{r}h) & \longrightarrow & T(\mathrm{r}h) \\ \downarrow & & \downarrow \\ P(\mathrm{s}h) & \dashrightarrow & T(\mathrm{s}h) \\ \uparrow & & \uparrow \\ P(\mathrm{r}h+1) & \longrightarrow & T(\mathrm{r}h+1). \end{array} \tag{5}$$

3. The factorisations in (5) assemble the objects $P(\mathrm{r}h)$ and $P(\mathrm{s}h)$ into a zigzag in $Z^{n-1}(\mathbf{C})$, and hence into an object P of $Z^n(\mathbf{C})$. The degeneracy maps $P(\mathrm{r}h) \to T(\mathrm{r}h)$ and $P(\mathrm{s}h) \to T(\mathrm{s}h)$ assemble into a parallel degeneracy map $d_P : P \to T$ in $Z^n(\mathbf{C})$. The factorisations of (3) and (4) assemble into factorisations in $Z^n(\mathbf{C})$:

$$\begin{array}{ccc} A_i & & \\ \downarrow & \overset{f_i}{\searrow} & \\ P & \longrightarrow & T \end{array}$$

Since the degeneracy map $P \to T$ is parallel, the maps $A_i \to P$ and $A_i \to T$ will have equal singular maps.

4. For those cospans $P(\mathrm{r}h) \to P(\mathrm{s}h) \leftarrow P(\mathrm{r}h+1)$ with both legs given by isomorphisms, and for which the singular object $P(\mathrm{s}h)$ is not in the image of any of the $A_i \to P$, remove them from the zigzag $P \in Z^n(\mathbf{C})$. This results in a smaller zigzag N, and a simple degeneracy map $d_S : N \to P$ which re-inserts these trivial cospans. The maps $A_i \to P$ then canonically factor through this map, since by construction the removed heights are not in their image:

$$\begin{array}{ccc} A_i & & \\ \downarrow & \searrow & \\ N & \longrightarrow & P \end{array}$$

5. Define $d : N \to T$ to be the composite $d = d_P \circ d_S$.

This concludes the description of the algorithm.

Example 5.3. We illustrate the algorithm in Figure 7, normalising a 1-element sink $(T, \{f : A \to T\})$. On the left of the figure we show the structure of A, T and f, while on the right of the figure we show the intermediate construction P, and the eventual normal form N.

– In step 1, we recursively apply relative normalisation to the 1-element sink $(T(\mathrm{r}0), \{f(\mathrm{r}0) : A(\mathrm{r}0) \to T(\mathrm{r}0)\})$ to obtain the factorization

$$A(\mathrm{r}0) \to P(\mathrm{r}0) \overset{q(\mathrm{r}0)}{\to} T(\mathrm{r}0)$$

and similarly for $f(\mathrm{r}1)$, $f(\mathrm{r}2)$, $f(\mathrm{r}3)$. This gives us the regular objects of P, and the regular slices $q(\mathrm{r}i) : P(\mathrm{r}i) \to T(\mathrm{r}i)$.

– In step 2, we build the singular levels $P(\mathrm{s}i)$, by recursively factorizing the sinks into $T(\mathrm{s}i)$. For example, for $i = 0$, we must factorize the following sink:

$$\Big(R(\mathrm{s}0), \big\{\, t_0 \circ q(\mathrm{r}0) : P(\mathrm{r}0) \to T(\mathrm{s}0), t_0' \circ q(\mathrm{r}1) : P(\mathrm{r}1) \to T(\mathrm{s}0), \\ f(\mathrm{s}0) : A(\mathrm{s}0) \to T(\mathrm{s}0), f(\mathrm{s}1) : A(\mathrm{s}1) \to T(\mathrm{s}0) \,\big\}\Big)$$

Factorizing this sink recursively yields the singular object $P(\mathrm{s}0)$ and the degeneracy maps $p_0 : P(\mathrm{r}0) \to P(\mathrm{s}0)$ and $p_0' : P(\mathrm{r}1) \to P(\mathrm{s}0)$, as well as factorizing maps $A(\mathrm{s}0) \to P(\mathrm{s}0)$ and $A(\mathrm{s}0) \to P(\mathrm{s}0)$.

