Strictly Associative and Unital ∞-Categories as a Generalized Algebraic Theory

Eric Finster*, Alex Rice†, and Jamie Vicary‡

April 7, 2024

Abstract

We present the first definition of strictly associative and unital ∞-category. Our proposal takes the form of a generalized algebraic theory, with operations that give the composition and coherence laws, and equations encoding the strict associative and unital structure. The key technical idea of the paper is an equality generator called insertion, which can "insert" an argument context into the head context, simplifying the syntax of a term. The equational theory is defined by a reduction relation, and we study its properties in detail, showing that it yields a decision procedure for equality.

Expressed as a type theory, our model is well-adapted for generating and verifying efficient proofs of higher categorical statements. We illustrate this via an OCaml implementation, and give a number of examples, including a short encoding of the syllepsis, a 5-dimensional homotopy that plays an important role in the homotopy groups of spheres.

1 Introduction

Background. The theory of higher categories has growing importance in computer science, mathematics, and physics, with fundamental applications now recognized in type theory [9, 19], quantum field theory [1, 14], and geometry [12]. Its relevance for logic was made abundantly clear by Hoffman and Streicher [9], whose groupoid model of Martin-Löf type theory violated the principle of *uniqueness of identity proofs* (UIP). This paved the way to the modern study of proof-relevant logical systems, in which one can reason about proofs themselves as fundamental objects, just as one may study the homotopy theory of paths of a topological space.

While this *proofs-as-paths* perspective yields considerable additional power, a key drawback is the extensive additional proof obligations which can arise when working with the resulting path types. These are commonly understood to organize into three classes: the *associator* witnessing that for three composable paths f, g, h, the compositions $f \circ (g \circ h)$ and $(f \circ g) \circ h$ are equivalent; the *unitors* witnessing that $\mathrm{id} \circ f$ and $f \circ \mathrm{id}$

*University of Birmingham, `e.l.finster@bham.cs.ac.uk`
†University of Cambridge, `alex.rice@cl.cam.ac.uk`
‡University of Cambridge, `jamie.vicary@cl.cam.ac.uk`

should both be equivalent to f itself; and the *interchanger* witnessing that compositions in different dimensions should commute. Worse, such path witnesses themselves admit further constraints, such as the *pentagon condition* for associators, for which further higher-dimensional witnesses must be computed, compounding the problem exponentially in higher dimensions.

The need to construct and manage such witnesses (also called *weak structure*) can complicate a path-relevant proof, potentially beyond the point of tractability. Indeed an explicit proof of the *syllepsis*, an important homotopy from low-dimensional topology which is entirely formed from such path witnesses, was formalized only recently [5, 16]. This motivates the search for a syntax that can trivialize witnesses as far as possible, an effort which has been underway in the mathematics community since at least the 1970s. An early realization was that in the fully strict case, where all witnesses are trivialized, too much expressivity is lost, with Grothendieck being one of the first to observe this [8, page 2].

The focus therefore turned to identifying a *semistrict* theory, where as many witnesses as possible are trivialized, while retaining equivalence with the fully weak theory. An early contribution by Gray [7] yielded a definition of semistrict 3-category with trivial associators and unitors, leaving only the interchangers nontrivial; later work by Gordon, Power and Street showed that this definition loses no expressibility [6]. Simpson conjectured that this could be extended to n-categories [15], and Street had sketched a possible approach based on iterated enrichment, where a closed monoidal category of semistrict n-categories is defined in each dimension [17]; however, an analysis by Dolan [3] indicated that the required monoidal closure properties could not be fulfilled.

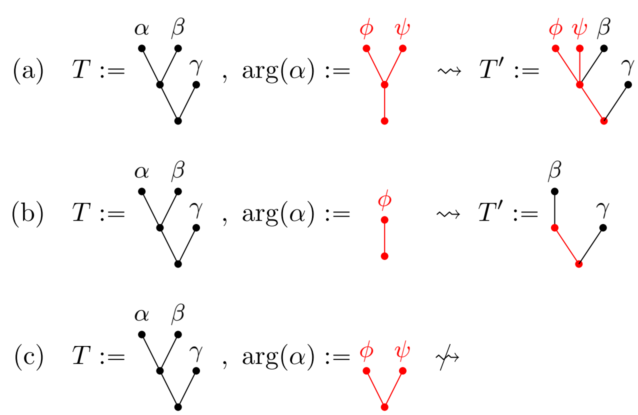


Figure 1: Illustrating the insertion operation.

