Invertible cells in ω-categories

Thibaut Benjamin* Ioannis Markakis†

May 2, 2024

Abstract

We study coinductive invertibility of cells in weak ω-categories. We use the inductive presentation of weak ω-categories via an adjunction with the categroy of computads, and show that invertible cells are closed under all operations of ω-categories. Moreover, we give a simple criterion for invertibility in computads, together with an algorithm computing the data witnessing the invertibility, including the inverse, and the cancellation data.

1 Introduction

Higher category theory is an emergent field with several newfound applications in computer science and mathematics. In particular, globular higher groupoids have been used to describe the structure of identity types in Homotopy Type Theory [1, 23, 30], establishing a connection with topology, via the Homotopy Hypothesis. The latter due to Grothendieck [16] states that weak ω-groupoids are equivalent to topological spaces up to weak homotopy equivalence, and is an active topic of research with recent progress [18]. Beyond Homotopy Type theory, globular higher categories have been investigated in connection with higher dimensional rewriting [26] and topological quantum field theory [3], and in homology [19].

Higher categories can be described starting from different shapes, and using several variations on their axioms. They can have various level of strictness, and can be truncated. Surveys on different definitions of higher categories have been published by Cheng and Lauda [12], and Leinster [22]. In this article, we focus on globular weak (∞, ∞)-categories, henceforth called ω-categories. Those were originally introduced by Batanin [4], and then Leinster [21] as algebras for the monad T defined as the initial contractible operad. Then, Matsiniotis [24] proposed another definition in terms of models of a globular theory, adapting an idea due to Grothendieck [16] for ω-groupoids. Those two definition have been proven equivalent by Ara [2] and Bourke [10]. More recently, Finster and

*University of Cambridge, `tjb201@cam.ac.uk`
†University of Cambridge, `ioannis.markakis@cl.cam.ac.uk`

Mimram [14] have proposed a more syntactic definition, seeing ω-categories as the models of a dependent type theory called catt. Benjamin, Finster and Mimram [6] have proven this definition to be equivalent to that of Grothendieck and Maltsiniotis. Inspired by this type theory, Dean et al. [13] have proposed an equivalent description of the monad T of Batanin and Leinster. They have constructed a category Comp whose objects are called computads, together with an adjunction

$$\mathsf{Free} : \mathsf{Glob} \rightleftarrows \mathsf{Comp} : \mathsf{Cell}$$

between globular sets and computads, whose induced monad is the monad T of Leinster. A direct comparison between the type theory catt and the computads introduced by Dean et al. has been established by Benjamin, Markakis and Sarti [8].

Understanding the homotopy theory of ω-categories is one of the major open problems of higher category. The homotopy theory of strict ω-categories has been studied by Lafont et al. [20], motivated by its strong connection with homology and its application to rewriting theory, continuing the work of Squier [28]. In particular, they have defined a model structure in which computads are the cofibrant objects, and for which the description of the weak equivalences relies heavily on the notion of weakly invertible cells. It is conjectured that a similar model structure should exist for weak ω-groupoids and weak ω-categories. It was proven by Henry that defining this model structure on ω-groupoids is the missing piece to prove the homotopy hyptohesis [17]. Partial results in this direction have been worked out, mainly by Henry and Lanary [18], who have proven the homotopy hypothesis in dimension 3. On weak ω-categories, a weak factorization corresponding to the putative cofibrations and trivial fibrations is known. Dean et al. [13] and Markakis [25] have proven that the computads defined by Dean et al. are exactly the cofibrant objects for this conjectured model structure.

In this article, we investigate weakly invertible cellls in ω-categories, a notion used in the characterisation of the weak equivalences for the conjectured model structure. Such cells have been defined coinductively for a broad class of globular higher structures by Cheng [11], and studied in the case of strict ω-categories by Lafont et al. [20]. More recently, Rice [27] has compared this coinductive notion of invertibility to other proposed notions of invertibility.

Fujii, Hoshino and Maehara [15] have also studied coinductively invertible cells in ω-categories, and shown that they are closed under all operations. Nonetheless, our work differs significantly from theirs in various aspects. We use the inductive presentation of Leinster's monad by Dean et al [13], which provides us with an explicit syntax to work with cells of ω-categories. This allows us to give a more precise version of their main theorem, together with an elementary proof of it. Our work further provides a syntactic criterion deciding the invertibility of a cell in a finite dimensional computad, as well as a structurally recursive algorithm computing the inverse of a cell and the cancellation data attached to it.

The complexity of out work originates from the complexity of weak ω-categories. To tackle this complexity, we expand the syntax provided by the inductive presentation of the free ω-category monad with reusable meta-operations that produce new operations from existing ones, continuing our previous work [7]. More precisely, we will use the *opposite* and *suspension* operations introduced there, which amount to interpreting an operation as an operation of the opposite or hom ω-category respectively. We also introduce two new operations, the *functorialisation*,which has only been studied using the type theory catt [5], and the *chain reduction* operation. The former relies on the idea that each operation of ω-categories is functorial, and hence can be applied to higher dimensional cells as well. The latter, replaces a composition operation with an equivalent more biased one over a simpler pasting diagram. Finally, to tackle the combinatorial complexity of computing the invertibility data for a composite of invertible cells, we further need to introduce some ad-hoc operations, specific to the problem. Those operations allow us to cancel the composition of a sequence of cells with their inverses. Due to the shape of the computad they live over, we call those operations *telescopes*.

Using those operations, we give an elementary proof that any composite of invertible cells in an ω-category is again invertible. We believe that the simplicity of this proof is strong evidence that the syntactic approach to ω-categories is promising, and can help further develop the theory of ω-categories. Proofs built using this syntactic presentation can often lead to algorithms, or meta-operations expanding the language itslef. For example, given a cell in a computad satisfying our invertibility criterion, our proof gives a recipe to construct its inverse and the invertible cancellation witnesses. This proceedure has been implemented as an extension of the proof assistant catt[1], based on the dependent type theory with the same name, dedicated to working in the language of ω-categories. With this new feature, a user can input a term corresponding to an invertible cell, and the proof assistant automatically computes the term corresponding to the chosen inverse, or the term corresponding to any of its invertibility data, more generally.

Overview of the paper

In Section 2, we recall the notion of globular pasting diagram and define operations on them making them into a free strict ω-category. Section 3 then recalls the definition of computads and the free ω-category monad on globular sets given by Dean et al. [13]. Section 4 is dedicated to defining several constructions in ω-categories that allow us to expand the language in which we work. In Section 5, we show our main theorem stating that a composite of invertible cells is invertible. Finally, Section 6 presents and evaluates the implementation of our main result in the proof assistant catt.

Acknowledgements

The authors would like to thank Prof. Jamie Vicary for his support during this project. Furthermore, the second author would like to acknowledge funding from the Onassis foundation - Scholarship ID: F ZQ 039-1/2020-2021.

[1] `http://www.github.com/thibautbenjamin/catt`

2 Globular pasting diagrams

In this background section, we will recall globular sets and globular pasting diagrams, the underlying shapes and the arities of the operations of ω-categories respectively. Globular pasting diagrams are a family of globular sets that has been studied extensively under different presentations, namely globular cardinals [29], globular sums [2], or pasting diagams [21]. For a survey of those presentations and their equivalence, we refer to the work of Weber [31, Section 4]. Here we will expand on the presentation of Dean et al [13] by also describing the composition operations of trees in their setting.

2.1 Globular sets

The category $\mathbb{G}$ of globes has objects the natural numbers, and morphisms generated by the *cosource* and *cotarget* $s, t \colon n \to (n+1)$ under the *coglobularity* relations:

$$s \circ s = t \circ s \qquad\qquad s \circ t = t \circ t.$$

The category Glob of globular sets is the category of presheaves on $\mathbb{G}$. More explicetely, a globular set $X \colon \mathbb{G}^{\mathsf{op}} \to \mathsf{Set}$ consists of a set X_n for every $n \in \mathbb{N}$ together with *source* and *target* functions $\mathsf{src}, \mathsf{tgt} \colon X_n \to X_{n+1}$ satisfying the *globularity* relations:

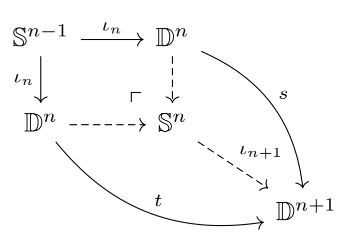
$$\mathsf{src} \circ \mathsf{src} = \mathsf{src} \circ \mathsf{tgt} \qquad\qquad \mathsf{tgt} \circ \mathsf{src} = \mathsf{tgt} \circ \mathsf{tgt}\,.$$

We will call the elements of X_n the n-cells of X. We also define the *m-source* and *m-target* of an n-cell $x \in X_n$ for $m < n$ by iterating the source and target functions:

$$\mathsf{src}_k\, x = = \mathsf{src}(\cdots(\mathsf{src}\, x)) \qquad\qquad \mathsf{tgt}_k\, x = = \mathsf{tgt}(\cdots(\mathsf{tgt}\, x)).$$

We will say that a pair of n-cells are *parallel* when they have the same source and target, where by convention, all 0-cells are parallel.

The *n-disk* $\mathbb{D}^n$ for $n \in \mathbb{N}$ is the representable globular set $\mathbb{G}(-, n)$. The *n-sphere* $\mathbb{S}^n$ for $n \geq -1$ is defined recursively with an inclusion $\iota_n \colon \mathbb{S}^{n-1} \to \mathbb{D}^n$ via the following pushout diagram



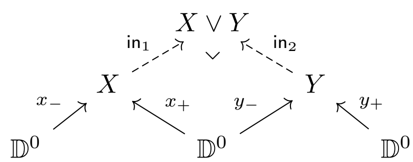
starting from $\mathbb{S}^{-1} = \emptyset$ being the initial globular set. By the Yoneda lemma, morphisms $\mathbb{D}^n \to X$ are in natural bijection to n-cells of X, while morphisms

$\mathbb{S}^n \to X$ are in natural bijection to pairs of parallel n-cells of X. Under those bijections, composition with the inclusion ι_n sends a cell to its source and target, which are parallel by the globularity relations.

Globular pasting diagrams are a family of globular sets defined recursively, using the suspension and the wedge sum of globular sets. The *suspension* of a globular set X is the globular set ΣX with cells given by

$$(\Sigma X)_0 = \{v_-, v_+\} \qquad\qquad (\Sigma X)_{n+1} = X_n.$$

Its source and target functions are given by those of X, with v_- being the source and v_+ being the target of every 1-cell. The *wedge sum* $X \vee Y$ of a pair of globular sets X and Y with respect to chosen 0-cells $x_-, x_+ \in X_0$ and $y_-, y_+ \in Y_0$ is defined to be the following pushout in Glob:



The wedge sum defines a monoidal product in the category of globular sets with two chosen 0-cells with unit the 0-disk, being the composition of cospans of globular sets.

2.2 Batanin trees and their positions

Batanin was the first to observe that globular pasting diagrams are indexed by isomorphisms classes of rooted planar trees and to give a combinatorial description of them [4]. As explained by Leinster [21], there exists one such tree of dimension 0, while trees of dimension at most $n+1$ are precisely list of trees of dimension at most n. This leads to the following *inductive* definition of rooted planar trees, which we will call *Batanin trees*.

Definition 1. The set of Batanin trees is inductively defined by one rule: there exists a Batanin tree $\mathsf{br}\, L$ for every list L of Batanin trees.

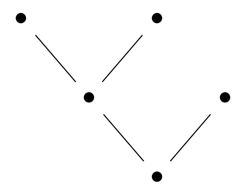
The rule specifies that the set Bat of Batanin trees is equipped with a function $\mathsf{br} \colon \mathsf{List}(\mathsf{Bat}) \to \mathsf{Bat}$ where List is the free monoid endofunctor

$$\mathsf{List}\, X = \coprod_{n \in \mathbb{N}} X^n.$$

Being inductively generated by this rule means precsely that the pair $(\mathsf{Bat}, \mathsf{br})$ is the initial algebra for the endofunctor List. In particular, there exists a tree $\mathsf{br}[]$ corresponding to the empty list, and using this tree, we can define more complicated trees, such as the tree

$$B = \mathsf{br}[\mathsf{br}[\mathsf{br}[], \mathsf{br}[]], \mathsf{br}[]].$$

We visualise those trees by letting $\mathsf{br}\,L$ be the tree with a new root and with branches given by L. For example, $\mathsf{br}[]$ is the tree with one vertex and no branches, while the tree B above can be visualised as follows:



The *dimension* of a Batanin tree is the height of the corresponding planar tree, or equivalently the maximum of the dimension of its positions, defined below. It can be computed recursively by

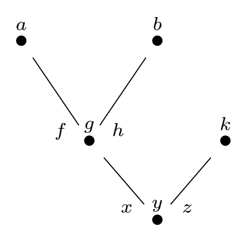
$$\mathsf{dim}(\mathsf{br}[B_1, \dots, B_n]) = \mathsf{max}(\mathsf{dim}\,B_1 + 1, \dots, \mathsf{dim}\,B_n + 1).$$

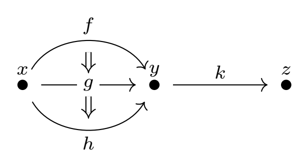
In particular, $\mathsf{br}[]$ is the unique Batanin tree of dimension 0.

Definition 2. The globular set of *positions* of a Batanin tree B is the globular set $\mathsf{Pos}(B)$ defined inductively by the formula

$$\mathsf{Pos}(\mathsf{br}[B_1 \dots, B_n]) = \bigvee_{i=1}^{n} \Sigma\,\mathsf{Pos}(B_i).$$

This is the globular pasting diagram corresponding to the planar tree B as explained by Leinster [21, Appendix F.2]. The positions of a Batanin tree correspond to sectors of the corresponding planar tree [9]. For example, the globular set of positions of the tree B above is the following one





Here the positions f, g, h, a, b are the positions of the left branch $\mathsf{br}[\mathsf{br}[], \mathsf{br}[]]$, while k is the position of theright branch $\mathsf{br}[]$. The 0-positions x, y, z are the new cells created by the suspension operation.

Definition 3. A position $p \in \mathsf{Pos}_k(B)$ of a Batanin tree B will be called *locally maximal* when it is not the source, nor the target of another position. It will be called *maximal* when $k = \mathsf{dim}\,B$.

One can show by induction that the set $\mathsf{Pos}_k(B)$ of k-positions of a Batanin tree B is empty if and only if $k > \mathsf{dim}\,B$, so maximal positions are locally maximal.

Moreover, the locally maximal positions of a tree can be defined recursively by letting the unique position of $\mathsf{br}[]$ be locally maximal, and letting the position $\mathsf{in}_j(p)$ of $\mathsf{br}[B_1, \ldots, B_n]$ be locally maximal when p is locally maximal in B_j, where in_j is the inclusion of the j-th summand in the wedge sum

$$\mathsf{in}_j : \Sigma\,\mathsf{Pos}(B_j) \to \bigvee_{i=1}^{n} \Sigma\,\mathsf{Pos}(B_i).$$

Example 4. The *suspension* of a Batanin tree B is the tree $\Sigma B = \mathsf{br}[B]$. By construction, $\Sigma\,\mathsf{Pos}(B) = \mathsf{Pos}(\Sigma B)$, so globular pasting diagrams are preserved by the suspension operation.

Example 5. Representable globular sets are globular pasting diagrams. More specifically, we can define recursively on $n \in \mathbb{N}$ a tree D_n together with an isomorphism $\mathsf{Pos}(D_n) \cong \mathbb{D}^n$. We start by letting $D_0 = \mathsf{br}[]$ and the isomorphism being the identity of $\mathbb{D}^0$, and then proceed to define $D_{n+1} = \Sigma D_n$ and the isomorphism to be the composite

$$\mathsf{Pos}(D_{n+1}) = \Sigma\,\mathsf{Pos}(D_n) \cong \Sigma\,\mathbb{D}^n \cong \mathbb{D}^{n+1}$$

where the last isomorphism sends the unique top-dimensional cell of $\mathbb{D}^n$ to the unique top-dimensional cell of $\mathbb{D}^{n+1}$.

2.3 Operations on pasting diagrams

Globular pasting diagrams famillialy represent the free strict ω-category monad. In this section, we will describe the free strict ω-category on a globular set, and discuss briefly the monad mutliplication.