– In step 3, we assemble this data into the zigzag P, and the zigzag map $g : A \to N$, as shown in the figure.

– In step 4, we inspect the maps p_i, p_i' to find identity zigzags. We suppose for the sake of example that $p_1 = \mathrm{id}$, $p_1' = \mathrm{id}$, $p_2 = \mathrm{id}$ and $p_2' = \mathrm{id}$. Since $T(\mathrm{s}1)$ is not in the image of $A \to T$, also $P(\mathrm{s}1)$ is not in the image of $A \to P$ (since those zigzag maps have equal singular maps), and we can therefore omit the entire p_1, p_1' cospan, and we proceed to construct N appropriately. Note that we retain the cospan p_2, p_2' in N, even though the legs are identities, since $P(\mathrm{s}2)$ *is* in the image of $A \to P$. The zigzag map $d_S : N \to P$ is then constructed as a simple degeneracy map, with face map omitting level 1.

– In step 5, we produce the entire normalising degeneracy map as the composite $d_P \circ d_S : N \to T$.

We are now done, and have factorized the original sink into the compsite of a degeneracy map $d_p \circ d_s$, and a new sink $(N, \{g : A \to N\})$.

Correctness. We now show that Construction 5.2 correctly produces the relative normalisation of a sink.

Proposition 5.4. *Let $\mathcal{S} = (T, \{f_i : A_i \to T\}_i)$ be a sink in $Z^n(\mathbf{C})$. The map $d : N \to T$ constructed in Construction 5.2 is the relative normalisation of T with respect to $\mathcal{S}$, i.e. the smallest element of $\mathrm{Deg}_{\mathcal{S}}(T)$. In particular, applied to the empty sink $\mathcal{S} = (T, \emptyset)$, the morphism $d : N \to T$ produced by Construction 5.2 is the normalisation of T.*

Proof. Recall that in Construction 5.2, the degeneracy map $d : N \to T$ is constructed as a composite $d_P \circ d_S$, where d_P and d_S are parallel and simple degeneracy maps respectively. We will prove the following three statements:

1. d_P is the initial parallel degeneracy map into T through which the sink $\mathcal{S}$ factors.
2. d_S is the initial simple degeneracy map into P such that the sink $\mathcal{S}$ factors through $d_P \circ d_S$.
3. $d_P \circ d_S$ is the initial degeneracy map into T through which $\mathcal{S}$ factors.

Assume the inductive hypothesis that the claim holds in $Z^k(\mathbf{C})$ for $k < n$. We then proceed as follows.

1. Consider any parallel degeneracy map $d : P' \to P$, such that $\mathcal{S}$ factors through $d_P \circ d'$. Any regular slice map of $d_P \circ d'$ satisfies the factorisation condition (3). Since Construction 5.2 has chosen the initial regular slice map for d_P satisfying the conditions, the regular slices of d' must be isomorphisms. But then the singular slice maps of $d_P \circ d'$ satisfy the factorisation conditions of (4) and (5). Similarly, it follows that the singular slices of d' must also be isomorphisms. So d' is an isomorphism.
2. The top-level degeneracy map $N \to P$ is chosen in Construction 5.2 to normalise as many trivial levels of P as possible while retaining compatibility.
3. Let $d' : N' \to T$ be any other degeneracy map via which $\mathcal{S}$ factors. By Lemma 3.5, d' decomposes into a simple degeneracy map $d_S' : N' \to P'$ followed by a parallel degeneracy map $d_P' : P' \to T$. By part 1 there is a parallel degeneracy map $P \to P'$ which fits into this diagram:

$$\begin{array}{ccc} P & \xrightarrow{d_P} & T \\ \downarrow & & \| \\ P' & \xrightarrow[d_P']{} & T \end{array}$$

By Proposition 4.2 the pullback of $d_S' : N' \to P'$ along $P \to P'$ exists and is a simple degeneracy map. But then by part 2 there exists a map $N \to N' \times_{P'} P$ which makes the following diagram commute:

$$\begin{array}{ccccc} N & & & & \\ \downarrow & \searrow^{d_S} & & & \\ N' \times_{P'} P & \longrightarrow & P & \xrightarrow{d_P} & T \\ \downarrow & & \downarrow & & \| \\ N' & \xrightarrow[d_S']{} & P' & \xrightarrow[d_P']{} & T \end{array}$$