Contribution. We present a generalized algebraic theory whose models are ∞-categories with strict associators and unitors, with only the interchanger structure remaining weak. To our knowledge this is the first such definition to have been presented in the literature. In this theory the operations give the compositional structure and coherence laws of the ∞-category, while the equational theory trivializes the associator and unitor structure.

For example, the equation $(f \circ g) \circ h = f \circ (g \circ h)$ is derivable in our theory.

Our proposal is implemented as a type theory, and the type checker can apply such equalities automatically, without requiring the user to specify explicit associator witnesses. This allows the user to work directly with the semistrict theory, building proofs which neglect associator and unitor witnesses and their higher-dimensional counterparts, with the computer nonetheless able to verify correctness of the resulting terms. For example, this would enable the user to directly compose higher paths of type $p \Rightarrow (f \circ g) \circ h$ and $f \circ (g \circ h) \Rightarrow q$, even though it seems they should not be directly composable.

Alternatively, given a proof which includes all such explicit witnesses, the system can "compute them out", returning a normal form where associator and unitor structure has been eliminated as far as possible. This has the potential to yield a simpler proof.

Our key technical contribution is an *insertion* procedure which generates the semistrict behaviour of our equality relation, and which we show satisfies a simple universal property. Any operation in our theory is defined with respect to a *pasting context*, an arrangement of discs that governs the geometry of the composition; combinatorially, these correspond to finite planar rooted trees. Any operation then has a head tree, with arguments corresponding to the leaves of the tree. We require two simple definitions: for a rooted tree, its *trunk height* is the height of the "tree trunk"; and for any leaf, its *branch height* is the height of the highest branch point sitting below it. For a compound operation, where arguments of the head tree are themselves assigned further operations, insertion operates as follows, for some chosen leaf α of the head tree: if the branch height of α is at most the trunk height of the argument tree at α, and if the argument operation is in "standard form", we can insert the argument tree into the head tree.

We illustrate this in Figure 1, using Greek letters to label the leaves. In each part, the head tree T has an argument α of branch height 1 (since descending from α, the first branch point we encounter is at height 1.) In part (a) the argument tree drawn in red has trunk height 1, and so the height test is satisfied and insertion can proceed, yielding an updated head tree T' in which the entire argument tree has been inserted and is now visible in red as a subtree, replacing the original leaf α. In part (b) the argument tree also has trunk height 1, and so the insertion can again proceed, although in this case, due to the short stature of the argument tree, the effect is simply to remove the leaf α of the head tree. In part (c) the trunk height of the argument tree is 0, so the height test is not satisfied and insertion cannot proceed. This last example makes it intuitively clear why the height test is important: since the trunk height of the argument tree is so small, there would be no reasonable way to insert the argument tree into the head tree without disrupting the leaf β.

We combine insertion with two other simpler procedures, *disc removal* and *endo-coherence removal*, to produce our reduction relation on operations of the theory. Our major technical results concern the behaviour of this reduction relation. We show that it terminates, and that it has unique normal forms, yielding a decision procedure for equality.

Implementation. We have produced an OCaml implementation of our scheme as a type theory, which we call $\mathsf{Catt}_{\mathsf{sua}}$, standing for "categorical type theory with strict unitors and associators". In Section 5 we give three worked examples, showing that the triangle and pentagon conditions of the definition of monoidal category completely trivialize, and demonstrating that a known proof of the 5-dimensional syllepsis homotopy reduces to a substantially simpler form.

Formalisation. All technical results in this paper have been formalised in Agda and is available at:

`https://github.com/alexarice/catt-agda/releases/tag/v0.1`

The formalisation compiles with Agda version 2.6.3 and standard library version 1.7.2. As well as the formalisations for theorems that appear in this paper, the repository contains a collection of proofs about various different versions of Catt, which is achieved by formalising the meta theory for a version of Catt with parameterised equality rules, which is presented in Section 2.5 of this paper.

Related Work. Our work is based on the theory of *contractible ∞-categories*, a well-studied model of globular ∞-categories originally due to Maltsiniotis [13], who was building on an early algebraic definition of ∞-groupoid by Grothendieck [8]. An excellent modern presentation is given by Leinster in terms of contractible globular operads [10].