Definition 6. The *k-boundary* of a Batanin tree B for $k \in \mathbb{N}$ is the Batanin tree defined recursively by the following formulae

$$\partial_0\,\mathsf{br}[B_1, \ldots, B_n] = \mathsf{br}[]$$
$$\partial_{k+1}\,\mathsf{br}[B_1, \ldots, B_n] = \mathsf{br}[\partial_k B_1, \ldots, \partial_k B_n]$$

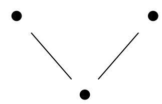
On the level of rooted, planar trees, one can show inductively that the tree $\partial_k B$ is obtained from B by removing all nodes of distance at least k from the root. In terms of pasting diagrams, it is obtained by removing all positions of dimension above k and identifying parallel k-positions. It follows that the positions of the k-boundary can be included back into the positions of the original tree in two ways by picking the left-most and right-most position for every branch. More formally, the *k-cosource* and *k-cotarget*

$$s_k^B, t_k^B : \mathsf{Pos}(\partial_k B) \to \mathsf{Pos}(B)$$

are defined recursively for a tree $B = \mathsf{br}[B_1, \ldots, B_n]$ by the following formulae:

$$s_0^B = \mathsf{in}_1(v_-) \qquad\qquad t_0^B = \mathsf{in}_n(v_+)$$
$$s_{k+1}^B = \bigvee_{i=1}^{n} \Sigma\, s_k^{B_i} \qquad\qquad t_{k+1}^B = \bigvee_{i=1}^{n} \Sigma\, t_k^{B_i}.$$

As the name suggests, the k-cosource and k-cotarget satisfy the coglobularity relations. Considering our example Batanin tree B, we have that

$\partial_1 B =$  $\qquad \mathsf{Pos}(\partial_1 B) = \overset{x}{\bullet} \xrightarrow{f} \overset{y}{\bullet} \xrightarrow{k} \overset{z}{\bullet}$

The source inclusion s_1^B is the one given by the names of the positions, while the target inclusion t_1^B is the one sending f to h and the other positions to the ones with the same name.

Remark 7. To simplify the notation, we will denote by ∂B the boundary $\partial_{\mathsf{dim}\, B-1} B$, and we will denote the corresponding source and target inclusions by s^B and t^B respectively.

Definition 8. The *k-composition* of a pair of Batanin trees B and B' sharing a common k-boundary is the Batanin tree $B *_k B'$ defined recursively by the formulae

$$\mathsf{br}[B_1, \ldots, B_n] *_0 \mathsf{br}[B'_1, \ldots, B'_m] = \mathsf{br}[B_1, \ldots, B_n, B'_1, \ldots, B'_m]$$
$$\mathsf{br}[B_1, \ldots, B_n] *_{k+1} \mathsf{br}[B'_1, \ldots, B'_n] = \mathsf{br}[B_1 *_k B'_1, \ldots, B_n *_k B'_n]$$

We observe first that those equations completely determine the composition of Batanin trees, since for a pair of Batanin trees $\mathsf{br}\, L$ and $\mathsf{br}\, L'$ to share a common positive-dimensional boundary, the lists L and L' must have the same lenght. The composition of a pair of Batanin trees along a common k-boundary amounts to appending the branches of the two trees at every node of distance k from the root. On the level of pasting diagrams, it realises the gluing of the corresponding pasting diagrams along their common boundary, as will be shown in Proposition 9. For example, consider the Batanin trees

$$B = \mathsf{br}[\mathsf{br}[], \mathsf{br}[\mathsf{br}[{\color{red}\mathsf{br}[]}], \mathsf{br}[{\color{red}\mathsf{br}[], \mathsf{br}[]}]]]$$
$$B' = \mathsf{br}[\mathsf{br}[], \mathsf{br}[\mathsf{br}[{\color{blue}\mathsf{br}[], \mathsf{br}[\mathsf{br}[]]}], \mathsf{br}[]]]$$

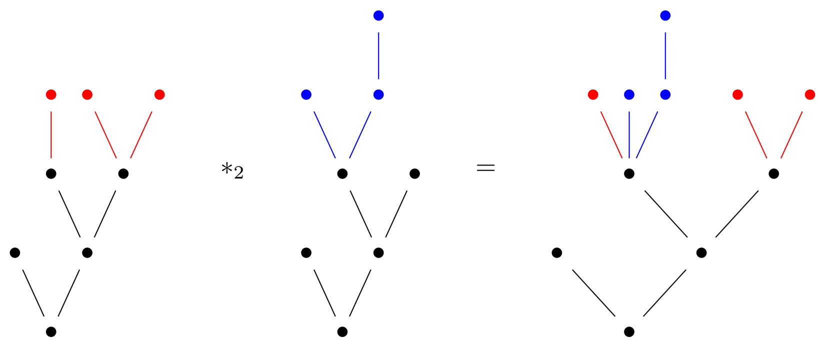
THey share a common 2-boundary, which is the tree

$$\partial_2 B = \partial_2 B' = \mathsf{br}[\mathsf{br}[], \mathsf{br}[\mathsf{br}[], \mathsf{br}[]]]$$

and theie 2-composition is the following tree

$$B *_2 B' = \mathsf{br}[\mathsf{br}[], \mathsf{br}[\mathsf{br}[{\color{red}\mathsf{br}[]}, {\color{blue}\mathsf{br}[], \mathsf{br}[\mathsf{br}[]]}], \mathsf{br}[{\color{red}\mathsf{br}[], \mathsf{br}[]}]]].$$

This process may be visualised as follows:



Proposition 9. *For every $k \in \mathbb{N}$ and every pair of Batanin trees B and B' sharing a common k-boundary, there exists a pushout square of the form*

$$\begin{array}{ccccc} \mathsf{Pos}(\partial_k B) & = & \mathsf{Pos}(\partial_k B') & \xrightarrow{s_k^{B'}} & \mathsf{Pos}(B') \\ {\scriptstyle t_k^B}\downarrow & & & & \downarrow{\scriptstyle \mathsf{in}^+_{k,B,B'}} \\ \mathsf{Pos}(B) & & \xrightarrow{\mathsf{in}^-_{k,B,B'}} & & \mathsf{Pos}(B *_k B') \end{array}$$

Proof. We will construct this pushout diagrams by induction on the Batanin trees $B = \mathsf{br}[B_1, \ldots, B_n]$ and $B' = \mathsf{br}[B'_1, \ldots, B'_m]$ and on $k \in \mathbb{N}$. We define first $\mathsf{in}^-_{0,B,B'}$ be the morphism induced by the inclusions $\mathsf{in}_1, \ldots, \mathsf{in}_n$ of the components of the wedge sum, and $\mathsf{in}^+_{0,B,B'}$ be the morphism induced by the inclusions $\mathsf{in}_{n+1}, \ldots, \mathsf{in}_{n+m}$. The resulting square is a pushout by the definition and associativity of the wedge sum operation. We then define recursively

$$\mathsf{in}^\pm_{k+1,B,B'} = \bigvee_{i=1}^{n} \Sigma\, \mathsf{in}^\pm_{k,B_i,B'_i}.$$

The resulting square is a pushout, since both the wedge sum and the suspension operations preserve connected colimits; the former because it factors as a left adjoint $\mathsf{Glob} \to \mathsf{Glob}_{**}$ [7, Section 2] followed by the forgetful functor from bipointed globular sets to globular sets, and the latter by distributivity of colimits over colimits. □

Definition 10. The *free strict ω-category* on a globular set X is the strict ω-category $F^{\mathsf{str}}X$ consists of the globular set

$$(F^{\mathsf{str}}X)_n = \coprod_{\dim B \le n} \mathsf{Glob}(\mathsf{Pos}(B), n)$$

with source and target functions given by

$$\mathsf{src}_k(B, f) = (\partial_k B, f \circ s_k^B) \qquad \mathsf{tgt}_k(B, f) = (\partial_k B, f \circ t_k^B),$$

with identity operations given by the obvious subset inclusions, and with composition operations given by

$$(B, f) *_k (B', f') = (B *_k B', \langle f, f' \rangle).$$

where $\langle f, f' \rangle$ is the morphism out of the pushout of Proposition 9.

It is out of the scope of this paper to show that $F^{\mathsf{str}}X$ is a strict ω-category. Nonetheless, we will use freely that the operations $*_k$ are associative and unital, and that they satisfy the interchange law, since those propoerties have been shown for example by Leinster [21, Appendix F.2]. An alternative way to prove those axioms is to restate them as equations between the cosources, cotargets, and the pushout inclusions, and show them by induction on the trees involved.

As the name suggests, the functor F^{str} is left adjoint to the underlying globular set functor U^{str} forgetting the identity and compoosition operations of a strict ω-category. We will denote the induced monad by $T^{\mathsf{str}} \colon \mathsf{Glob} \to \mathsf{Glob}$. As an endofunctor, T^{str} is famillially represented by the collection of Batanin trees, so it is cartesian, finitary and coproduct preserving [21, Theorem F.2.2]. The unit of the monad $\eta^{\mathsf{str}} \colon \mathsf{id} \to T^{\mathsf{str}}$ is the natural transformation, whose component at a globular set X, sends an n-cell $x \in X_n$ to morphism $\mathsf{Pos}(D_n) \to \mathbb{D}^n \to X$, where the first morphism is the isomoprhism of Example 5 and the second is obtained by the Yoneda lemma. The monad multiplication is described by the following proposition of Weber [31, Proposition 4.7].

To state the proposition, we observe that the free strict ω-category on the terminal globular set $T^{\mathsf{str}}1$ has as n-cells Batanin trees of dimension at most n, and it has the k-boundary as its k-source and k-target function, so we will denote it by Bat when no confusion may arise. The assignment of the globular set of positions of a Batanin tree can be seen as a functor

$$\mathsf{Pos} \colon \textstyle\int \mathsf{Bat} \to \mathsf{Glob}\,.$$

from the category of elements of Bat sending an object (n, B) to the globular set $\mathsf{Pos}(B)$, and the generating morphisms $s, t \colon (n, \partial_n B) \to (n+1, B)$ to s_n^B and t_n^B respectively.

Proposition 11. *Let $f \colon \mathsf{Pos}(B) \to \mathsf{Bat}$ a morphism of globular sets. Then there exists a Batanin tree $\mu_1^{\mathsf{str}}(f)$ of dimension at most $\dim B$ such that*

$$\mathsf{Pos}(\mu_1^{\mathsf{str}}(B, f)) = \operatorname*{colim}_{(k,p) \in \int \mathsf{Pos}(B)} \mathsf{Pos}(f_k(p))$$

we will denote the canonical cocone by $j^f \colon \mathsf{Pos} \circ \int f \Rightarrow \mathsf{Pos}(\mu_1^{\mathsf{str}}(B, f))$. Moreover, this construction commutes with cosource and cotargets.

A cell of $x \in T^{\mathsf{str}}T^{\mathsf{str}}X$ is a pair (B, f) where $f \colon \mathsf{Pos}(B) \to T^{\mathsf{str}}X$. Writting $f(p) = (f^1(p), f^2(p))$ where $f^1(p)$ is a Batanin tree and $f^2(p) \colon \mathsf{Pos}(f^1(p)) \to X$, we see that f is a morphism of globular sets if and only if f^1 is a morphism and f^2 is a cocone $\mathsf{Pos} \circ \int f^1 \Rightarrow X$. The monad multiplication $\mu_X^{\mathsf{str}} \colon T^{\mathsf{str}}T^{\mathsf{str}}X \to T^{\mathsf{str}}X$ is then given by

$$\mu_X^{\mathsf{str}}(B, f) = (\mu_1^{\mathsf{str}}(B, f^1), \langle f^2 \rangle)$$

where $\langle f^2 \rangle$ is the morphism out of the colimit induced by the cocone f^2. The last part of Proposition 11 ensures that each μ_X^{str} is a morphism of globular sets.

3 Weak ω-categories

As seen from Definition 10, strict ω-categories are globular sets equipped with a unique way to compose diagrams of cells indexed by a globular pasting diagram. Weak ω-categories are a generalisation, in which such diagrams admit a unique composite up to an invertible higher cell. Following Leinster [21], we define weak ω-categories as algebras for certain monad on globular sets, the initial contractible globular globular operad. We will use the description of this monad given by Dean et al. [13], in terms of an adjunction

$$\mathsf{Free} \colon \mathsf{Glob} \rightleftarrows \mathsf{Comp} \colon \mathsf{Cell}$$

with the category Comp of computads. In this section, we recall this description of the ω-category monad, and certain operations on ω-categoriesa and their computads that were introduced in our previous work [7].

3.1 Computads

Computads are generating data for ω-categories, and they are defined by induction on the dimension $n \in \mathbb{N}$. More precisely, we define mutually inductively the category Comp_n of n-computads together with four functors

$$\mathsf{Free}_n \colon \mathsf{Glob} \to \mathsf{Comp}_n \qquad \mathsf{Cell}_n \colon \mathsf{Comp}_n \to \mathsf{Set}$$
$$u_n \colon \mathsf{Comp}_n \to \mathsf{Comp}_{n-1} \qquad \mathsf{Sphere}_n \colon \mathsf{Comp}_n \to \mathsf{Set}$$

three natural transformations

$$\mathsf{bdry}_n \colon \mathsf{Cell}_n \Rightarrow \mathsf{Sphere}_{n-1}\, u_n \qquad \mathsf{pr}_1, \mathsf{pr}_2 \colon \mathsf{Sphere}_n \Rightarrow \mathsf{Cell}_n$$

and an auxilliary subset $\mathsf{Full}_n(B) \subseteq \mathsf{Sphere}_n\, \mathsf{Free}_n(\mathsf{Pos}(B))$ of spheres for every Batanin tree B that we will call full. The functor u_n forgets the top-dimensional generators of a computad. The functor Free_n views a globular set as a computad whose generators are its cells. The functor Cell_n returns the n-cells of the ω-category generated by the computad, while the functor Sphere_n returns pairs of parallel n-cells. The projections pr_i pick the first and second cell of a parallel pair, while the natural transformation bdry_n assigns to each cell its source and target. For the base case, we let Comp_{-1} be the terminal category and Sphere_{-1} the functor picking the terminal set.

An n-computad is a triple $(C_{n-1}, V_n^C, \phi_n^C)$ consisting of an $(n-1)$-computad, a set of n-generators and an attaching function $V_n^C \to \mathsf{Sphere}_{n-1}(C_{n-1})$, assigning to each generator a source and target. A morphism of computads $f \colon C \to D$ is a morphism between the ω-categories generated by the computads, and it is defined to be a pair $(f_{n-1}.f_V)$ of a morphism $C_{n-1} \to D_{n-1}$ together with a function $V_n^C \to \mathsf{Cell}_n(D)$ preserving the source and target, in that the following diagram commutes

$$\begin{array}{ccc} V_n^C & \xrightarrow{\quad f_V \quad} & \mathsf{Cell}_n(D) \\ {\scriptstyle \phi_n^C}\downarrow & & \downarrow{\scriptstyle \mathsf{bdry}_{n,D}} \\ \mathsf{Sphere}_{n-1}(C_{n-1}) & \xrightarrow{\mathsf{Sphere}_{n-1}(f_{n-1})} & \mathsf{Sphere}_{n-1}(D_{n-1}) \end{array}$$

The truncation functor is the first projection on both objects and morphsisms.

The set $\mathsf{Cell}_n\, C$ is defined inductively by two rules. The first states that every generator $v \in V_n^C$ gives rise to a cell that we denote by $\mathsf{var}\, c$. The second rule states that there exists a cell $\mathsf{coh}(B, A, f)$ for every Batanin tree B of dimension at most n, every full $A \in \mathsf{Full}_{n-1}(B)$ and every morphism $f\colon \mathsf{Free}_n\, \mathsf{Pos}(B) \to C$. The boundaries of those cells are defined recursively by

$$\mathsf{bdry}_n(\mathsf{var}\, v) = \phi_n^C(v) \qquad \mathsf{bdry}_n(\mathsf{coh}(B, A, f)) = \mathsf{Sphere}_{n-1}(f_{n-1})(A)$$

Notice that the definition of morphisms uses cells and vice versa. This apparent circularity is resolved using induction-recursion, i.e. reading the definitions above as the description the initial algebra for some polynomial endofunctor [13, Section 3.3].