So $d_P \circ d_S$ represents a smaller subobject of T. The claim follows since d' was chosen arbitrarily among the compatible degeneracy maps. □

Reflective Localisation. This relative sink normalisation may be considered a special case of the following general machinery. Given an object T in a category $\mathbf{A}$ and a class of morphisms $\mathfrak{D}$ in $\mathbf{A}$, let $\mathbf{A}/T$ denote the over-category, and consider the inclusion of the full subcategory $\mathbf{A}/^{\mathfrak{D}}T \hookrightarrow \mathbf{A}/T$ of those $d : A \to T$ which are in $\mathfrak{D}$. This inclusion has a left-adjoint $L : \mathbf{A}/T \to \mathbf{A}/^{\mathfrak{D}}T$ if and only if, for every morphism $f : A \to T$ in $\mathbf{A}$, the evident category of factorisations of f into a morphism in $\mathbf{A}$ followed by a morphism in $\mathfrak{D}$ has an initial object. The image $L(f) \in \mathbf{A}/^{\mathfrak{D}}T$ is the $\mathfrak{D}$-morphism part of this initial factorisation of f.

An analogous observation applies to the full inclusion $\mathbf{A}/^{\mathfrak{D}}T \hookrightarrow \mathrm{Sink}_{\mathbf{A}}(T)$, where $\mathrm{Sink}_{\mathbf{A}}(T)$ is an appropriate category of sinks $\mathcal{S} = (T, \{f_i : A_i \to T\})$ in $\mathbf{A}$ into T. This inclusion has a left adjoint if and only if, for every sink $\mathcal{S}$, the associated category of factorisations of $\mathcal{S}$ into a sink $(N, \{g_i : A_i \to N\})$ followed by a morphism $N \to T$ in $\mathfrak{D}$ has an initial object.

Applied to the situation where $\mathbf{A} = Z^n(\mathbf{C})$ and $\mathfrak{D}$ is the class of degeneracy maps, Proposition 5.1 may therefore be understood as asserting that for every object $T \in Z^n(\mathbf{C})$, the inclusion

$$R : Z^n(\mathbf{C})/^{\deg}T \to \mathrm{Sink}_{Z^n(\mathbf{C})}(T)$$

has a left adjoint L. The relative normalisation of a sink $(T, \{f_i : A_i \to T\})$ is then constructed as its image under L.

Following standard terminology [17, § IV.3], this says that $Z^n(\mathbf{C})/^{\deg}T$ is a *reflective subcategory* of $\mathrm{Sink}_{Z^n(\mathbf{C})}(T)$, and relative normalisation is the corresponding *reflective localisation* functor $\mathrm{Sink}_{Z^n(\mathbf{C})}(T) \to Z^n(\mathbf{C})/^{\deg}T$.

6 Globularity

Associative n-categories form a globular theory of higher categories, meaning that for any diagram, the boundary of the source matches the boundary of the target. This is enforced in the proof assistant by requiring that diagrams have a globularity property, meaning intuitively that regular slices have to act trivially. We define this formally as follows.

Definition 6.1. In $Z^n(\mathbf{C})$, a map f is a *globular map* if $n = 0$, or both the following properties hold:

1. all regular slice maps of f are isomorphisms;
2. all singular slice maps of f are globular in $Z^{n-1}(\mathbf{C})$.

An object of $Z^n(\mathbf{C})$ is a *globular object* if $n = 0$, or it is a zigzag of globular objects and globular maps in $Z^{n-1}(\mathbf{C})$.

To be valid in the proof assistant, a diagram must be globular, and its normalisation must also be globular. It is therefore a requirement that normalization preserves globularity, and we verify this here.

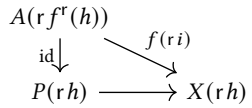
The core of the argument is that the normalisation algorithm maintains the invariant that all maps in the sinks of the recursive applications already normalise the regular levels, so the factorisations can be globular. We define this property formally as follows.