Our algebraic theory builds on an existing type theory Catt for contractible ∞-categories presented at LICS 2017 [4], and its extension $\mathsf{Catt}_{\mathsf{su}}$ presented at LICS 2021 [5] which describes the strictly unital case. That work includes a reduction relation called *pruning*, which removes a single leaf variable from a pasting context. Here the pruning operation is replaced by insertion, which includes pruning as a special case, but which in general performs far more radical surgery on the head context. As a result, the termination and confluence properties here are significantly more complex to establish. Our termination proof uses new techniques that quantify the syntactic complexity of a term, while our confluence proof must analyze many additional critical pairs, some of which are fundamentally more complex than those handled previously. To allow surgery on pasting contexts we also require a different presentation of contexts in terms of trees, which changes many aspects of the formal development. Furthermore, unlike the LICS 2021 paper, all technical lemmas here are formalized, putting the work on a stronger foundation.

Above we discussed the problem faced by Street's historical approach to defining semistrict n-categories. While his approach aimed to build up the semistrict theory one dimension at a time, we retain all the operations of the fully weak structure, instead obtaining semistrictness via a reduction relation on operations. We do not require the explicit definition of a tensor product on semistrict n-categories, nor do we require any closure properties. As a result Dolan's critique does not apply.

2 The Type Theory Catt

In this section we recall the type theory Catt and some of its basic properties. We generalize the original presentation [4] by parameterizing the theory over a given set of equality rules, thus allowing us to prove some general structural properties generically, and allowing future investigations into strictness results to build on the theory developed here. We will then specialize to the theory $\mathsf{Catt}_{\mathsf{sua}}$, which is the focus of the present work.

Special cases of this general framework include the original presentation of Catt by Finster and Mimram [4] in which the set of equality rules is empty, as well as the theory $\mathsf{Catt}_{\mathsf{su}}$ [5] whose rules we recall below in the present framework.

2.1 Syntax for Catt

Catt has 4 classes of syntax: contexts, substitutions, types, and terms; the rules for each can be found in Fig. 2. We parameterise substitutions, types and terms by their context of definition in order to avoid issues with undefined variables and write Term_Γ for a term in context Γ, Type_Γ for a type in context Γ, and $\sigma : \Gamma \to \Delta$ for a substitution from Γ to Δ.

We let $\equiv$ denote syntactic equality up to alpha renaming. Our presentation will be in terms of named variables to improve readability, though in practice any ambiguity introduced by this choice can be avoided by the use of de Brujin indices.

$$\frac{}{\varnothing : \mathsf{Ctx}} \qquad \frac{\Gamma : \mathsf{Ctx} \qquad A : \mathsf{Type}_\Gamma}{\Gamma, (x : A) : \mathsf{Ctx}}$$

$$\frac{}{\langle\rangle : \varnothing \to \Gamma} \qquad \frac{\sigma : \Delta \to \Gamma \qquad t : \mathsf{Term}_\Gamma}{\langle \sigma, x \mapsto t\rangle : (\Delta, x : A) \to \Gamma}$$

$$\frac{}{\star : \mathsf{Type}_\Gamma} \qquad \frac{A : \mathsf{Type}_\Gamma \qquad s : \mathsf{Term}_\Gamma \qquad t : \mathsf{Term}_\Gamma}{s \to_A t : \mathsf{Type}_\Gamma}$$

$$\frac{x \in \Gamma}{x : \mathsf{Term}_\Gamma} \qquad \frac{\Delta : \mathsf{Ctx} \qquad A : \mathsf{Type}_\Delta \qquad \sigma : \Delta \to \Gamma}{\mathsf{coh}\ (\Delta : A)[\sigma] : \mathsf{Term}_\Gamma}$$

Figure 2: Syntax constructions in Catt.

A substitution $\sigma : \Delta \to \Gamma$ maps variables of context Δ to terms of context Γ. For any $t : \mathsf{Term}_\Delta$, $A : \mathsf{Type}_\Delta$, and $\tau : \Theta \to \Delta$, one defines semantic substitution operations

$$t[\![\sigma]\!] : \mathsf{Term}_\Gamma \qquad A[\![\sigma]\!] : \mathsf{Type}_\Gamma \qquad \tau \circ \sigma : \Theta \to \Gamma$$

by mutual recursion on types, terms and substitutions (as in [4]). Note the use of doubled brackets $[\![-]\!]$ to denote this operation, which we are careful to distinguish from the single brackets $[-]$ which are part of the syntax of the coherence constructor. Every context has an identity substitution $\mathrm{id}_\Gamma : \Gamma \to \Gamma$ which maps each variable to itself and the operation of substitution is associative and unital so that the collection of contexts and substitutions forms a category which we will denote Catt as an abuse of notation.