Composition with a morphism $f\colon C \to D$ and its action on cells are defined mutually recursively by

$$\begin{aligned} \mathsf{Cell}(f)(\mathsf{var}\, v) &= f_V(v) \\ \mathsf{Cell}(f)(\mathsf{coh}(B, A, g)) &= \mathsf{coh}(B, A, f \circ g) \\ f \circ g &= (f_{n-1} \circ g_{n-1}, \mathsf{Cell}_{n-1}(f_{n-1}) \circ g_{n-1}) \end{aligned}$$

together with a proof that bdry_n is natural. Using mutual induction, we can then show that this composition operation is associative and unital, and that Cell_n is a functor.

The free functor Free_n sends a globular set X to the n-computad consisting of $\mathsf{Free}_{n-1}\, X$, the set X_n and the attaching function given by

$$\phi_n^{\mathsf{Free}\, X}(x) = (\mathsf{var}\,\mathsf{src}\, x, \mathsf{var}\,\mathsf{tgt}\, x).$$

It sends a morhpism of globular sets $f\colon X \to Y$ to the morhpism of n-computads consisting of $\mathsf{Free}_{n-1}\, f$ and the function $\mathsf{var} \circ f_n$. The functor of n-spheres together with the projection natural transformations are defined via the following pullback square

$$\begin{array}{ccc} \mathsf{Sphere}_n & \xrightarrow{\ \mathsf{pr}_1\ } & \mathsf{Cell}_n \\ {\scriptstyle \mathsf{pr}_2}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathsf{bdry}_n} \\ \mathsf{Cell}_n & \xrightarrow[\mathsf{bdry}_n]{} & \mathsf{Sphere}_{n-1}\, u_n \end{array}$$

that is an n-sphere is a pair of n-cells with the same boundary. We will often denote such a pair $(a, b) \in \mathsf{Sphere}_n(C)$ by $a \to b$, and write $c\colon a \to b$ to denote that a cell $c \in \mathsf{Cell}_{n+1}(C)$ has boundary (a, b).

To conclude the induction, it remains to define when an n-sphere of a Batanin tree B is full. To do that, we define the *support* of a cell $c \in \mathsf{Cell}_n(C)$ to be the set of generators appearing in c. More formally, the support of c is defined recursively by

$$\mathsf{supp}_n(\mathsf{var}\, v) = \{v\} \qquad \mathsf{supp}_n(\mathsf{coh}(B, A, f)) = \bigcup_{p \in \mathsf{Pos}_n(B)} \mathsf{supp}_n(f_V(p))$$

and then a sphere $(a, b) \in \mathsf{Sphere}_n(\mathsf{Free}_n(\mathsf{Pos}(B)))$ is declared to be full when the support of a consists of the n-positions in the image of s_n^B and the support of b consists of the n-positions in the image of t_n^B. Moreover, if $n > 0$, we require that the $(n-1)$-sphere $\mathsf{bdry}_n(a) = \mathsf{bdry}_n(b)$ is also full. Full spheres amount to ways to compose the boundary of a globular pasting diagram when $n = \dim B - 1$, and they amount to pairs of parallel ways to compose the whole diagram when $n \geq \dim B$.

The category of computads is then defined to be the limit of the forgetful functors u_n, so a computad $C = (C_n)_{n\in\mathbb{N}}$ is a sequence of n-computads such that $u_{n+1}C_{n+1} = C_n$. The free functors $\mathsf{Free}_n \colon \mathsf{Glob} \to \mathsf{Comp}_n$ are compatible with the forgetful functors, so they give rise to a functor

$$\mathsf{Free} \colon \mathsf{Glob} \to \mathsf{Comp}\,.$$

In the opposite direction, we may define a functor

$$\mathsf{Cell} \colon \mathsf{Comp} \to \mathsf{Glob}$$

sending a computad C to the globular set with cells given by $\mathsf{Cell}_n(C_n)$ and with source and target functions given by the composition of the boundary natural transformation bdry_n with the projections pr_i. The functor Free is left adjoint to Cell. The unit η of the adjunciton is given by the morphisms of globular sets

$$\eta_X \colon X \to \mathsf{Cell}\,\mathsf{Free}(X)$$

sending $x \in X_n$ to the generator cell $\mathsf{var}\, x$. The counit ε consists of the morphisms of computads

$$\varepsilon_C \colon \mathsf{Free}\,\mathsf{Cell}\, C \to C$$

determined by the identity functions

$$V_n^{\mathsf{Free}\,\mathsf{Cell}\, C} = \mathsf{Cell}_n\, C$$

The triangle equations for this adjunction can be easily checked.

Remark 12. The inductive fullness condition above is equivalent to the following one, as shown by Dean et al [13]. We can define more generally for a cell $c \in \mathsf{Cell}_n(C)$ its k-support $\mathsf{supp}_k(c)$ to be the set of k-dimensional generators used in the definition of c, or its source and target. We will say that c covers C when its support contains every generator of C. We then say that a sphere $A = (a, b) \in \mathsf{Sphere}_n(\mathsf{Free}\,\mathsf{Pos}(B))$ is full if and only if the support of a is the image of s_n^B and that of b is the image of t_n^B. This is equivalent in turn to $a = T(s_n^B)(a')$ and $b = T(t_n^B)(b')$ for cells $a', b' \in (T\,\mathsf{Pos}(\partial_n B))_n$ that cover $\mathsf{Free}\,\mathsf{Pos}(\partial_n B)$.

3.2 Operations in ω-categories

Having defined the adjunction between computads and globular sets, we may now define ω-categories. This definition is equivalent to the one of Leinster, as shown by Dean et al [13].

Definition 13. The *free weak ω-category monad* (T, η, μ) is the monad induced by the adjunction $\mathsf{Free} \dashv \mathsf{Cell}$. The category Cat_ω of (weak) ω-categories is the category of T-algebras.

By definition, an ω-category is a pair $\mathbb{X} = (X, \alpha \colon TX \to X)$ satisfying an associativity and a unit axiom. In particular, for every Batanin tree B and every cell $c \in (T\,\mathsf{Pos}(B))_n$, there exists an operation

$$\begin{gathered} c^{\mathbb{X}} \colon \mathsf{Glob}(\mathsf{Pos}(B), X) \to X_n \\ c^{\mathbb{X}}(f) = (\alpha \circ Tf)(c) \end{gathered}$$

that is natural in that for every morphism of ω-categories $g \colon \mathbb{X} \to \mathbb{Y}$,

$$g(c^{\mathbb{X}}(f)) = c^{\mathbb{Y}}(Ug \circ f).$$

It was shown by the second author [25] that those operations, and more specifically the operations of the form

$$\mathsf{coh}^{\mathbb{X}}(B, A, -) = \mathsf{coh}(B, A, \mathsf{id})^{\mathbb{X}}$$

fully determine the structure morphism α and that they can be chosen freely subject to source and target conditions. We will call such operations *compositions* when $\dim B = n$ and *coherences* when $\dim B < n$.

Utilizing the idea that natural operations in an ω-category with arity the pasting diagram $\mathsf{Pos}(B)$ correspond to elements of $T\,\mathsf{Pos}(B)$, to define composition and identity operations in an ω-category, we will construct a family of cells over pasting diagrams. More precisely, we define recursively for every Batanin tree B and every natural number n, a full n-sphere $A_{B,n} \in \mathsf{Full}_n(B)$, and we define a cell $\mathsf{comp}_{n,B}$ with boundary $A_{B,n-1}$ when $n \geq \dim B$ by

$$\begin{aligned} \mathsf{comp}_{B,n} &= \begin{cases} \mathsf{var}(\mathsf{id}_n) & \text{when } B = D_n \\ \mathsf{coh}(B, A_{B,n-1}, \mathsf{id}) & \text{otherwise} \end{cases} \\ A_{B,n} &= (T(s_n^B)(\mathsf{comp}_{\partial_n B,n}), T(t_n^B)(\mathsf{comp}_{\partial_n B,n})) \end{aligned}$$

The *identity* of an ω-category $\mathbb{X}$ is the operation

$$\mathsf{id}_n^{\mathbb{X}} = \mathsf{comp}_{D_n,n+1}^{\mathbb{X}} \colon X_n \cong \mathsf{Glob}(\mathsf{Pos}(D_n), X) \to X_{n+1}$$

taking a cell $x \in X_n$ to a cell with source and target x. The *unbiased composition* over a tree B is the operation

$$\mathsf{comp}_B^{\mathbb{X}} = \mathsf{comp}_{B,\dim B}^{\mathbb{X}} \colon \mathsf{Glob}(\mathsf{Pos}(B), X) \to X_{\dim B},$$

taking a diagram $f\colon \mathsf{Pos}(B) \to X$ to a cell with source and target the unbiased composite of $f \circ s^B$ and $f \circ t^B$ respectively. In particular, for the Batanin tree $B = D_{n_0} *_{k_1} \dots *_{k_m} D_{n_m}$, a morphism $f\colon \mathsf{Pos}(B) \to X$ amounts to a sequence of cells $x_i \in X_{n_i}$ such that x_i and x_{i+1} are k_{i+1}-composable and we will write the unbiased composite of f as

$$\mathsf{comp}^{\mathbb{X}}_B(f) = x_0 *_{k_1} \dots *_{k_m} x_m$$

omitting the index k_i when $\dim x_i = \dim x_{i+1} = k+1$.

3.3 Suspensions and opposites

Constructing high-dimensional operations of ω-category, or equivalently cells over a globular pasting diagram, tends to be a difficult task, leading us to introduce meta-operations that produce new such cells from existing ones. In our previous paper [7], we introduced two such meta-operations, the *suspension* and the *opposite*. The former is obtained by interpreting an operation in the hom ω-categories of an ω-category, while the latter is obtained by interpreting it in its opposites, introduced in the same paper.

More formally, we define mutually recursively the suspension of a computad $\Sigma\colon \mathsf{Comp} \to \mathsf{Comp}$ extending the one for globular sets, together with a natural transformation

$$\begin{array}{ccccc}
\mathsf{Glob} & \xrightarrow{\mathsf{Free}} & \mathsf{Comp} & \xrightarrow{\mathsf{Cell}} & \mathsf{Glob} \\
{\scriptstyle\Sigma}\big\downarrow & = & {\scriptstyle\Sigma}\big\downarrow & \Downarrow{\scriptstyle\Sigma^{\mathsf{Cell}}} & \big\downarrow{\scriptstyle\Sigma} \\
\mathsf{Glob} & \xrightarrow{\mathsf{Free}} & \mathsf{Comp} & \xrightarrow{\mathsf{Cell}} & \mathsf{Glob}
\end{array}$$

by the following recursive formulae

$$\begin{aligned}
(\Sigma\, C)_0 &= \{v_-, v_+\} & (\Sigma\, C)_{n+1} &= ((\Sigma\, C)_n, V^C_n, (\Sigma^{\mathsf{Cell}}, \Sigma^{\mathsf{Cell}}) \circ \phi^C_n) \\
(\Sigma\, f)_0 &= \mathsf{id} & (\Sigma\, f)_{n+1} &= ((\Sigma\, f)_n, \Sigma^{\mathsf{Cell}} \circ \circ f_{V,n}) \\
\Sigma^{\mathsf{Cell}}(\mathsf{var}\, v) &= \mathsf{var}\, v & \Sigma^{\mathsf{Cell}}(\mathsf{coh}(B, A, f)) &= \mathsf{coh}(\Sigma\, B, (\Sigma^{\mathsf{Cell}}, \Sigma^{\mathsf{Cell}})A, \Sigma\, f)
\end{aligned}$$

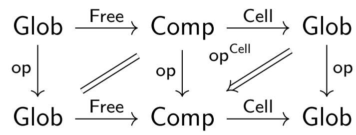
The suspension operation allows us to take an operation in the form of a cell $c \in (T\,\mathsf{Pos}(B))_n$ and produce a new operation $\Sigma^{\mathsf{Cell}}(c) \in (T\,\mathsf{Pos}(\Sigma\, B))_{n+1}$ of higher dimension. For example, suspensing the unbiased composition over the tree $\mathsf{Chain}_k = D_1 *_0 \cdots *_0 D_1$, we obtain the unbiased composition operation over the tree $\Sigma^n\,\mathsf{Chain}_k = D_{n+1} *_n \cdots *_n D_{n+1}$, which corresponds to the composition of k consecutive $(n+1)$-cells.

To define the opposite of a computad, we proceed similarly. Given a set of positive natural numbers $w \in \mathbb{N}_{>0}$, we may define an autoequivalence

$$\mathsf{op}\colon \mathsf{Glob} \to \mathsf{Glob}$$

by swapping the source and target of every n-cell of a globular set when $n \in w$. This operation preserves globular pasting diagrams, in the sense that there exists an automorphism $\mathsf{op}\colon \mathsf{Bat} \to \mathsf{Bat}$ together with isomorphisms $\mathsf{op}^B\colon \mathsf{Pos}(\mathsf{op}\, B) \to$

$\mathsf{op}\,\mathsf{Pos}(B)$ for every Batanin tree B compatible with the source and taregt inclusions. We then extend op to an autoequivalence of the category of computads $\mathsf{op}\colon \mathsf{Comp} \to \mathsf{Comp}$ together with a natural isomorphisms



by similar recursive formulae as above:

$$\begin{aligned}
(\mathsf{op}\,C)_n &= ((\mathsf{op}\,C)_{n-1}, V_n^C, \mathsf{swap}_n \circ (\mathsf{op}^{\mathsf{Cell}}, \mathsf{op}^{\mathsf{Cell}})\phi_n^C) \\
(\mathsf{op}\,f)_n &= ((\mathsf{op}\,f)_{n-1}, \mathsf{op}^{\mathsf{Cell}} \circ f_{V,n}) \\
\mathsf{op}^{\mathsf{Cell}}(\mathsf{var}\,v) &= \mathsf{var}\,v \\
\mathsf{op}^{\mathsf{Cell}}(\mathsf{coh}(B, A, f)) &= \mathsf{coh}(\mathsf{op}\,B, A', (\mathsf{op}\,f) \circ \mathsf{Free}_n(\mathsf{op}^B)) \\
A' &= (T(\mathsf{op}_w^B)^{-1}, T(\mathsf{op}_w^B)^{-1}) \circ \mathsf{swap}_n \circ (\mathsf{op}^{\mathsf{Cell}}, \mathsf{op}^{\mathsf{Cell}})A
\end{aligned}$$

where swap_n swaps the two components of the pullback when $n+1 \in w$ and it is the identity otherwise. The opposite operation allows us to take an operation, e.g. an unbiased composition, and construct a new operation over the pasting diagram with all the cells reversed. For example, the $\{1\}$-opposite of the pasting diagram $B = D_1 *_0 D_2$ is the pasting diagram $\mathsf{op}\,B = D_2 *_0 D_1$. The unbiased composition over B is the left whiskering of an 1-cell with an 2-cell, and it is sent to the unbiased composite over $\mathsf{op}\,B$ which is the right whiskering.

3.4 Invertible and equivalent cells

A crucial notion in the study of higher categories is that of equivalence. This is usually defined by induction on the dimension of the structure: two elements of a 0-category are equivalent when they are equal, while two objects x, y in an $(n+1)$-category are equivalent when there exist morphisms $f\colon x \to y$ and $g\colon y \to x$ such that the compositions $f \circ g$ and $g \circ f$ are equivalent to identities in the respective n-categories of morphisms. In such case, we say that the morphisms f and g are invertible. This definition of equivalence fails may be also used for ω-categories if interpreted coinductively, as observed by Cheng [11]:

Definition 14. The collection of *invertible* cells of an ω-category $\mathbb{X}$ is defined coinductively by saying that a positive-dimensional cell $x\colon u \to v \in X_{n+1}$ is invertible if there exists a cell $x^{-1}\colon v \to u$, together with a pair of invertible cells

$$\mathsf{u}_x\colon x *_n x^{-1} \to \mathsf{id}_n(u) \qquad\qquad \mathsf{v}_x\colon x^{-1} *_n x \to \mathsf{id}_n(v).$$

We say that a pair of cells $c, c' \in X_n$ are *equivalent* and write $c \sim c'$ when there exists an invertible cells with source c and target c'.

The general semantics of coinduction is beyond the scope of this article. To explain the definition, let $\mathbf{X} = \sqcup_n X_{n+1}$ the set of all positive-dimensional cells of $\mathbb{X}$, and consider the endofunction F sending a subset $U \subset \mathbf{X}$ to the set of cells $x \in \mathbf{X}$ for which there exist $x^{-1} \in \mathbf{X}$ and $\mathsf{u}_x, \mathsf{v}_x \in U$ satisfying the same boundary conditions as in Definition 14. The function F is monotone, so by Tarski's fixed point theorem, it has a greatest postfixed point W, which we will call the set of invertible cells of $\mathbb{X}$. By definition, it is the maximum set such that $W \subset F(W)$. This provides a method for proving invertibility of a set of cells U: it suffices to show that $U \subseteq F(U)$ to conclude that $U \subseteq W$. We will use this method to show for example that coherence operations produce invertible cells.