Definition 6.2. In $Z^n(\mathbf{C})$, a map is *regularly normalising* if $n = 0$, or both the following properties hold:

1. all regular slice maps are normalising;
2. all singular slice maps are regularly normalising.

Lemma 6.3. *Let $d : N \to X$ be the relative normalisation of a globular object $X \in Z^n(\mathbf{C})$ with respect to a sink $(X, \{f_i : A_i \to X\}_i)$ of regularly normalising maps. Then N is a globular object, the factorisations of the maps in the sink are globular maps and d is regularly normalising.*

Proof. By Proposition 5.4 the normalisation algorithm correctly computes the relative normalisation. Since the maps f_i are regularly normalising, the regular slices of A_i are already normalised and so the algorithm fills the diagram (3) as follows:



In particular, the regular slices of the computed factorisations $A_i \to P$ are identities and the regular slices of the degeneracy map d_P are normalising.

Since X is globular so are its singular slices. Since the sink consists of regularly normalising maps, the solid maps arising from (4) are regularly normalising maps into globular objects. The maps $P(\mathrm{r}h) \to X(\mathrm{s}h)$ and $P(\mathrm{r}h{+}1) \to X(\mathrm{s}h)$ in (5) are composites of a normalising map followed by a globular one, so they are certainly regularly normalising. Therefore the sinks formed in (4) and (5) satisfy the conditions of this Lemma. By induction the singular slices of d_P are regularly normalising, the singular slices of factorisations $A_i \to P$ are globular, and the cospans $P(\mathrm{r}h) \to P(\mathrm{s}h) \leftarrow P(\mathrm{r}h + 1)$ are globular maps between globular objects.

By the observations above, the parallel degeneracy map d_P is regularly normalising, P is a globular object and the factorisations $A_i \to P$ are globular maps. These properties are preserved by the final step which precomposes d_P by the simple degeneracy map d_S. □

Proposition 6.4. *The normalisation of a globular object is globular.*

Proof. The normalisation of some globular object X is the relative normalisation of X with respect to the empty sink. Thus the result follows by Lemma 6.3. □

The invariants of Lemma 6.3 can also be of help in the implementation of the normalisation algorithm. The regular slices of the degeneracy maps are determined by their targets and thus do not need to be represented explicitly. All diagrams are globular, as well as all the factorisation maps, allowing them to be represented by simpler data structures for which globularity is hard-coded. The non-globular sink maps can be represented as formal composites of a degeneracy map followed by a globular map.

7 Examples

In this section we sketch the type checking scheme, and show some worked examples of interest in higher category theory, the *Eckmann-Hilton Move* and the *Syllepsis*.

Type Checking. We first give an informal overview of type checking, focusing on its relevance for normalisation.

For an n-diagram D given as an object of $Z^n(\mathbb{N})$, we define its *singular content* as a 1-element set if $n = 0$, or else by recursion as the disjoint union of the singular content of its

singular objects. For example, the 2-diagram of Figure 1 has singular content of cardinality 10. We then break D into a number of *pieces*, one for each element of singular content, by taking the preimages of the elements of singular content under the singular map structures defining D.

The type checking procedure works with respect to a *signature* of allowed algebraic generators. Given a globular n-diagram, we normalize each piece, and then check if the resulting n-diagram is an element of the signature. If this is the case for all pieces, the diagram is declared valid.

Examples. Here we illustrate the type checking procedure for two examples. Although we label points of diagrams in this section with generator names, for the purpose of normalisation we implicitly use the generator dimensions to obtain an $\mathbb{N}$-labelling, as on the right of Figure 1.

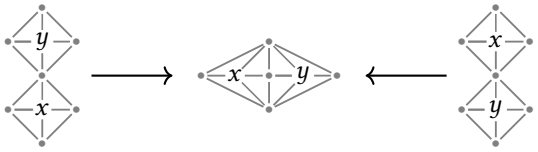
Each example is accompanied by a hyperlink to the type-checked formalisation in the proof assistant, which will display a 3d model. Left-click and drag to rotate; right-click and drag to pan; use the mouse wheel to zoom. A video is also provided for each example, showing how it is constructed.

Example 7.1.