The set of free variables for each syntactic class is defined in a standard way by induction. Given a context Γ and set of variables $S \subseteq \Gamma$, we say that S is *downwards closed* when for all $x : A \in \Gamma$, $x \in S$ implies that the free variables of A are a subset of S. For any set $S \subseteq \Gamma$ we can form its *downward closure*. We can then define the support of a piece of syntax, which intuitively is the set of variables the syntax depends on.

Definition 1. Given a term $t : \mathsf{Term}_\Gamma$, the *support* of t, $\mathsf{supp}(t)$, is the downwards closure of the free variables of t in Γ. The support of a type or substitution is defined similarly.

Example 2. Consider the context $\Gamma = x : \star, y : \star, f : x \to_\star y$. Then the variable f is a valid term in this context whose set of free variables is simply the singleton set $\{f\}$. The support of f, however, is $\{x, y, f\}$.

Lastly we define the dimension of a type A, $\dim(A)$, by $\dim(\star) = 0$ and $\dim(s \to_A t) = 1 + \dim(A)$. The dimension of a term $t : \mathsf{Term}_\Gamma$ is given by $\dim(x) = \dim(A)$ when $x : A \in \Gamma$ and $\dim(\mathsf{coh}\ (\Delta : A)[\sigma]) = \dim(A)$. The dimension of a context $\dim(\Gamma)$ is the maximum of the dimension of the types it contains. One proves easily by induction that dimension is preserved by substitution.

2.2 Typing for Catt

The coherences of Catt are determined by a special class of contexts which we refer to as *pasting contexts*. These correspond intuitively to configurations of globular cells which should admit a unique composition. In particular, for a coherence term $\mathsf{coh}\ (\Delta : A)[\sigma]$ to be well typed it is necessary that Δ is a pasting context. Determining whether an arbitrary context is a pasting context is decidable and we write the judgment

$$\Delta \vdash_p$$

when Δ is a pasting context.

Crucial for the typing rules of Catt is the fact that every pasting context Δ has a well defined *boundary*, which is again a pasting context. For a natural number n, we can construct ∂^n_Δ, the n-dimensional boundary of Δ, and there are inclusion substitutions $\delta^{\epsilon,n}_\Delta : \partial^n_\Delta \to \Delta$ for each n and $\epsilon \in \{-,+\}$.

Rules for pasting contexts as well as a definition of the boundary operators can be found in [4]. We will give an alternative description of pasting contexts as trees in later in Section 2.4.

$$\frac{}{\varnothing \vdash} \qquad \frac{\Gamma \vdash \quad \Gamma \vdash A}{\Gamma, (x : A) \vdash}$$

$$\frac{}{\Gamma \vdash \star} \qquad \frac{\Gamma \vdash A \quad \Gamma \vdash s : A \quad \Gamma \vdash t : A}{\Gamma \vdash s \to_A t}$$

$$\frac{}{\Delta \vdash \langle\rangle : \varnothing} \qquad \frac{\Delta \vdash \sigma : \Gamma \quad \Gamma \vdash A \quad \Delta \vdash t : A[\![\sigma]\!]}{\Delta \vdash \langle \sigma, x \mapsto t \rangle : (\Gamma, x : A)}$$

$$\frac{\Gamma \vdash \quad (x : A) \in \Gamma}{\Gamma \vdash x : A}$$

$$\frac{\begin{array}{c}\Delta \vdash_p \quad \Delta \vdash s \to_A t \quad \Gamma \vdash \sigma : \Delta \\ \mathsf{supp}(s) = \mathsf{supp}(\delta^{-,\dim(\Delta)-1}_\Delta) \quad \mathsf{supp}(t) = \mathsf{supp}(\delta^{+,\dim(\Delta)-1}_\Delta)\end{array}}{\Gamma \vdash \mathsf{coh}\ (\Delta : s \to_A t)[\sigma] : s[\![\sigma]\!] \to_{A[\![\sigma]\!]} t[\![\sigma]\!]}$$

$$\frac{\begin{array}{c}\Delta \vdash_p \quad \Delta \vdash s \to_A t \quad \Gamma \vdash \sigma : \Delta \\ \mathsf{supp}(s) = \mathsf{supp}(t) = \Gamma\end{array}}{\Gamma \vdash \mathsf{coh}\ (\Delta : s \to_A t)[\sigma] : s[\![\sigma]\!] \to_{A[\![\sigma]\!]} t[\![\sigma]\!]}$$

Figure 3: Typing rules for Catt.