Lemma 15. *The relation $\sim$ is symmetric and preserved by every morphism.*

Proof. Let $\mathbb{X}$ be an ω-category and $c \sim c'$ be equivalent cells. Then there exists an invertible cell x with source c and target c'. The inverse x^{-1} is again invertible with $(x^{-1})^{-1} = x$, $\mathsf{u}_{x^{-1}} = \mathsf{v}_x$ and $\mathsf{v}_{x^{-1}} = \mathsf{u}_x$. Its source is c' and its target is c, so $c' \sim c$. Therefore, the relation $\sim$ is symmetric.

To show that a morphism $f\colon \mathbb{X} \to \mathbb{Y}$ preserves equivalence, we will show coinductively that it preserves invertibility. More precisely, let U the set of cells of $\mathbb{Y}$ that are the image of some invertible cell of $\mathbb{X}$. Then for every cell $y \in U$, there exists some invertible cell x of $\mathbb{X}$ such that $f(x) = y$. Then we may define $y^{-1} = f(x^{-1})$ and $\mathsf{u}_y = f(\mathsf{u}_x) \in U$ and $\mathsf{v}_y = f(\mathsf{v}_x) \in U$. This shows that $U \subset F(U)$, so every cell of U is invertible in $\mathbb{Y}$. □

Proposition 16. *Let B be a Batanin tree and $A \in \mathsf{Full}_{n-1}(B)$ be a full sphere such that $\dim B < n$. Then for every ω-category $\mathbb{X}$ and every morphism of globular sets $f\colon \mathsf{Pos}(B) \to X$, the coherence cell $\mathsf{coh}^{\mathbb{X}}(B, A, f)$ is invertible.*

Proof. We will first show coinductively that for a Batanin tree B, the set of cells U of the form $\mathsf{coh}(B, A, \mathsf{id})$ for some full sphere $A \in \mathsf{Full}_{n-1}(B)$ such that $\dim B < n$ consists of invertible cells. For that, let $x = \mathsf{coh}(B, A, \mathsf{id}) \in X_n$ be such a cell and let $A = (u, v)$. By the assumption on the dimension of A, the n-spheres (u, u), (v, v) and (v, u) are all full, so we may define the inverse cell $x^{-1} = \mathsf{coh}(B, (v, u), \mathsf{id})$. The support of the cells $x *_{n-1} x^{-1}$, $\mathsf{id}_n(u)$, $x^{-1} *_{n-1} x$ and $\mathsf{id}_n(v)$ are empty, since B has no positions of dimension n, and their boundaries are full as explained above. Therefore, we may also define the cells

$$\mathsf{u}_x = \mathsf{coh}(B, x *_{n-1} x^{-1} \to \mathsf{id}_n(u), \mathsf{id}) \in U$$
$$\mathsf{v}_x = \mathsf{coh}(B, x^{-1} *_{n-1} x \to \mathsf{id}_n(v), \mathsf{id}) \in U.$$

By coinduction, it follows that every cell in U is invertible.

Let now $\mathbb{X} = (X, \alpha)$ be an ω-category and $f\colon \mathsf{Pos}(B) \to X$ be a morphism of globular sets. Then $Tf\colon T\,\mathsf{Pos}(B) \to TX$ is a morphism of free ω-categories $F\,\mathsf{Pos}(B) \to FX$, and $\alpha\colon TX \to X$ is a morphism $FX \to \mathbb{X}$, hence the cell

$$\mathsf{coh}^{\mathbb{X}}(B, A, f) = (\alpha \circ Tf)(\mathsf{coh}(B, A, \mathsf{id}))$$

is the image of an invertible cell under a morphism of ω-categories. It follows by Lemma 15 that $\mathsf{coh}^{\mathbb{X}}(B, A, f)$ is invertible. □

Corollary 17. *The relation $\sim$ is reflexive.*

Proof. The corollary follows by invertibility of identity cells. □

4 Constructions in weak ω-categories

This section is dedicated to expanding our toolbox to work with ω-categories. We extend the language of computads with additional constructions, defining simpler ways to describe some of the cells.

4.1 Unbiased unitors

We first define a family of cells that we call the *unbiased unitors*. Intuitively, those are cells which take composite of identities onto an identity. More formally, a composite of identities is a composite where all top-dimensional cells are identities. In order to define those precisely, we rely on the following result.

Lemma 18. *For a Batanin tree B and every $n \in \mathbb{N}$, the source and target inclusions $s_n^B, t_n^B : \mathsf{Pos}(\partial_n B) \to \mathsf{Pos}(B)$ induce bijections between positions of dimension $k < n$. Moreover, they are injective on positions of dimension n, and for every position $p \in \mathsf{Pos}_n(B)$, there exists unique position $q \in \mathsf{Pos}_n(B)$ such that $s_n^B(q)$, $t_n^B(q)$ and p are parallel.*

Proof. The proof is by straightforward induction on the tree and on n. It can be seen also from the pictorial description of trees and their positions. □

Consider a Batanin tree B of dimension d. Using Lemma 18 we define the map $\underline{\mathsf{id}} : \mathsf{Free}\,\mathsf{Pos}(B) \to \mathsf{Free}\,\mathsf{Pos}(\partial_{d-1} B)$, defined recursiveley as follows:

- To each position p of dimension $k < d-1$, it assigns the cell $\mathsf{var}\, q$, where q is the preimage of p
- To each position p of dimension $d-1$, it assigns the cell $\mathsf{var}\, q$, where q is the unique position such that $s_{d-1}^B(q)$ and p are parallel
- To each position p of dimension d, it assigns the cell $\mathsf{id}(\underline{\mathsf{id}}(\mathsf{src}\, p))$

By construction, it follows that $\underline{\mathsf{id}}$ is a common retraction of the source and target inclusions:

$$\underline{\mathsf{id}} \circ \mathsf{Free}(s_{d-1}^B) = \underline{\mathsf{id}} \circ \mathsf{Free}(t_{d-1}^B) = \mathsf{id}_{\mathsf{Free}\,\mathsf{Pos}(\partial B)}$$

Definition 19. Consider a Batanin tree B of dimension d together with a cell $a \in (T\,\mathsf{Pos}(\partial_{d-1} B))_{d-1}$ that covers $\partial_{d-1} B$. The *unbiased unitor* $\mathsf{unitor}(B, a)$ is the cell

$$\mathsf{unitor}(B, a) = \mathsf{coh}(\partial B, u \to v, \mathsf{id}_{\mathsf{Pos}(\partial_{d-1} B)}) \in (T\,\mathsf{Pos}(\partial B))_d$$

where

$$u = \mathsf{coh}(B, T(s_{d-1}^B)(a) \to T(t_{d-1}^B)(a), \underline{\mathsf{id}}) \qquad v = \mathsf{id}(a)$$

The assumption on the support of a implies that both u and v are well-defined cells, while the two cells are parallel by the computation of the source and target of $\underline{\mathsf{id}}$. The cell acts as a unitor, since it takes a composite over B where all top dimensional cells are sent to identities to the identity of the composite.

4.2 Filler cells

We define a collection of coherence operations, that we call fillers, generalising the associators, unitors and interchangers. Those are operations in ω-categories, that we describe again using cells over a globular pasting diagram. Recall that given a diagram of Batanin trees $f\colon \mathsf{Pos}(B) \to \mathsf{Bat}$ indexed by a Batanin tree B, we may form the composite tree $\mu^{\mathsf{str}} f$.

Definition 20. Suppose that for each $i \in \{1, 2\}$, we are given

- a Batanin tree B_i,
- morphisms $f_i : \mathsf{Pos}(B_i) \to \mathsf{Bat}$ such that $\mu^{\mathsf{str}}(f_1) = \mu^{\mathsf{str}}(f_2)$,
- full spheres A_i in $\mathsf{Pos}(B_i)$
- covering morphisms $\sigma_i : \mathsf{Free}\,\mathsf{Pos}(B_i) \to \mathsf{Free}\,\mathsf{Pos}(\mu^{\mathsf{str}}(f_i))$ such that

$$\mathsf{Sphere}(\sigma_1)(A_1) = \mathsf{Sphere}(\sigma_2)(A_2)$$

we define the filler cell

$$\mathsf{fill}(B_i, f_i, a_i \to b_i, \sigma_i) = \mathsf{coh}(\mu^{\mathsf{str}}(f_1), c_1 \to c_2, \mathsf{id})$$

where

$$c_1 = \mathsf{coh}(B_1, A_1, \sigma_1) \qquad\qquad c_2 = \mathsf{coh}(B_2, A_2, \sigma_2).$$

The assumptions on A_i and σ_i ensure exactly that the cells c_1 and c_2 are cell defined, parallel and covering the tree $\mu^{\mathsf{str}}(f_i)$. By Proposition 16, all the fillers are all invertible cells.

Example 21. The associator with source is three binary composed 1-cells, associated on the left and its target is three binary composed 1-cells, associated on the right can be obtained as a filler with:

- The tree $B_1 = D_1 *_0 D_1$, the map $f_1 = \langle D_1 *_0 D_1, D_1\rangle$, the cells $a_1 = \mathsf{var}\, p$ and $b_1 = \mathsf{var}\, q$ where p, q are respectively the left- and right-most 0-positions of B_1, and the morphism $\sigma_1 = \langle \mathsf{comp}_{D_1 *_0 D_1}, \mathsf{id}_{D_1}\rangle$,
- The tree $B_2 = D_1 *_0 D_1$, the map $f_2 = \langle D_1, D_1 *_0 D_1\rangle$, the cells $a_2 = \mathsf{var}\, p$ and $b_2 = \mathsf{var}\, q$ where p, q are respectively the left- and right-most 0-positions of B_2, and morphism $\sigma_2 = \langle \mathsf{id}_{D_1}, \mathsf{comp}_{D_1 *_0 D_1}\rangle$.

Example 22. The unbiased unitor $\mathsf{u}(B, a)$ can also be obtained as a filler, by choosing:

- The tree $B_1 = B$, the map f_1 associating to every position p the tree $D_{\mathsf{min}(\mathsf{dim}\, p, \mathsf{dim}\, B-1)}$, the cells $a_1 = T(s^B_{d-1})(a)$ and $a_2 = T(t^B_{d-1})(a)$, and the map $\sigma_1 = \underline{\mathsf{id}}$.

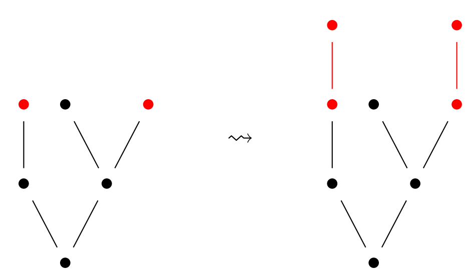
- The tree $B_2 = D_{\mathsf{dim}\, B-1}$, the map f_2 associating to every position p the tree $\partial_{\mathsf{dim}\, p} B$, the cells $a_2 = b_2 = \mathsf{var}\, p$ where p is the unique maximap position of $D_{\mathsf{dim}\, B-1}$, and the map σ_2 corresponding to the cell a via the Yoneda lemma.

4.3 Functorialisation of coherences

Given a Batanin tree B of dimension d together with a set $X \subseteq \mathsf{Pos}_d(B)$ of maximal positions of B (c.f. Definition 3), we define the functorialisation of the tree B with respect to the set X by induction on B, denoted $B \uparrow X$ as follows

$$
\begin{aligned}
B \uparrow \emptyset &= B \\
(\mathsf{br}[]) \uparrow X &= \mathsf{br}[\mathsf{br}[]] \\
(\mathsf{br}[B_1, \ldots, B_n]) \uparrow X &= \mathsf{br}[B_1 \uparrow (\mathsf{in}_1^{-1}(X)), \ldots, B_n \uparrow (\mathsf{in}_n^{-1}(X))]
\end{aligned}
$$

Intuitively, this operation consists in selecting a set of leaves of the tree, and growing one more branch on top of all the selected leaves. For instance, consider the tree displayed on the left. The functorialisation of this tree with respect to the set of locally maximal positions indicated as red produces the tree represented on the right. The newly created branches are also displayed in red, to improve legibility.



Consider for instance the Batanin tree $\mathsf{Chain}_k = D_1 *_0 \ldots *_0 D_1$ and the position f_i^{Chain}. Then the functorialisation is given by

$$
\mathsf{Chain}_k \uparrow f_i^{\mathsf{Chain}} = D_1 *_0 \ldots *_0 D_1 *_0 D_2 *_0 D_1 *_0 \ldots *_0 D_1,
$$

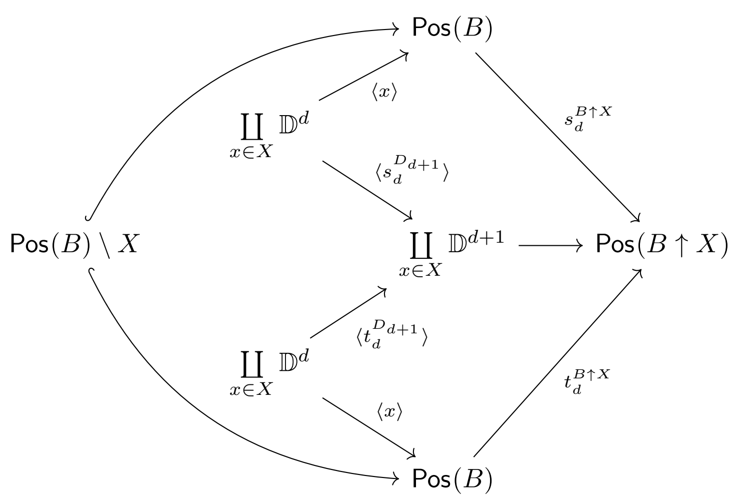
where in this expression, the disk D_2 appears in the i-th position.

Lemma 23. *For a Batanin tree B of dimension d, and a non-empty set X of maximal positions in B, the dimension of the functorialisation is given by $\mathsf{dim}(B \uparrow X) = d + 1$, and we have $\partial_d(B \uparrow X) = B$.*

Proof. We proceed by induction on B: If $B = \mathsf{br}[]$, then $\mathsf{dim}\, B = 0$ and $\mathsf{dim}(B \uparrow X) = \mathsf{dim}(D_1) = 1$, and moreover, $\partial_0(B \uparrow X) = B$. If $B = \mathsf{br}[B_1, \ldots, B_n]$, then $\mathsf{dim}\, B = 1 + \mathsf{max}\,(\mathsf{dim}\, B_i)$, and $\mathsf{dim}(B \uparrow X) = 1 + \mathsf{max}\,(\mathsf{dim}(B_i \uparrow \mathsf{in}_i^{-1}(X)))$. By induction, we have $\mathsf{dim}(B_i \uparrow \mathsf{in}_i^{-1}(X)) = 1 + \mathsf{dim}\, B_i$ if $\mathsf{in}_i^{-1}(X)$ is non-empty, and $\mathsf{dim}\, B_i$ otherwise. Moreover, since X contains at least one maximal position of B there exists at least one i such that $\mathsf{in}_i^{-1}(X)$. We then have necessarily $1+\mathsf{dim}\, B_i = \mathsf{dim}\, B$, and thus $\mathsf{dim}(B \uparrow X) = 1+\mathsf{dim}\, B$. Moreover, denote $\mathsf{dim}\, B = d$, we have $\partial_d(B \uparrow X) = \mathsf{br}[\partial_{d-1}(B_1 \uparrow \mathsf{in}_1^{-1}(X)), \ldots, \partial_{d-1}(B_n \uparrow \mathsf{in}_n^{-1}(X))]$. For every i such that $\mathsf{in}_i^{-1}(X) \neq \emptyset$, we have by induction $\partial_{d-1}(B_i \uparrow \mathsf{in}_i^{-1}(X)) = B_i$. Moreover, if $\mathsf{in}_i^{-1}(X) = \emptyset$, then $B_i \uparrow \mathsf{in}_i^{-1}(X) = B_i$ is of dimension at most $d-1$ and $\partial_{d-1} B_i = B_i$. Hence $\partial_d(B \uparrow X) = B$. □

Definition 24. Given a Batanin tree B with a set X of maximal positions, we define the globular set $\mathsf{Pos}(B) \setminus X$ by letting $(\mathsf{Pos}(B) \setminus X)_n = \mathsf{Pos}_n(B) \setminus X$, with the source and target maps the restriction of those of $\mathsf{Pos}(B)$.