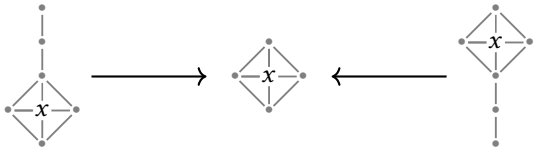
Proof: http://95.179.192.207/eh_proof
Video: http://95.179.192.207/eh_video

The *Eckmann-Hilton Move* is a 3-morphism in a finitely presented 3-category, generated by a single 0-cell $\bullet$, and 2-cells $x, y : \mathrm{id}(\bullet) \to \mathrm{id}(\bullet)$. The signature therefore comprises the following nontrivial diagrams:

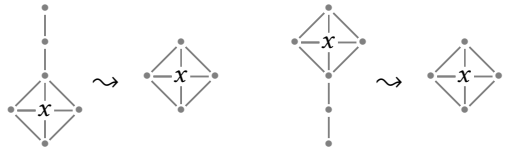
The Eckmann-Hilton Move itself is represented by the following 3-diagram, interpreted as x "braiding" around y:



This has singular content $\{x, y\}$. The piece containing singular content x is the following 3-diagram, which we name D:



To normalise this 3-diagram piece we apply our normalisation algorithm, as presented in Construction 5.2. Step 1 invokes recursive calls which normalise the left and right boundaries of D, with the following results:



In Steps 2 and 3, we use these results to obtain the intermediate normalisation zigzag P, a 3-diagram of length 1:

We note that this is an identity cospan, and so in Step 4 of the algorithm we omit this cospan when we form N:

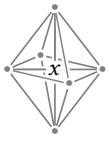
This is the normal form of our original piece D. This is an element of our signature, hence the piece D is validated by the type checker. The piece corresponding to y is also valid, and so the entire Eckmann–Hilton 3-diagram type checks.

Example 7.2.

Proof: http://95.179.192.207/syll_proof
Video: http://95.179.192.207/syll_video

The *Syllepsis* is a 5-morphism in a finitely presented 5-category, generated by a single 0-cell $\bullet$, and two 3-cells with types $x, y : \mathrm{id}(\mathrm{id}(\bullet)) \to \mathrm{id}(\mathrm{id}(\bullet))$. The signature therefore contains these 3-diagrams, which we draw in a quasi-3d style:

We depict the Syllepsis 5-diagram in Figure 8. Intuitively, it represents the equivalence between the braid and its inverse when immersed in 4-dimensional space. In the live proof, use the "Slice" control on the right to navigate through this equivalence. It has singular content $\{x, y\}$, and we extract the piece containing singular content x, depicting it in Figure 9. Applying our normalisation algorithm, following a long sequence of recursive calls, we obtain the normal form:



Since this is an element of our signature, we determine that the piece is valid. Similarly, the piece corresponding to singular content y is valid, and hence the entire Syllepsis 5-diagram type checks.

References

[1] Michael Atiyah. 2009. Topological quantum field theories. In *The geometry and physics of knots*. CUP, 12–23.

[2] Steve Awodey and Michael A. Warren. 2008. Homotopy theoretic models of identity types. *Mathematical Proceedings of the Cambridge Philosophical Society* 146, 01 (2008), 45.

[3] John Baez and James Dolan. 1995. Higher-dimensional algebra and topological quantum field theory. *JMP* 36, 11 (1995), 6073. arXiv:q-alg/9503002

[4] Roberto Bruni, José Meseguer, and Ugo Montanari. 2002. Symmetric monoidal and cartesian double categories as a semantic framework for tile logic. *MSCS* 12, 01 (2002).

[5] Christoph Dorn. 2018. Associative n-categories. (2018). Ph.D. dissertation. arXiv:1812.10586.

[6] Christoph Dorn, Christopher Douglas, and Jamie Vicary. 2022. The theory of associative n-categories. (2022).

[7] Eric Goubault. 2003. Some geometric perspectives in concurrency theory. *Homology, Homotopy and Applications* 5, 2 (2003), 95–136.

[8] Alexander Grothendieck. 1971. *Revêtements étales et groupe fondamental (SGA 1)*. Lecture notes in mathematics, Vol. 224. Springer-Verlag.

[9] Alexander Grothendieck. 1983. Pursuing Stacks. (1983). Available online.