With these notions in hand, the typing rules for Catt are given in Fig. 3. All these rules are relatively standard for dependent type theories with the exception of the last two which describe the typing of coherence terms. Both of these latter rules for typing a coherence $\mathsf{coh}\ (\Delta : s \to_A t)[\sigma]$ require that Δ must be a pasting context and that $s \to_A t$ and σ are both well typed. The first rule says that s must be supported by the source of Δ and t by the target of Δ. Terms typed with this rule represent compositions. The second rule instead states that s and t are *full*, they are supported by the whole context. These terms witness that any two full terms over a pasting diagram should be equivalent.

2.3 Constructions and Examples

Disc Contexts. Among the pasting contexts, we may distinguish the *disc contexts* which play an important roll in further constructions. An example of a disc context can be seen in Fig. 4.

Definition 3. The n-dimensional *disc context* D^n has a top-dimensional variable d_n, and variables d_k^-, d_k^+ for each $k < n$. We have $d_0^\pm : \star$, and $d_k^\pm : d_{k-1}^- \to d_{k-1}^+$ for all $0 < k \leqslant n$.

Substitutions out of D^n are special in that they are fully determined by a type of dimension n, and a term of that type. That is, given $A : \mathsf{Type}_\Gamma$ and $t : \mathsf{Term}_\Gamma$, there is a substitution $\{A, t\} : D^{\dim(A)} \to \Gamma$, and any substitution from a disc is of this form.

Unbiased Operations. A given pasting context Γ generally gives rise to many different valid coherences. Among these, there is a particular class of coherences, the *unbiased* ones, which will play an important role in our definitional equality below. Intuitively speaking, these are the coherences which compose all of the cells in a pasting diagram "at once" instead of first composing sub-diagrams. They may be defined recusively as follows:

Definition 4. Given a pasting diagram Δ, we mutually define for all n the *unbiased coherence* $\mathcal{C}_\Delta^n$, the *unbiased term* $\mathcal{T}_\Delta^n$, and the *unbiased type* $\mathcal{U}_\Delta^n$:

$$\begin{aligned}
\mathcal{C}_\Delta^n &= \mathsf{coh}\ (\Delta : \mathcal{U}_\Delta^n)[\mathrm{id}] \\
\mathcal{T}_\Delta^n &= \begin{cases} d_n & \text{when } \Delta \text{ is a disc} \\ \mathcal{C}_\Delta^n & \text{otherwise} \end{cases} \\
\mathcal{U}_\Delta^0 &= \star \\
\mathcal{U}_\Delta^{n+1} &= \mathcal{T}_{\partial_\Delta^n}^n \llbracket \delta_\Delta^{-,n} \rrbracket \to_{\mathcal{U}_\Delta^n} \mathcal{T}_{\partial_\Delta^n}^n \llbracket \delta_\Delta^{+,n} \rrbracket
\end{aligned}$$

The unbiased type takes the unbiased term over each boundary of Δ, includes these all back into Δ and assembles them into a type. When $n = \dim(\Delta)$ we will refer to the unbiased coherence as an *unbiased composite*.

First Examples. We start with some basic examples of categorical operations. The following pasting context contains the composable pair of morphisms $f : x \to y$ and $g : y \to z$:

$$\Gamma = x : \star, y : \star, f : x \to_\star y, z : \star, g : y \to_\star z$$

We can use this to form the composite of f and g, as a term in Γ:

$$f \cdot g := \mathsf{coh}\ (\Gamma : x \to_\star z)[\mathrm{id}] : x \to_\star z$$

This satisfies the support conditions for the first typing rule for coherences, since x is full over the context $x : \star$, and similarly for z. Given a variable t of type $\star$, we can form the identity coherence associated to t (not to be confused with the identity substitution) as follows:

$$\mathbb{1}_t := \mathsf{coh}\ ((x : \star) : x \to_\star x)[\langle x \mapsto t\rangle] : t \to_\star t$$

This can be typed using the second typing rule for coherences.