Lemma 25. *Given a Batanin tree B of dimension d with a set X of maximal positions, there exits a colimit cocone of the form*



The unnamed morphism picks the generators corresponding to the newly grown branches of the functorialised tree.

The proof of the lemma is by a straightforward induction on the tree B. This gives a method for constructing a map $\sigma : \mathsf{Pos}(B \uparrow X) \to Y$: Such a map amounts to a pair of maps $\sigma_-, \sigma_+ : \mathsf{Pos}(B) \to Y$ which coincide on $\mathsf{Pos}(B) \setminus X$ as well as, for every $x \in X$, a cell $f_x \in Y_{n+1}$ such that $\mathsf{src}(f_x) = \sigma_-(x)$ and $\mathsf{tgt}(f_x) = \sigma_+(x)$.

Remark 26. Lemma 25 gives a characterisation of the functorialisation without relying on the Batanin tree, hinting at a more genreal functorialisation operation valid on every globular set, or even every computad with respect to well chosen generators. This construction has been studied by the first author in the setting of the type theory catt [5], and will not be useful for the purpose of this article.

Having defined the functorialisation of a Batanin tree, we may define the functorialisation of an operation $\mathsf{coh}((,B,,)A,\mathsf{id})$ in a straightforward manner. This is a special instance of the functorialisation of a cell, described by the first author [5].

Definition 27. Given a Batanin tree B of dimension d together with a set of maximal positions $X \subset \mathsf{Pos}(B)$, and two parallel cells $a, b \in T\,\mathsf{Pos}(\partial_{d-1}B)$ that cover $\partial_{d-1}B$, we define the functorialisation

$$(B, a \to b) \uparrow X = \mathsf{coh}(B \uparrow X, T(s_d^{B\uparrow X})(c) \to T(t_d^{B\uparrow X})(c), \mathsf{id}_{\mathsf{Pos}(B\uparrow X)})$$

where $c = \mathsf{coh}(B, a \to b, \mathsf{id}_{\mathsf{Free\,Pos}(B)})$ is covering $T\,\mathsf{Pos}(B)$.

4.4 Chain reduction

Before introducing our final operation on trees, that we call *chain reduction*, we need to introduce a family of trees that we call *chains*. Those are easily defined for $k \in \mathbb{N}$ as the composites $\mathsf{Chain}_k = D_1 *_0 \cdots *_0 D_1$, which correspond to the composition of k consecutive 1-cells. Explicitly the globular set of positions of Chain_k is the following diagram

$$x_0^{\mathsf{Chain}} \xrightarrow{f_1^{\mathsf{Chain}}} x_1^{\mathsf{Chain}} \xrightarrow{f_2^{\mathsf{Chain}}} \dots \xrightarrow{f_k^{\mathsf{Chain}}} x_k^{\mathsf{Chain}}$$

We note that chains are precisely the trees of dimension at most 1. We can then get higher dimensional chains $\Sigma^n\,\mathsf{Chain}_d$ by using the suspension operation. The tree $\Sigma^n\,\mathsf{Chain}_d = D_{n+1} *_n \cdots *_n D_{n+1}$ is the tree consisting of k consecutive $(n+1)$-cells.

The *chain reduction* of a Batanin tree B of dimension d is a new tree B^{red} obtained by merging the chains of positions of dimension d in B into a unique position. It is defined together with a morphism

$$\mathsf{red}_B \colon \mathsf{Free\,Pos}(B^{\mathsf{red}}) \to \mathsf{Free\,Pos}(B)$$

that sends the merged position into the composite of the original chain. To define the chain reduction of a tree, we define more generally for $i \geq \dim B - 1$ a new tree B^{red_i} recursively as follows

$$\begin{aligned} \mathsf{br}[]^{\mathsf{red}_i} &= \mathsf{br}[] \\ \mathsf{Chain}_k{}^{\mathsf{red}_0} &= \mathsf{br}[\mathsf{br}[]] \\ \mathsf{br}[B_1, \dots, B_n]^{\mathsf{red}_{i+1}} &= \mathsf{br}[B_1{}^{\mathsf{red}_i}, \dots, B_n{}^{\mathsf{red}_i}]. \end{aligned}$$

In these definitions, the first case takes precedence over the second one. It follows that $B^{\mathsf{red}_i} = B$ for $i \geq \dim B$, so that the only newly introduced operation is $B^{\mathsf{red}_{\dim B-1}}$ which we will denote simply by B^{red}. For the morphism of computads red_B, we define again a morphisms $\mathsf{red}_{i,B} \colon \mathsf{Free\,Pos}(B^{\mathsf{red}_i}) \to \mathsf{Free\,Pos}(B)$

recursively as follows:

$$\begin{aligned}\mathsf{red}_{i,\mathsf{br}[]} &= \mathsf{id}_{\mathsf{Free}\,\mathbb{D}^0}\\ \mathsf{red}_{0,\mathsf{Chain}_k} &= \mathsf{comp}_{\mathsf{Chain}_k}\\ \mathsf{red}_{i+1,\mathsf{br}[B_1,\ldots,B_n]} &= \langle \Sigma\,\mathsf{red}_{i,B_1},\ldots,\Sigma\,\mathsf{red}_{i,B_n}\rangle.\end{aligned}$$

Those morphisms are again identities for $i \geq \mathsf{dim}\,B$ and we denote simply $\mathsf{red}_{\mathsf{dim}\,B-1,B}$ by red_B. Here, $\mathsf{comp}_{\mathsf{Chain}_k}$ is the morphism $\mathsf{Free}\,\mathbb{D}^1 \to \mathsf{Free}\,C_k$, which via the adjunction $\mathsf{Free} \dashv \mathsf{Cell}$ corresponds to the map $\mathbb{D}^1 \to TC_k$ given by the Yoneda lemma on the cell with the same name. We note also that the last case uses the equality $\Sigma\,\mathsf{Free} = \mathsf{Free}\,\Sigma$ proven in [7], as well as the fact that Free is left adjoint and thus preserves colimits.

Lemma 28. *Consider a Batanin tree B of dimension d. Then, for every maximal position p of B^{red}, there exists a natural number $\mathsf{length}_B(p)$ and a map $\mathsf{chain}_{B,p}$ satsifying the following:*

$$\begin{aligned}&\mathsf{chain}_{B,p}\colon\ \mathsf{Pos}(\Sigma^{d-1}\,\mathsf{Chain}_{\mathsf{length}_B(p)}) \to \mathsf{Pos}(B)\\ &\mathsf{red}_B(p) = T(\mathsf{chain}_{B,p})(\mathsf{comp}_{\Sigma^{d-1}\,\mathsf{Chain}_{\mathsf{length}_B(p)}}).\end{aligned}$$

Proof. For the sake of simplicity, we often omit the index B in length and chain. We prove this result by induction on d: If $d \leq 1$, then by definition $B = \mathsf{Chain}_k$ for unique k, so we may let $\mathsf{length}(p) = k$ and $\mathsf{chain}_p = \mathsf{id}_{\mathsf{Pos}(B)}$. So it suffices to prove the result for $d > 1$, assuming it holds for trees of dimension lower than d. In this case, we write $B = \mathsf{br}[B_1,\ldots,B_l]$. By induction, considering j such that $\mathsf{dim}(B_j) = \mathsf{dim}(B) - 1$, for every maximal position p_j of B_j, we get an number $\mathsf{length}_{B_j}(p_j) \in \mathbb{N}_{>0}$ and a map

$$\begin{aligned}&\mathsf{chain}_{p_j}\colon\ \mathsf{Free}\,\mathsf{Pos}(\Sigma^{d-2}\,\mathsf{Chain}_{\mathsf{length}(p_j)}) \to \mathsf{Free}\,\mathsf{Pos}(B_j)\\ &\mathsf{red}_B(p_j) = T(\mathsf{chain}_{p_j})(\mathsf{comp}_{\Sigma^{d-2}\,\mathsf{Chain}_{\mathsf{length}(p_j)}}).\end{aligned}$$

Consider the canonical map $\mathsf{in}_j : \Sigma\,\mathsf{Pos}(B_i) \to \mathsf{Pos}(B)$, then for $p = \mathsf{in}_j(\Sigma\,p_j)$, we let $\mathsf{length}_B(p) = \mathsf{length}_{B_j}(p_j)$ together with the map chain_p to be the following composite

$$\mathsf{chain}_p = \mathsf{in}_j \circ \Sigma\,\mathsf{chain}_{p_j}\,.$$

For this definition, we use the equality $\Sigma \circ \mathsf{Pos} = \mathsf{Pos} \circ \Sigma$ (c.f. [7]). We have the following equation:

$$\begin{aligned}\mathsf{red}_B(p) &= \langle \Sigma\,\mathsf{red}_{i,B_1},\ldots,\Sigma\,\mathsf{red}_{i,B_n}\rangle(\mathsf{in}_j(\Sigma\,p_j))\\ &= T(\mathsf{in}_j)\,\Sigma^{\mathsf{Cell}}(\mathsf{red}_{i,B_j}(p_j))\\ &= T(\mathsf{in}_j \circ (\Sigma\,\mathsf{chain}_{p_j}))(\mathsf{comp}_{\Sigma^{d-1}\,\mathsf{Chain}_{\mathsf{length}(p_j)}})\\ &= T(\mathsf{chain}_p)(\mathsf{comp}_{\Sigma^{d-1}\,\mathsf{Chain}_{\mathsf{length}(p)}}).\end{aligned}$$ □

Taking the reduction of a Batanin tree does not change its boundary. This equality is compatible with the source and target inclusion of the boundary in the pasting scheme, as stated by the following result.

Lemma 29. *Given a Batanin tree B of dimension d, we have the equality $\partial_{d-1}B^{\mathsf{red}} = \partial B$ and the following squares commute*

$$\begin{array}{ccc} \mathsf{Free}\,\mathsf{Pos}(\partial_{d-1}(B^{\mathsf{red}})) & \xrightarrow{\mathsf{Free}\ s_{d-1}^{B^{\mathsf{red}}}} & \mathsf{Free}\,\mathsf{Pos}(B^{\mathsf{red}}) \\ \| & & \downarrow{\scriptstyle \mathsf{red}_B} \\ \mathsf{Free}\,\mathsf{Pos}(\partial_{d-1}B) & \xrightarrow{\mathsf{Free}\ s_{d-1}^{B}} & \mathsf{Free}\,\mathsf{Pos}(B) \end{array}$$

$$\begin{array}{ccc} \mathsf{Free}\,\mathsf{Pos}(\partial_{d-1}(B^{\mathsf{red}})) & \xrightarrow{\mathsf{Free}\ t_{d-1}^{B^{\mathsf{red}}}} & \mathsf{Free}\,\mathsf{Pos}(B^{\mathsf{red}}) \\ \| & & \downarrow{\scriptstyle \mathsf{red}_B} \\ \mathsf{Free}\,\mathsf{Pos}(\partial_{d-1}B) & \xrightarrow{\mathsf{Free}\ t_{d-1}^{B}} & \mathsf{Free}\,\mathsf{Pos}(B) \end{array}$$

Proof. We will show by induction that $\partial_i B^{\mathsf{red}_i} = B^{\mathsf{red}_i}$ for all $i \geq \mathsf{dim}\, B - 1$ and that the corresponding diagrams commute. The case where $i \geq \mathsf{dim}\, B$ is trivial with both sides equal to B, so we let $i = \mathsf{dim}\, B - 1$. The case where $B = \mathsf{br}[]$ is then vacuously true. In the case where $B = \mathsf{Chain}_k$ for some k and $i = 0$, we have $\partial_0 \mathsf{Chain}_k = \mathsf{br}[] = \partial_0 \mathsf{br}[\mathsf{br}[]]$. Moreover, we have

$$\mathsf{red}_{0,\mathsf{Chain}_k} \circ \mathsf{Free}(s_0^{D_1}) = \mathsf{src}(\mathsf{comp}_{\mathsf{Chain}_k}) = \mathsf{Free}(s_0^{\mathsf{Chain}_k}).$$

Finally, for $i > 0$, we have $B = \mathsf{br}[B_1, \ldots, B_n]$, and by induction, $\partial_{i-1}B_k = \partial_{i-1}B_k^{\mathsf{red}_{i-1}}$, which imply the equality $\partial_i B = \partial_i B^{\mathsf{red}_i}$. Moreover, we have that

$$\begin{aligned} &\mathsf{red}_{i,B} \circ \mathsf{Free}(s_i^{B^{\mathsf{red}_i}}) \\ &\quad = \langle \Sigma(\mathsf{red}_{i-1,B_1} \circ \mathsf{Free}(s_{i-1}^{B_1^{\mathsf{red}_{i-1}}})), \ldots, \Sigma\, \mathsf{red}_{i-1,B_n}\, \mathsf{Free}(s_{i-1}^{B_n^{\mathsf{red}_{i-1}}})\rangle \\ &\quad = \langle \Sigma\, \mathsf{Free}(s_{i-1}^{B_1}), \ldots \Sigma\, \mathsf{Free}(s_{i-1}^{B_n})\rangle \\ &\quad = \mathsf{Free}(s_i^{B}) \end{aligned}$$

□

If two Batanin trees B and B' have the same chain reduction, then they must be of the same dimension and composable. The following lemma identifies the reduction of their composite, as well as the chain maps.

Lemma 30. *Given two Batanin trees B, B' of dimension d such that $B^{\mathsf{red}} = B'^{\mathsf{red}}$, we have $(B *_{d-1} B')^{\mathsf{red}} = B^{\mathsf{red}}$. Moreover, for every maximal position $p \in \mathsf{Pos}_d(B^{\mathsf{red}})$, we have that*

$$\mathsf{length}_{B *_{d-1} B'}(p) = \mathsf{length}_B(p) + \mathsf{length}_{B'}(p)$$

and that the following diagrams commute:

$$\begin{array}{ccc}\Sigma^{d-1}\,\mathsf{Pos}(\mathsf{Chain}_{\mathsf{length}_B(p)}) & \xrightarrow{\mathsf{chain}_{B,p}} & \mathsf{Pos}(B) \\ {\scriptstyle \Sigma^{d-1}\,\mathsf{in}^-}\big\downarrow & & \big\downarrow{\scriptstyle \mathsf{in}^-} \\ \Sigma^{d-1}\,\mathsf{Pos}(\mathsf{Chain}_{\mathsf{length}_{B*_{d-1}B'}(p)}) & \xrightarrow[\mathsf{chain}_p]{} & \mathsf{Pos}(B *_{d-1} B')\end{array}$$

$$\begin{array}{ccc}\Sigma^{d-1}\,\mathsf{Pos}(\mathsf{Chain}_{\mathsf{length}_{B'}(p)}) & \xrightarrow{\mathsf{chain}_{B',p}} & \mathsf{Pos}(B) \\ {\scriptstyle \Sigma^{d-1}\,\mathsf{in}^+}\big\downarrow & & \big\downarrow{\scriptstyle \mathsf{in}^+} \\ \Sigma^{d-1}\,\mathsf{Pos}(\mathsf{Chain}_{\mathsf{length}_{B*_{d-1}B'}(p)}) & \xrightarrow[\mathsf{chain}_p]{} & \mathsf{Pos}(B *_{d-1} B')\end{array}$$

for $\mathsf{in}^{\pm}$ *the pushout inclusions associated to* $\mathsf{Chain}_{k+l} = \mathsf{Chain}_k *_0 \mathsf{Chain}_l$.