[10] Yves Guiraud and Philippe Malbos. 2012. Higher-dimensional normalisation strategies for acyclicity. *Adv. Math.* 231, 3-4 (2012), 2294–2351.

[11] Martin Hofmann and Thomas Streicher. 1994. The groupoid model refutes uniqueness of identity proofs. In *LICS 1994*. IEEE.

[12] Arthur Jaffe, Zhengwei Liu, and Alex Wozniakowski. 2016. Holographic Software for Quantum Networks. (2016). arXiv:1605.00127

[13] Yves Lafont. 1997. Two-dimensional rewriting. In *Rewriting Techniques and Applications*. 228–229.

[14] Tom Leinster. 2004. *Higher Operads, Higher Categories*. CUP.

[15] Jacob Lurie. 2008. On the Classification of Topological Field Theories. *Current Developments in Mathematics* 2008, 1 (2008), 129–280.

[16] Jacob Lurie. 2009. *Higher Topos Theory (AM-170)*.

[17] Saunders MacLane. 1971. *Categories for the working mathematician*. Springer-Verlag, New York-Berlin. ix+262 pages.

[18] Samuel Mimram. 2014. Towards 3-Dimensional Rewriting Theory. *LMCS* 10, 2 (2014).

[19] David Reutter and Jamie Vicary. 2016. Biunitary constructions in quantum information. (2016). arXiv:1609.07775.

[20] David Reutter and Jamie Vicary. 2019. High-level methods for homotopy construction in associative n-categories. In *LICS 2019*. IEEE.

[21] Christopher Schommer-Pries. 2009. *The Classification of Two-Dimensional Extended Topological Field Theories*. Ph. D. Dissertation. University of California, Berkeley. arXiv:1112.1000

[22] Peter Selinger. 2009. A Survey of Graphical Languages for Monoidal Categories. *New Structures for Physics* (2009). arXiv:0908.3347

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

[24] Angelo Vistoli. 2007. Notes on Grothendieck topologies, fibered categories and descent theory. arXiv:math/0412512 [math.AG]

[25] Vladimir Voevodsky. 2006. A very short note on the homotopy λ-calculus. (2006). Available online.

[26] Gavin Wraith. 1993. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* 34, 4 (1993), 259–266.