Proof. We proceed by induction on the dimension of B and B'. If both trees are of dimension at most 1, then the identity $\mathsf{Chain}_{k+l} = \mathsf{Chain}_k *_0 \mathsf{Chain}_l$ imploes the result on length, the maps chain_p are identity maps. If B and B' are of dimension $d+1$, then we write $B = \mathsf{br}[B_1, \dots, B_n]$ and $B' = \mathsf{br}[B'_1, \dots, B'_m]$. The condition $B^{\mathsf{red}} = B'^{\mathsf{red}}$ implies that $m = n$ and that for every i, ${B_i}^{\mathsf{red}} = {B'_i}^{\mathsf{red}}$. Then, given a maximal position $p \in \mathsf{Pos}(B^{\mathsf{red}})$, there exists $p_i \in \mathsf{Pos}({B_i}^{\mathsf{red}})$ such that $\mathsf{in}_i(p_i) = p$. We then have by induction

$$\begin{aligned}\mathsf{length}_{B*_dB'}(p) &= \mathsf{length}_{B_i *_{d+1} B'_i}(p_i) \\ &= \mathsf{length}_{B_i}(p_i) + \mathsf{length}_{B'_i}(p_i) \\ &= \mathsf{length}_B(p) + \mathsf{length}_{B'}(p).\end{aligned}$$

Moreover, we can prove the first square by induction as follows:

$$\begin{aligned}\mathsf{chain}_{B*_dB',p} \circ \Sigma^d\,\mathsf{in}^- &= \mathsf{in}_i \circ \Sigma(\mathsf{chain}_{B_i *_d B'_i, p_i} \circ \Sigma^{d-1}\,\mathsf{in}^-) \\ &= \mathsf{in}_i \circ \Sigma(\mathsf{in}^- \circ \mathsf{chain}_{B_i,p_i}) \\ &= \mathsf{in}^- \circ \mathsf{in}_i \circ \Sigma\,\mathsf{chain}_{B_i,p_i} \\ &= \mathsf{in}^- \circ \mathsf{chain}_{B,p}\,.\end{aligned}$$

The commutativity of the other square is similar. □

Using those lemmas, we may now define the chain reduction of a composite cell in a computad. Suppose that a computad C is given together with a cell

$$c = \mathsf{coh}(B, A, \sigma) \in \mathsf{Cell}_d(C)$$

where $d = \dim B$. As explained in Remark 12 that imples that

$$A = T(s^B_{d-1})(a) \to T(t^B_{d-1})(b)$$

for some covering cells $a, b \in (T\,\mathsf{Pos}(\partial_{d-1}B))_{d-1}$. Using Lemma 29, we define the chain reduction of c to be

$$c^{\mathsf{red}} = \mathsf{coh}(B^{\mathsf{red}}, s_{d-1}^{B^{\mathsf{red}}}(a) \to t_{d-1}^{B^{\mathsf{red}}}(b), \sigma \circ \mathsf{red}_B) \in \mathsf{Cell}(C).$$

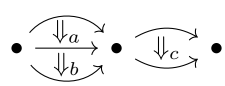
This cell defines a particular biasing of c, in the sense that it is a cell which is weakly equivalent to the cell c. This can be seen by constructing a filler

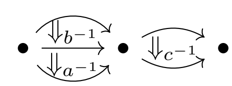
$$\mathsf{assoc}(c) = \mathsf{fill}\,(B_i, f_i, a_i \to b_i, \sigma_i) : c \to c^{\mathsf{red}}.$$

where $B_1 = B$, f_1 is the map associating to p the disc $D^{\mathsf{dim}\,p}$, $a_1 = T(s_{d-1}^{B})(a)$, $b_1 = T(t_{d-1}^{B})(b)$ and $\sigma_1 = \sigma$, while $B_2 = B^{\mathsf{red}}$, f_2 is the map associating to every maximal position p the tree $\Sigma^{d-1}\,\mathsf{Chain}_{\mathsf{length}(p)}$, $a_2 = T(s_{d-1}^{B^{\mathsf{red}}})(a)$, $b_2 = T(t_{d-1}^{B^{\mathsf{red}}})(b)$ and $\sigma_2 = \sigma \circ \mathsf{red}_B$. This filler is an invertible cell whose source is c and whose target is c^{red}.

5 Composite of invertible cells

Our aim in this section is to show that a cell c obtained as a composite of invertible cells is invertible. We achieve this by constructing explicitly the inverse c^{-1} and the witnesses $\mathsf{u}_c, \mathsf{v}_c$. Our strategy for constructing these cells can be illustrated on a simple example: Consider the following composite cell c displayed on the left hand side (with a, b, c invertible in $\mathbb{X}$), the inverse that we construct is displayed on the right hand-side

$c =$ 

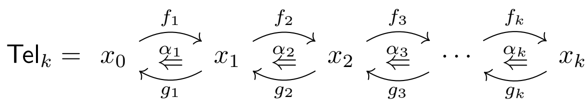
$c^{-1} =$ 

We note that the order of a and b must be swapped, so c^{-1} must be a composite over the opposite pasting diagram. Defining the cell u_x requires composing u_a, u_b and u_c together in order to cancel each of the cells with its inverse. This can be done in multiple ways, as long as b is canceled before a. In order to get a systematic scheme, we define the cancellation of the composite of a and b with the composite of b and a as an intermediate step, called a *telescope* below, and we then perform the cancellation of this composite in parallel with the cancellation of c in an unbiased way. In general, we cancel the composition of maximal cells in codimension 1 first and then proceed in an unbiased way.

5.1 Telescopes

We define the *telescopes*, a family of operations that allows us to cancel sequences of consecutive cells composed with their inverses. Again, using the suspension, it suffices to define those operations for the composite of k consecutive 1-cells in codimension 0. Contrary to previously defined operations such as identities and composites, the telescopes have arities computads that are not globular pasting diagrams. In general, such cells can be thought of as "proof tactics". Here the tactics that we are describing consist of the following steps: associate together the two cells in the middle of a composite of even arity, rewrite their composition into an identity, cancel the identity, and repeat inductively.

We start by defining the 2-computad Tel_k for $k \in \mathbb{N}_{>0}$, which are the smallest computads in which the telescopes cells are defined. Those computads can be visualised as



where $\alpha_i \colon f_i *_0 g_i \to \mathsf{id}$. More formally, the computad Tel_k is determined by

$$
\begin{aligned}
V_0^{\mathsf{Tel}_k} &= \{x_0^{\mathsf{Tel}}, \dots, x_k^{\mathsf{Tel}}\} & \phi_1^{\mathsf{Tel}_k}(f_i^{\mathsf{Tel}}) &= x_i^{\mathsf{Tel}} \to x_{i+1}^{\mathsf{Tel}} \\
V_1^{\mathsf{Tel}_k} &= \{f_1^{\mathsf{Tel}}, g_1^{\mathsf{Tel}} \dots, f_k^{\mathsf{Tel}}, g_k^{\mathsf{Tel}}\} & \phi_1^{\mathsf{Tel}_k}(g_i^{\mathsf{Tel}}) &= x_{i+1}^{\mathsf{Tel}} \to x_i^{\mathsf{Tel}} \\
V_2^{\mathsf{Tel}_k} &= \{\alpha_1^{\mathsf{Tel}}, \dots, \alpha_k^{\mathsf{Tel}}\} & \phi_2^{\mathsf{Tel}_k}(\alpha_i^{\mathsf{Tel}}) &= f_i^{\mathsf{Tel}} *_0 g_i^{\mathsf{Tel}} \to \mathsf{id}(x_{i-1}^{\mathsf{Tel}})
\end{aligned}
$$

and by $V_n^{\mathsf{Tel}_k} = \emptyset$ for $k > 2$, where the generators x_i, f_i, g_i, α_i are drawn from disjoint countable sets. The inclusion of the generators of Tel_k into generators of Tel_{k+1} induces a generator-preserving monomorphism of computads, that we denote by $\mathsf{i}_k : \mathsf{Tel}_k \hookrightarrow \mathsf{Tel}_{k+1}$. Moreover, the 1-generators of the computad Tel_k can be composed: we may first define the morphism of computads loop_k : $\mathsf{Free}\,\mathsf{Pos}(\mathsf{Chain}_{2k}) \to \mathsf{Tel}_k$, characterised by

$$
\mathsf{loop}_k(f_i^{\mathsf{Chain}}) = \begin{cases} \mathsf{var}\, f_i^{\mathsf{Tel}} & \text{If } 1 \leq i \leq k \\ \mathsf{var}\, g_{2k-i+1}^{\mathsf{Tel}} & \text{If } k+1 \leq i \leq 2k \end{cases}
$$

This lets us construct the cell $\mathsf{Cell}(\mathsf{loop}_k)(\mathsf{comp}_{\mathsf{Chain}_{2k}}) \in \mathsf{Cell}(\mathsf{Tel}_k)$. One can check that it is a map whose source and target are both given by $\mathsf{var}\, x_0$.

Theorem 31. *There exists a cell* $\mathsf{tel}_k \in \mathsf{Cell}(\mathsf{Tel}_k)$ *whose source is the composite* $\mathsf{Cell}(\mathsf{loop}_k)(\mathsf{comp}_{\mathsf{Chain}_{2k}})$ *and whose target is the identity* $\mathsf{id}_0(x_0)$.

Proof. We construct the cell tel_k by induction on k, strting from tel_0 being the identity. We choose then $\mathsf{tel}_1 = \mathsf{var}\,\alpha_1^{\mathsf{Tel}}$, and we define tel_{k+1} as the composite of four cells, corresponding to the four steps in the proof tactics that tel_{k+1} encodes:

$$
\mathsf{tel}_{k+1} = \mathsf{tel}_{k+1}^1 *_1 \mathsf{tel}_{k+1}^2 *_1 \mathsf{tel}_{k+1}^3 *_1 \mathsf{Cell}(\mathsf{i}_k)(\mathsf{tel}_k).
$$

It suffices to define the cells tel_{k+1}^1, tel_{k+1}^2 and tel_{k+1}^3. The cell tel_{k+1}^1 is the application of an associator that takes the unbiased composite of the boundary of the telescope to a composite where the cells $\mathsf{var}\, f_{k+1}^{\mathsf{Tel}}$ and $\mathsf{var}\, g_{k+1}^{\mathsf{Tel}}$ are associated together. To define this cell, we note that there is a morphism of computads $\sigma\colon \mathsf{Free}\,\mathsf{Pos}(\mathsf{Chain}_{2k+1}) \to \mathsf{Free}\,\mathsf{Pos}(\mathsf{Chain}_{2k+2})$, defined by

$$
\sigma_V(f_i^{\mathsf{Chain}}) = \begin{cases} \mathsf{var}\, f_i^{\mathsf{Chain}} & \text{if } i < k+1 \\ (\mathsf{var}\, f_{k+1}^{\mathsf{Chain}}) *_0 (\mathsf{var}\, f_{k+2}^{\mathsf{Chain}}) & \text{if } i = k+1 \\ \mathsf{var}\, f_{i+1}^{\mathsf{Chain}} & \text{if } i > k+1 \end{cases}
$$

We then define the first cell of the telescope to be the following cell, given here with its source and target:

$$\begin{aligned}
\mathsf{tel}^1_{k+1} &= \mathsf{coh}(\mathsf{Chain}_{2k+2}, \mathsf{comp}_{\mathsf{Chain}_{2k+2}} \to \mathsf{Cell}(\sigma)(\mathsf{comp}_{\mathsf{Chain}_{2k+1}}), \mathsf{loop}_{k+1}) \\
\mathsf{src}(\mathsf{tel}^1_{k+1}) &= f^{\mathsf{Tel}}_1 *_0 \dots *_0 f^{\mathsf{Tel}}_{k+1} *_0 g^{\mathsf{Tel}}_{k+1} *_0 \dots *_0 g^{\mathsf{Tel}}_1 \\
\mathsf{tgt}(\mathsf{tel}^1_{k+1}) &= f^{\mathsf{Tel}}_1 *_0 \dots *_0 f^{\mathsf{Tel}}_k *_0 (f^{\mathsf{Tel}}_{k+1} *_0 g^{\mathsf{Tel}}_{k+1}) *_0 g^{\mathsf{Tel}}_k *_0 \dots *_0 g^{\mathsf{Tel}}_1.
\end{aligned}$$

The second cell consists in using the generator $\alpha^{\mathsf{Tel}}_{k+1}$ in order to relate $f^{\mathsf{Tel}}_{k+1} *_0 g^{\mathsf{Tel}}_{k+1}$ to the identity on x^{Tel}_k. We define the cell as follows, given here with its source and target:

$$\begin{aligned}
\mathsf{tel}^2_{k+1} &= \mathsf{Cell}(\langle f^{\mathsf{Tel}}_1, \dots, f^{\mathsf{Tel}}_k, \alpha^{\mathsf{Tel}}_{k+1}, g^{\mathsf{Tel}}_k, \dots, f^{\mathsf{Tel}}_1 \rangle)((\mathsf{comp}_{(\mathsf{Chain}_{2k+1})\uparrow f^{\mathsf{Chain}}_{k+1}})) \\
\mathsf{src}(\mathsf{tel}^2_{k+1}) &= f^{\mathsf{Tel}}_1 *_0 \dots *_0 f^{\mathsf{Tel}}_k *_0 (f^{\mathsf{Tel}}_{k+1} *_0 g^{\mathsf{Tel}}_{k+1}) *_0 g^{\mathsf{Tel}}_k *_0 \dots *_0 g^{\mathsf{Tel}}_1 \\
\mathsf{tgt}(\mathsf{tel}^2_{k+1}) &= f^{\mathsf{Tel}}_1 *_0 \dots *_0 f^{\mathsf{Tel}}_k *_0 \mathsf{id}(x^{\mathsf{Tel}}_k) *_0 g^{\mathsf{Tel}}_k *_0 \dots *_0 g^{\mathsf{Tel}}_1.
\end{aligned}$$

The last cell removes the identity in the middle of the composite. We define a map $\tau : \mathsf{Free\,Pos}(\mathsf{Chain}_{2k+1}) \to \mathsf{Free\,Pos}(\mathsf{Chain}_{2k})$, by the following assignment:

$$\tau(f^{\mathsf{Chain}}_i) = \begin{cases} \mathsf{var}\, f^{\mathsf{Chain}}_i & \text{if } i < k+1 \\ \mathsf{id}(\mathsf{var}\, x^{\mathsf{Chain}}_k) & \text{if } i = k+1 \\ \mathsf{var}\, f^{\mathsf{Chain}}_{i-1} & \text{if } i > k+1. \end{cases}$$

This lets us define the last of the three cells as follows, givene again with their sources and targets:

$$\begin{aligned}
\mathsf{tel}^3_{k+1} &= \mathsf{coh}(\mathsf{Chain}_{2k}, \mathsf{Cell}(\tau)(\mathsf{comp}_{\mathsf{Chain}_{2k+1}}) \to \mathsf{comp}_{\mathsf{Chain}_{2k}}, \mathsf{loop}_k) \\
\mathsf{src}(\mathsf{tel}^3_{k+1}) &= f^{\mathsf{Tel}}_1 *_0 \dots *_0 f^{\mathsf{Tel}}_k *_0 \mathsf{id}(x^{\mathsf{Tel}}_k) *_0 g^{\mathsf{Tel}}_k *_0 \dots *_0 g^{\mathsf{Tel}}_1 \\
\mathsf{tgt}(\mathsf{tel}^3_{k+1}) &= f^{\mathsf{Tel}}_1 *_0 \dots *_0 f^{\mathsf{Tel}}_k *_0 g^{\mathsf{Tel}}_k *_0 \dots *_0 g^{\mathsf{Tel}}_1.
\end{aligned}$$

This finishes the definition of the cell tel_{k+1}. □

5.2 Pointwise inversion of a map

When a map sends all maximal position onto invertible cells, one can derive a new map, obtained by inverting all the images of the maximal positions. This process changes the order of compositions of maximal positions in codimension 1, which we handle by using the opposite pasting diagram.

Lemma 32. *Consider an ω-category $\mathbb{X} = (X, \alpha : TX \to X)$, and a Batanin tree B of dimension d, together with a map $\sigma : \mathsf{Pos}(B) \to X$ sending every maximal position $p \in \mathsf{Pos}(B)$ onto an invertible cell in $\mathbb{X}$. Then, there exists a map $\bar{\sigma} : \mathsf{Pos}(B) \to X$ such that for every maximal position p of $\mathsf{Pos}(B)$, we have $\bar{\sigma}_V(p) = (\mathsf{op}_{\{d\}}(\sigma)(\mathsf{op}^B_{\{d\}}(p)))^{-1}$. Moreover, we have*

$$\sigma \circ s^B_{d-1} = \bar{\sigma} \circ t^B_{d-1} \qquad \sigma \circ t^B_{d-1} = \bar{\sigma} \circ s^B_{d-1}.$$

Proof. We proceed by induction on the dimension of B, and since a cell of dimension 0 cannot be invertible, we initialise the induction at dimension 1. In this case, there exists $k \in \mathbb{N}_{>0}$ such that $B = \mathsf{Chain}_k$, and we proceed by induction on k. In the initialisation, $B = D_1$ is the 1-dimensional disk, and the map $\sigma : \mathbb{D}^1 \to X$ corresponds by the Yoneda lemma to a cell $x \in X$, which by assumption is invertible, and we then choose $\bar{\sigma} = x^{-1} : \mathbb{D}^1 \to X$. For the iterative case, $B = \mathsf{Chain}_{k+1} = \mathsf{Chain}_k *_0 D_1$, and we have a map $\langle \sigma, x \rangle : \mathsf{Pos}(\mathsf{Chain}_k *_0 D_1) \to X$, with σ a map sending all maximal positions to invertible cells and x an invertible cell. We choose the pointwise inverse to be $\langle x^{-1}, \bar{\sigma} \rangle : \mathsf{Pos}(D_1 *_0 \mathsf{Chain}_k) \to X$. This concludes the induction for the initialisation case of a Batanin tree of dimension 1. Consider now a Batanin tree B of dimension $d+1$, we have $B = \mathsf{br}[B_1, \ldots, B_n]$ and a map $\sigma : \mathsf{Pos}(B) \to X$ decomposes as $\langle \Sigma\, \sigma_1, \ldots, \Sigma\, \sigma_n \rangle$ with $\sigma_i : \mathsf{Pos}(B_i) \to \Omega\, X$. We then define σ'_i to be either $\bar{\sigma}_i$ constructed by induction if $\mathsf{dim}(B_i) = d$ or σ_i otherwise. This lets us define $\bar{\sigma} = \langle \Sigma\, \sigma'_1, \ldots, \Sigma\, \sigma'_n \rangle$. □

5.3 Invertibility of composition

From now on, we consider an ω-category $\mathbb{X} = (X, \alpha : TX \to X)$. Our aim is to show that a composite of invertible cells in $\mathbb{X}$ is itself invertible, we start by making this statement precise. Given a set of cells A of $\mathbf{X} = \sqcup X_n$, we define the set $C(A)$ of composites of cells of A to be the set of all cells of the form

$$\mathsf{coh}^{\mathbb{X}}(B, T(s^B_{d-1})(a) \to T(t^B_{d-1})(b), \sigma),$$

where B is a Batanin tree of dimension d, the cells $a, b \in (T\,\mathsf{Pos}(\partial_{d-1} B))$ are cover $\partial_{d-1} B$ and $\sigma : \mathsf{Pos}(B) \to X$ is a map sending maximal positions of B onto cells in A. Given the set W of invertible cells in $\mathbb{X}$, the set of composites of invertible cells is the set $C(W)$, and we prove here that every cell in $C(W)$ is invertible. For the coinductive hypothesis, we need a slightly stronger statement, namely that iterated composited of invertible cells are invertible. To set up the notation, define

$$\begin{cases} W_0 = W \\ W_{n+1} = C\left(W_{\leq k}\right) \end{cases} \qquad W_{\leq n} = \bigcup_{k \leq n} W_k \qquad W_\infty = \bigcup_n W_n.$$

The set W_∞ is the set of iterated composites of invertible cells.

Theorem 33. *All cells in W_∞ are invertible, or in other words, $W_\infty \subset W$.*

Proof. By coinduction, it suffices to show that given a cell $c \in W_\infty$, there exists cells $c^{-1} \in \mathbf{X}$ and $\mathsf{u}_c, \mathsf{v}_c \in W_\infty$ satisfying the correct boundary conditions. Indeed, this translates into W_∞ being a postfixed point of the function defining W, thus it implies $W_\infty \subset W$. We prove the required statement by induction on the natural number n such that $c \in W_n$. The base case $n = 0$ follows immediately from the definition of the set W as a postfixed point. Additionnally, if $c \in W$, so is c^{-1}, by choosing $\left(c^{-1}\right)^{-1} = c$. The rest of this section is dedicated to a construction that proves the inductive case, which can be summarised into the following:

Claim. *Given $n \in \mathbb{N}$, assume that for every $k \leq n$ and every cell $c \in W_k$, the cells $c^{-1} \in X$ and $\mathsf{u}_c, \mathsf{v}_c \in W_\infty$, and that we have $(c^{-1})^{-1} = c$. Then for any $c \in W_{n+1}$, there exist cells $c^{-1} \in X$ and $\mathsf{u}_c, \mathsf{v}_c \in W_\infty$, and $(c^{-1})^{-1} = c$.*

For the rest of this proof, we fix a number $n \in \mathbb{N}$ and we assume the following inductive hypothesis: Given any cell $x \in W_k$ with $k \leq n$, the cells $x^{-1} \in X$ and $\mathsf{u}_x, \mathsf{v}_x \in W_\infty$ have been constructed. We consider a cell $c \in W_{n+1}$, that we can write as

$$c = \mathsf{coh}^{\mathbb{X}}(B, T(s^B_{d-1})(a) \to T(t^B_{d-1})(b), \sigma)$$

where B is a Batanin tree of dimensions d, $a, b \in T\,\mathsf{Pos}(\partial_{d-1}B)$ are covering cells and σ sends all maximal positions of B onto cells in $W_{\leq n}$. We construct the cells c^{-1}, u_c and v_c. To simplify the notations, we introduce the cell $c_0 = \mathsf{coh}(B, T(s^B_{d-1})(a) \to T(t^B_{d-1})(b), \mathsf{id}_{\mathsf{Free}\,\mathsf{Pos}(B)}) \in T\,\mathsf{Pos}(B)$, in such a way that $c = (c_0)^{\mathbb{X}}(\sigma)$. Moreover, we write $\mathsf{op}\,B$ instead of $\mathsf{op}_{\{d\}}\,B$ and $B * B$ instead of $B *_{d-1} B$.

By the inductive hypothesis, for any maximal position $p \in \mathsf{Pos}(B)$, the cell $\sigma_V(p) \in W_{\leq n}$ admits an inverse. Therefore Lemma 32 lets us define the map $\bar{\sigma} : \mathsf{Pos}(\mathsf{op}\,B) \to X$ and then choose c^{-1} as follows:

$$c^{-1} = \mathsf{coh}^{\mathbb{X}}(B, T(s^B_{d-1})(b) \to T(t^B_{d-1})(a), \bar{\sigma}).$$

We then check using Lemma 32 that the cell c^{-1} constructed this way has the expected source and boundary.

$$\begin{aligned}\mathsf{src}(c^{-1}) &= \alpha \circ T(\bar{\sigma})(T(s^B_{d-1})(b)) = \alpha(T(t^B_{d-1})(b)) = \mathsf{tgt}(c)\\ \mathsf{tgt}(c^{-1}) &= \alpha \circ T(\bar{\sigma})(T(t^B_{d-1})(a)) = \alpha(T(s^B_{d-1})(b)) = \mathsf{src}(c)\end{aligned}$$

Moreover, the equality $(c^{-1})^{-1} = c$ can be verified by induction. Additionally The counit v_c is can be obtained for free, if we can construct $\mathsf{u}_{c^{-1}}$. Indeed, $c^{-1} \in W_{n+1}$, satisfies $(c^{-1})^{-1} = c$, hence, we choose $\mathsf{v}_c = \mathsf{u}_{c^{-1}}$. So to conclude the proof, it suffices to construct the cell u_c.

We construct the cell $\mathsf{u}_c \in W_\infty$ as a composite of three different cells as follows:

$$\mathsf{u}_c = m^1 *_{n+1} m^2 *_{n+1} m^3.$$

In such a way that we have

$$\mathsf{src}(\mathsf{u}_c) = c *_n c^{-1} \qquad\qquad \mathsf{tgt}(\mathsf{u}_c) = \mathsf{id}(\mathsf{src}\,c).$$

So it suffices to define the three cells m^1, m^2 and m^3. We define these cells as three different steps.

Associator. The first step in constructing u_c consists in reassociating the binary composite of c and c^{-1}. In the target, all the maximal cells that are composed in codimension 1 are associated together, and the result of those compositions are composed in an unbiased way. First, we define the filler cell

$$m_0^1 = \mathsf{fill}(B_i, f_i, a_i \to b_i, \sigma_i),$$

with $B_1 = D_d *_{d-1} D_d$, the map $f_1 = \langle B, \mathsf{op}\, B\rangle$, the sphere $a_1 \to b_1$ is the full sphere defining the unbiased composite of B_1, and $\sigma_1 = \langle c_0, {c_0}^{-1}\rangle$; as well as $B_2 = (B *_{d-1} B)^{\mathsf{red}}$, f_2 is the map that sends any maximal position p onto $\mathsf{Chain}_{\mathsf{length}(p)}$, the sphere $a_2 \to b_2$ is given by $a_2 = T(s^{(B*_{d-1}B)^{\mathsf{red}}})(a)$ and $b_2 = T(t^{(B*_{d-1}B)^{\mathsf{red}}})(a)$, and the map $\sigma_2 = \mathsf{red}_{B*_{d-1}B}$. One can check that $\mu^{\mathsf{str}}(f_1) = \mu^{\mathsf{str}}(f_2) = B *_{d-1} B$. Moreover, by Lemma 29, we can compute

$$\mathsf{Cell}(\mathsf{red}_{B*B}) \circ T(s_{d-1}^{B*B^{\mathsf{red}}})(a) = T(s_{d-1}^{B*B})(a) = \mathsf{src}(c_0)$$
$$\mathsf{Cell}(\mathsf{red}_{B*B}) \circ T(t_{d-1}^{B*B^{\mathsf{red}}})(a) = T(t_{d-1}^{B*B})(a) = \mathsf{tgt}({c_0}^{-1}).$$

This justifies that the filler m_0^1 is well defined. The source and target are given by

$$\mathsf{src}(m_0^1) = c_0 *_{d-1} {c_0}^{-1}$$
$$\mathsf{tgt}(m_0^1) = \mathsf{coh}((B*B)^{\mathsf{red}}, T(s_{d-1}^{B*B^{\mathsf{red}}})(a) \to T(t_{d-1}^{B*B^{\mathsf{red}}})(a), \mathsf{red}_{B*B})$$

We then define $m^1 = (m_0^1)^{\mathbb{X}}(\langle\sigma,\bar\sigma\rangle)$, in such a way that it satisfies

$$\mathsf{src}(m^1) = c *_{d-1} c^{-1}$$
$$\mathsf{tgt}(m^1) = \mathsf{coh}^{\mathbb{X}}((B*B)^{\mathsf{red}}, T(s_{d-1}^{(B*B)^{\mathsf{red}}})(a) \to T(t_{d-1}^{(B*B)^{\mathsf{red}}})(a), \mathsf{red}_{B*B})(\langle\sigma,\bar\sigma\rangle).$$

Telescope cancellation. Our next step consists in cancelling away the codimension 1 composite of maximal cells into identities, using the telescopes. To construct the cell that preforms all telescope operation at the same time, we use the functorialisation of the reduced coherence cell with respect to all its maximal arguments. Denote M the set of maximal positions of the tree $(B*B)^{\mathsf{red}}$. We define the maps

$$\tau_- : \mathsf{Pos}((B*B)^{\mathsf{red}}) \to X \qquad \tau_+ : \mathsf{Pos}((B*B)^{\mathsf{red}}) \to X$$
$$\tau_- = \alpha \circ T(\langle\sigma,\bar\sigma\rangle) \circ \mathsf{Cell}(\mathsf{red}_{B*B}) \qquad \tau_+ = \alpha \circ T(\langle\sigma,\bar\sigma\rangle \circ s_{d-1}^{B*B}) \circ \mathsf{Cell}(\underline{\mathsf{id}}_{B*B}).$$

Moreover, for every position p of $(B*B)^{\mathsf{red}}$, we define a cell τ_p such that $\mathsf{src}(\tau_p) = \tau_-(p)$ and $\mathsf{tgt}(\tau_p) = \tau_+(p)$, in such a way that applying Lemma 25 shows that τ_- and τ_+ together with (τ_p) define a map

$$\tau\colon\ \mathsf{Pos}(((B*B)^{\mathsf{red}}) \uparrow X) \to X$$

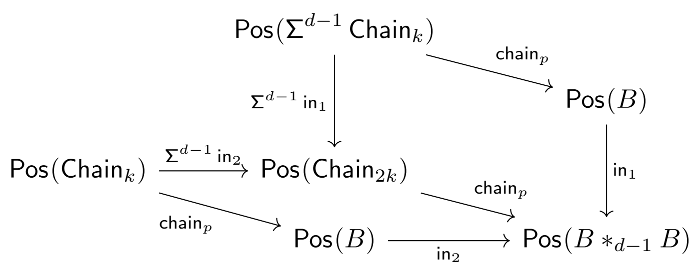
$$\tau \circ s_{d-1}^{(B*B)^{\mathsf{red}}} = \tau_- \qquad \tau \circ t_{d-1}^{(B*B)^{\mathsf{red}}} = \tau_+.$$

To construct the cell τ_p, we apply Lemma 30: Denoting $k = \mathsf{length}_B(p)$, we have $\mathsf{length}_{B*B}(p) = 2k$. For each $1 \leq i \leq k$, the position f_i^{chain} satisfies by assumption, $\mathsf{Cell}(\sigma)(\mathsf{chain}_p(\Sigma^{d-1} f_i^{\mathsf{Chain}})) \in W_{\leq n}$, thus we may assume by induction that $\sigma_V(\mathsf{chain}_p(\Sigma^{d-1} f_i^{\mathsf{Chain}}))^{-1} \in X$ and $\eta_{\sigma_V(\mathsf{chain}_p(\Sigma^{d-1} f_i^{\mathsf{Chain}}))} \in W_\infty$ are already constructed. This lets us define the map $\mathsf{can}_p : \Sigma^{d-1} \mathsf{Tel}_k \to X$, by the following assignment:

$$f_i^{\mathsf{Tel}} \mapsto \sigma_V(\mathsf{chain}_p(\Sigma^{d-1} f_i^{\mathsf{Chain}})) \qquad g_i^{\mathsf{Tel}} \mapsto \sigma_V(\mathsf{chain}_p(\Sigma^{d-1} f_i^{\mathsf{Chain}}))^{-1}$$

$$\alpha_i^{\mathsf{Tel}} \mapsto \eta_{\sigma_V(\mathsf{chain}_p(\Sigma^{d-1} f_i^{\mathsf{Chain}}))}.$$

Using Lemma 30, we get the following commutative diagram of globular sets, where the two cospans are the legs of pushout square:



from which we deduce $\langle \sigma, \bar{\sigma} \rangle \circ \mathsf{chain}_p = \langle \sigma \circ \mathsf{chain}_p, \bar{\sigma} \circ \mathsf{chain}_p \rangle$. Moreover, one can show the following equalities, by showing that the involved maps coincide on all locally maximal positions:

$$\mathsf{can}_p \circ \Sigma^{d-1}(\mathsf{loop}_k \circ \mathsf{in}_1) = \sigma \circ \mathsf{chain}_p \quad \mathsf{can}_p \circ \Sigma^{d-1}(\mathsf{loop}_k \circ \mathsf{in}_2) = \bar{\sigma} \circ \mathsf{chain}_p$$

This map lets us define the cell $\tau_p = (\mathsf{tel}_k)^{\mathbb{X}}(\mathsf{can}_p) \in X$, satisfying the following equations:

$$\begin{aligned}
\mathsf{src}(\tau_p) &= \alpha \circ T(\mathsf{can}_p \circ \Sigma^{d-1} \mathsf{loop}_k)(\mathsf{comp}_{\Sigma^{d-1} \mathsf{Chain}_{2k}}) \\
&= \alpha \circ T(\langle \mathsf{can}_p \circ \Sigma^{d-1}(\mathsf{loop}_k \circ \mathsf{in}_1), \mathsf{can}_p \circ \Sigma^{d-1}(\mathsf{loop}_k \circ \mathsf{in}_2) \rangle) \\
&= \alpha \circ T(\langle \sigma, \bar{\sigma} \rangle \circ \mathsf{chain}_p)(\mathsf{comp}_{\Sigma^{d-1}(\mathsf{Chain}_{2k})}) \\
&= \tau_-(p) \\
\mathsf{tgt}(\tau_p) &= \mathsf{id}_{d-1}(\mathsf{src}(\sigma_V(p_1))) = \tau_+(p).
\end{aligned}$$