A Syllepsis Figures

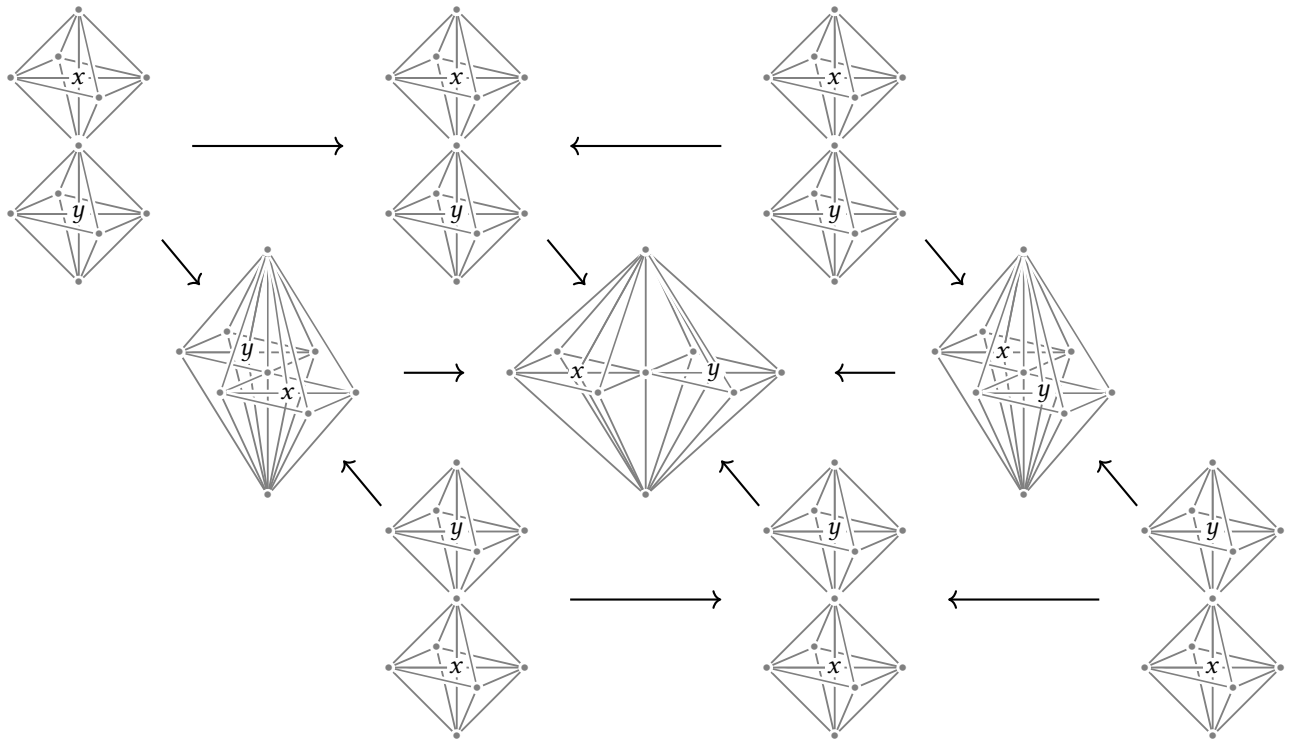


Figure 8. *The zigzag structure of the syllepsis as a 5-diagram.*

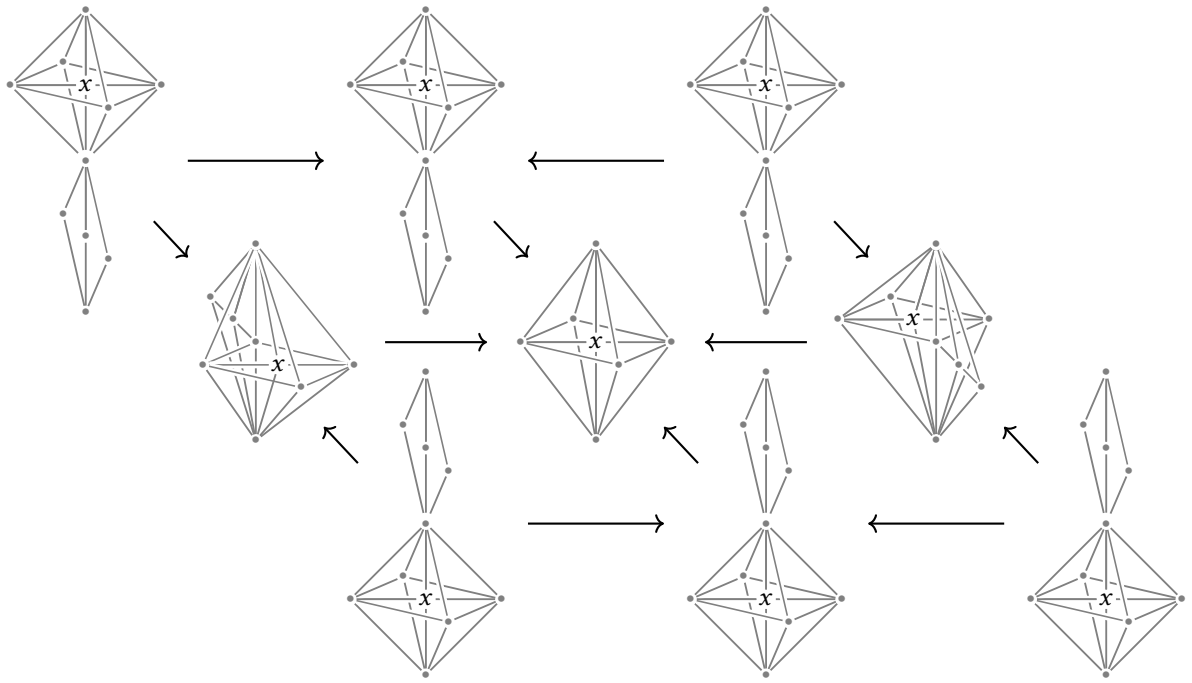


Figure 9. *The singular piece containing the generator x in the zigzag structure of the syllepsis.*