We now define the cell

$$m^2 = (((B * B)^{\mathsf{red}}, T(s_{d-1}^{(B*B)^{\mathsf{red}}})(a) \to T(t_{d-1}^{(B*B)^{\mathsf{red}}})(a)) \uparrow M)^{\mathbb{X}}(\tau).$$

The source and target of m^3 are given by

$$\begin{aligned}
\mathsf{src}(m^2) &= \mathsf{coh}^{\mathbb{X}}((B * B)^{\mathsf{red}}, T(s_{d-1}^{(B*B)^{\mathsf{red}}})(a) \to T(t_{d-1}^{(B*B)^{\mathsf{red}}})(a), \mathsf{red}_{B*B})(\langle \sigma, \bar{\sigma} \rangle) \\
\mathsf{tgt}(m^2) &= \mathsf{coh}^{\mathbb{X}}(\partial_{d-1} B, a \to a, \underline{\mathsf{id}}_{B*B})(\langle \sigma, \bar{\sigma} \rangle \circ s_{d-1}^{B*B}).
\end{aligned}$$

Unbiased unitor. The final step of our construction consists in the application of an unbiased unitor, which allow us to cancel the composite of identities that we have produced into a single identity. Explicitly, we consider the cells

$$m^3_0 = \mathsf{unitor}((B * B)^{\mathsf{red}}, a) \qquad m^3 = (m^3_0)^{\mathbb{X}}(\langle \sigma, \bar{\sigma} \rangle \circ s^{B*B}_{d-1}).$$

The source and target of m^3 are given by

$$\begin{aligned} \mathsf{src}(m^3) &= \mathsf{coh}^{\mathbb{X}}(\partial_{d-1} B, a \to a, \underline{\mathsf{id}}_{B*B})(\langle \sigma, \bar{\sigma} \rangle \circ s^{B*B}_{d-1}) \\ \mathsf{tgt}(m^3) &= \mathsf{id}((a)^{\mathbb{X}}(\langle \sigma, \bar{\sigma} \rangle \circ s^{B*B}_{d-1})). \end{aligned}$$

This terminates the construction of u_c, which proves the claim and thus concludes the proof of the theorem. □

A consequence of the fact that a composite of invertible cells is invertible is the fact that the relation of equivalence between cells in an ω-category define an equivalence relation

Corollary 34. *The relation $\sim$ is an equivalence relation*

Proof. We have proven that this relation is symmetric (Lemma 15) and reflexive (Corollary 17), so it suffices to prove transitivity. Consider three cells c_1, c_2, c_3 such that $c_1 \sim c_2$ and $c_2 \sim c_3$. There exist two invertible cells x, x' with

$$\mathsf{src}(x) = c_1 \qquad \mathsf{tgt}(x) = c_2 \qquad \mathsf{src}(x') = c_2 \qquad \mathsf{tgt}(x') = c_3.$$

By Theorem 33, the cell $x * x'$ is invertible, it witnesses the relation $c_1 \sim c_3$. □

5.4 Invertible cells of a computad

We now consider the question of the invertibility of cells in an ω-category freely generated by a computad. Proposition 16 and Theorem 33 applied in a computad give a syntactic recognition criterion for invertibility:

Proposition 35. *Consider a cell $c \in \mathsf{Cell}_d(C)$ in a computad C. If $\mathsf{supp}_d(c) = \emptyset$ then the cell c is invertible.*

Proof. We proceed by structural induction on the cell c. When c is a generator, the statement is vacuously true since the support can not be empty. Suppose therefore that $c = \mathsf{coh}(B, A, \sigma)$ and that the result holds for the cell $\sigma_V(p)$ for every $p \in \mathsf{Pos}_d(B)$. If no such position exists, then $\dim B < d$ and c is invertible by Proposition 16. Otherwise, the support of each $\sigma_V(p)$ is empty, so it is invertible. By Theorem 33, we concude then that c is invertible as well. □

This sufficient condition may not be necessary in general, indeed in an arbitrary computad, a generator may be invertible, violating this condition. However, this condition is necessary in finite dimensional computads. To prove this, we rely on the following technical result.

Lemma 36. *Consider a cell $c \in \mathsf{Cell}_d(C)$ in a computad C and a cell c. such that $\mathsf{supp}_d(c) = \emptyset$, then $\mathsf{supp}_{d-1}(\mathsf{src}(c)) = \mathsf{supp}_{d-1}(\mathsf{tgt}(c)) = \mathsf{supp}_{d-1} c$.*

Proof. We proof this result by structural induction on c. Again, when c is a generator the statement is vacuous, because its d-support is not empty. Suppose that $c = \mathsf{coh}((,B,,)A,\sigma)$ and that for every $p \in \mathsf{Pos}()d(B)$ the result holds for the cell $\sigma_V(p)$. If no such cell exists, then $\dim B < d$, so a, b cover $\mathsf{Pos}(B)$, thus we have

$$\begin{aligned}\mathsf{supp}_{d-1}(c) &= \bigcup_{p \in \mathsf{Pos}_{d-1}(B)} \mathsf{supp}_{d-1}(\sigma_V(p)) \\ &= \mathsf{supp}_{d-1}(\mathsf{Cell}(\sigma)(a)) \\ &= \mathsf{supp}_{d-1}(\mathsf{Cell}(\sigma)(b)).\end{aligned}$$

Otherwise, $\dim B = d$, and so $a = T(s_d^B)(a')$ and $b = T(t_d^B)(b')$ with both $a', b' \in T(\mathsf{Pos}(\partial_{d-1}B))$ covering it. We then have:

$$\mathsf{supp}_{d-1}(c) = \bigcup_{p \in \mathsf{Pos}_{d-1}(B)} \mathsf{supp}_{d-1}(\sigma_V(p)) \cup \bigcup_{p \in \mathsf{Pos}_d(B)} \mathsf{supp}_{d-1}(\sigma_V(p)).$$

For every position $p \in \mathsf{Pos}_d(B)$, we have by induction $\mathsf{supp}_{d-1}(\sigma_V(p)) = \mathsf{supp}_{d-1}(\sigma_V(\mathsf{src}(p)))$. Moreover, for every position $p \in \mathsf{Pos}_{d-1}(B)$, there exists a unique position q in the image of s_{d-1}^B parallel to p. This positions are connected by a sequence of positions of dimension d, thus by induction, we have $\mathsf{supp}_{d-1}(p) = \mathsf{supp}_{d-1}(q)$. This proves that we have

$$\mathsf{supp}_{d-1}(c) = \bigcup_{p \in \mathsf{Pos}_{d-1}(\partial_{d-1}B)} \mathsf{supp}_{d-1}(s_{d-1}^B(p)) = \mathsf{supp}(\mathsf{Cell}(\sigma \circ \mathsf{Free}(s_{d-1}^B))(a)).$$

A similar argument shows the result for the target. □

Proposition 37. *In a finite dimensional computad C, a cell $c \in \mathsf{Cell}_d(C)$ is invertible if and only if $\mathsf{supp}_d(c) = \emptyset$.*

Proof. We show that if we have an invertible cell $c \in \mathsf{Cell}_d(C)$ and a generator $x \in \mathsf{supp}_d(c)$, then C is infinite dimensional by coinduction. Indeed, considering such a cell c, we have the cell $\mathsf{u}_c \in \mathsf{Cell}_{d+1}(C)$. Moreover, $x \in \mathsf{supp}_d(\mathsf{src}(\mathsf{u}_c))$ and $\mathsf{supp}_{d-1}(\mathsf{tgt}(\mathsf{u}_c)) = \mathsf{supp}_{d-1}(\mathsf{tgt}(c)) = \emptyset$. By Lemma 36, this implies that there exists $\mathsf{supp}_{d+1}(\mathsf{u}_c) \neq \emptyset$. Since moreover u_c is invertible, this implies by coinduction that C is infinite dimensional. □

6 Implementation in CaTT

The description of computads that we work with in this article can be equivalently formulated as a dependent type theory called catt, introduced by Finster and Mimram [14]. In fact the formulation of computads proposed by Dean et al. [13] was heavily influenced by this dependent type theory, and the syntactic

	LoC	LoC (ratio)	declarations	declarations (ratio)
vanilla	531	+0%	93	+0%
`s`	470	−11.5%	85	−8.6%
`sf`	397	−25.2%	76	−18.2%
`sfb`	378	−28.8%	70	−24.7%
`sfbi`	73	−86.2%	16	−82.8%

Figure 1: Mechanisation of the unit of the Eckmann-Hilton cell.

equivalence between the two was proved by the two authors of the paper and Sarti [8]. An implementation of a typechecker for the dependent type theory `catt` is avaiblable and maintained by the first author[2]. We have integrated the work presented in this article to the implementation, in such a way that given a term in the theory whose variables are all of dimension lower than the term, one can automatically compute its inverse or a witness of equivalence for this term. In practice, the user has defined a term `t`, which happens to be invertible, they can access to the chosen inverse computed by our algorithm by inputting `I(t)`, and they can access the chosen witness of equivalence by inputting `U(t)`.

This allows for improved mechanisation of terms in `catt`, complementing the suspension and the functorialisation of terms that were already implemented. We have assessed the relevance of this mechanisation principle on a practical example: the definition of the term corresponding to the Eckmann-Hilton cell, as well as its inverse and the witness that these two cancel each other. The choice of this particular example is motivated by several considerations: They are important examples in higher category theory, due to their connection with topology and homotopy theory. These cells are complicated enough to define the simplification will be significant on it, and is not definable in a pasting scheme, making the mechanisation non-trivial. Yet they are among the simplest examples with this property, and are reasonable to define even without this mechanisation principle, making them a very good example.

We have formalised the Echmann-Hilton and its inverse as well as the unit cell for the equivalence, and verified them in `catt` using various levels of mechanisation. The results are compiled in Figure 1, and the files that we used to assess these are available under the `examples/eckmann-hilton-versions/` directory of the repository. To assess the complexity of a file, we use as proxies the number of lines of code written in the file and the number of individual declaration the file has. The number of declaration fails to account for the the complexity of each of the declaration, while the number of lines of codes magnifies this parameter by accounting for line skips for code formatting and comments. Overall, these two proxies together provide a reasonable proxy for the complexity of the definitions. The levels of mechanisation that we consider are cumulative: “vanilla” has no automation, and then `s` inducates that the suspension is used, `f` inducates that the functorialisation is used, `b` indicates that the compositions and identities are taken as built-ins and `i` indicates that inverses and witnesses of composition are

[2] `http://www.github.com/thibautbenjamin/catt`

computed automatically using inverses. The ratios are always considered against the vanilla case with no machanisation at all. Both of the chosen metrics indicate that the mechanisation of inverses is by far the most efficient mechanisation principle that we have defined for this example. It is also by far the most intricate of those mechanisation principle. Since our example consists in defining a cell, its inverse, and the unit of the inversion, one would expect that adding the mechanisation of the inverses to divide the size by 3. However, comparing `sfbi` and `sfb`, we observes a diminution of 80% in the number of lines of code and of 77% of the number of declarations. This can be explained by the fact that the unit cell is more complex than both the Eckmann-Hilton cell and its inverse, and also by the fact that the inverse mechanisation also allows for simplification in the definition of the Eckmann-Hilton cell itself.

References

[1] Thorsten Altenkirch and Ondrej Rypacek. A syntactical approach to weak omega-groupoids. page 15 pages. doi:10.4230/LIPICS.CSL.2012.16.

[2] Dimitri Ara. Sur les ∞-groupoïdes de grothendieck et une variante ∞-catégorique. URL: https://www.i2m.univ-amu.fr/perso/dimitri.ara/files/these.pdf.

[3] Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries, and Jamie Vicary. Modular categories as representations of the 3-dimensional bordism 2-category. URL: http://arxiv.org/abs/1509.06811, arXiv:1509.06811.

[4] Michael A. Batanin. Monoidal globular categories as a natural environment for the theory of weak n-categories. 136(1):39–103. doi:10.1006/aima.1998.1724.

[5] Thibaut Benjamin. A type theoretic approach to weak ω-categories and related higher structures. URL: https://theses.hal.science/tel-03106197.

[6] Thibaut Benjamin, Eric Finster, and Samuel Mimram. Globular weak ω-categories as models of a type theory. doi:10.48550/ARXIV.2106.04475.

[7] Thibaut Benjamin and Ioannis Markakis. Opposites of weak ω-categories and the suspension and hom adjunction. doi:10.48550/ARXIV.2402.01611.

[8] Thibaut Benjamin, Ioannis Markakis, and Chiara Sarti. CaTT contexts are finite computads. URL: https://arxiv.org/abs/2405.00398, arXiv:2405.00398.

[9] Clemens Berger. A cellular nerve for higher categories. 169(1):118–175. doi:10.1006/aima.2001.2056.

[10] John Bourke. Iterated algebraic injectivity and the faithfulness conjecture. 4(2):183–210. doi:10.21136/HS.2020.13.

[11] Eugenia Cheng. An ω-category with all duals is an ω-groupoid. 15(4):439–453. doi:10.1007/s10485-007-9081-8.

[12] Eugenia Cheng and Aaron Lauda. Higher-dimensional categories: an illustrated guide book. URL: https://eugeniacheng.com/wp-content/uploads/2017/02/cheng-lauda-guidebook.pdf.

[13] Christopher J. Dean, Eric Finster, Ioannis Markakis, David Reutter, and Jamie Vicary. Computads for weak ω-categories as an inductive type. URL: http://arxiv.org/abs/2208.08719, arXiv:2208.08719.

[14] Eric Finster and Samuel Mimram. A type-theoretical definition of weak ω-categories. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–12. ACM. arXiv:1706.02866, doi:10.5555/3329995.3330059.

[15] Soichiro Fujii, Keisuke Hoshino, and Yuki Maehara. Weakly invertible cells in a weak ω-category. doi:10.48550/ARXIV.2303.14907.

[16] Alexander Grothendieck. Pursuing stacks. URL: http://arxiv.org/abs/2111.01000, arXiv:2111.01000.

[17] Simon Henry. Algebraic models of homotopy types and the homotopy hypothesis. URL: http://arxiv.org/abs/1609.04622, arXiv:1609.04622.

[18] Simon Henry and Edoardo Lanari. On the homotopy hypothesis in dimension 3. 39:735–768. arXiv:1905.05625.

[19] Yves Lafont and François Métayer. Polygraphic resolutions and homology of monoids. 213(6):947–968. doi:10.1016/j.jpaa.2008.10.005.

[20] Yves Lafont, François Métayer, and Krzysztof Worytkiewicz. A folk model structure on omega-cat. 224(3):1183–1231. arXiv:0712.0617, doi:10.1016/j.aim.2010.01.007.

[21] Tom Leinster. *Higher operads, higher categories*. Number 298 in London Mathematical Society lecture note series. Cambridge University Press. URL: https://arxiv.org/abs/math/0305049, arXiv:math/0305049.

[22] Tom Leinster. A survey of definitions of n-category. URL: https://arxiv.org/abs/math/0107188, arXiv:math/0107188.

[23] Peter LeFanu Lumsdaine. Weak ω-categories from intensional type theory. In Pierre-Louis Curien, editor, *Typed Lambda Calculi and Applications*, volume 5608, pages 172–187. Springer Berlin Heidelberg. doi:10.1007/978-3-642-02273-9_14.

[24] Georges Maltsiniotis. Grothendieck ∞-groupoids, and still another definition of ∞-categories. URL: http://arxiv.org/abs/1009.2331, arXiv:1009.2331.

[25] Ioannis Markakis. Computads for generalised signatures. 228(9):107675. doi:10.1016/j.jpaa.2024.107675.

[26] Samuel Mimram. Towards 3-dimensional rewriting theory. 10(2):1. arXiv:1403.4094, doi:10.2168/LMCS-10(2:1)2014.

[27] Alex Rice. Coinductive Invertibility in Higher Categories. URL: http://arxiv.org/abs/2008.10307, arXiv:2008.10307.

[28] Craig C Squier. Word problems and a homological finiteness condition for monoids. *Journal of Pure and Applied Algebra*, 49(1-2):201–217, 1987.

[29] Ross Street. The petit topos of globular sets. 154(1-3):299–315. doi:10.1016/S0022-4049(99)00183-8.

[30] Benno van den Berg and Richard Garner. Types are weak ω-groupoids. 102(2):370–394. arXiv:0812.0298, doi:10.1112/plms/pdq026.

[31] Mark Weber. Generic morphisms, parametric representations and weakly cartesian monads. 13:191–234. URL: https://eudml.org/doc/124614.