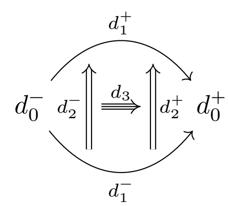


Figure 4: The disc context D^3.

The substitution part of the coherence allows us to form compound operations. For example, the following syntax represents a term in the context $\Delta = x : \star, y : \star, f : x \to_\star y$:

$$f \cdot \mathbb{1}_y := \mathsf{coh}\ (\Gamma : x \to_\star y)[\langle f \mapsto f, g \mapsto \mathbb{1}_y\rangle] : x \to_\star y$$

Here we omit the lower-dimensional components of the substitution as they can be inferred, a technique that we will use repeatedly. Building on this last example we can form the unitor ρ_f witnessing the unitality of $\mathbb{1}$:

$$\rho_f := \mathsf{coh}\ (\Delta : f \cdot \mathbb{1}_y \to_{x \to_\star y} f)[\mathrm{id}] : f \cdot \mathbb{1}_y \to_{x \to_\star y} f$$

and similarly we can generate the associator as a coherence over the context $\Theta = \Gamma, w : \star, h : z \to_\star w$.

$$\begin{aligned}\alpha_{f,g,h} := \mathsf{coh}\ (\Theta : (f \cdot g) \cdot h \to_{x \to_\star w} f \cdot (g \cdot h))[\mathrm{id}] \\ : (f \cdot g) \cdot h \to_{x \to_\star w} f \cdot (g \cdot h)\end{aligned}$$

We note that both the composition and identity examples are in fact examples of unbiased composites as can be seen by straightforward calculuation.

More generally we can define the *identity term* for any n-dimensional term s of type A as follows:

$$\mathbb{1}[\![\{A, s\}]\!] = \mathcal{C}_{D^n}^{n+1}[\![\{A, s\}]\!]$$

2.4 Trees

Pasting diagrams like the ones used in Catt have a well known correspondence to finite planar rooted trees (henceforth, simply "trees") [18]. Our insertion construction (see Section 3) is more easily defined using the representation of a pasting context as a tree, and so we pause to work out this correspondence here.

Suspension. In topology, given a space X, we can construct a new space ΣX which is obtained by stretching X into a cylinder and then collapsing each of the top and bottom "caps" to a point. More formally, the suspension is a quotient of the product space $X \times [0, 1]$, where everything $X \times \{0\}$ is identified, and everything in $X \times \{1\}$ is identified. This construction may also be described simply in terms of paths: the space ΣX has two points with a path between these points for each element of X. As an example, suspending a circle yields a sphere with the original circle embedded as the equator. Each point of the circle then determines a meridian, and this motivates the convention of called the two additional points "North" and "South".

An analogue of the suspension operation exists in the theory Catt [2]. As with many constructions in Catt, it is mutually inductively defined on all pieces of syntax.

- For context Γ, its suspension $\Sigma\Gamma$ has two new variables $N : \star, S : \star$ (N for north and S for south), as well as a variable $x : \Sigma(A)$ for each $x : A \in \Gamma$.
- For type $A : \mathsf{Type}_\Gamma$, its suspension $\Sigma A : \mathsf{Type}_{\Sigma(\Gamma)}$ is given by $\Sigma\star = N \to_\star S$ and $\Sigma(s \to_A t) = \Sigma s \to_{\Sigma A} \Sigma t$. Note that this raises the dimension of the type by 1.
- For term $s : \mathsf{Term}_\Gamma$, its suspension $\Sigma s : \mathsf{Term}_{\Sigma(\Gamma)}$ is defined by $\Sigma x = x$ for variables $x \in \Gamma$, and $\Sigma(\mathsf{coh}\ (\Delta : A)[\sigma]) = \mathsf{coh}\ (\Sigma\Delta : \Sigma A)[\Sigma\sigma]$.
- For substitution $\sigma : \Delta \to \Gamma$, the suspension $\Sigma\sigma : \Sigma\Delta \to \Sigma\Gamma$ sends N to N, S to S and x to Σt for each $x \mapsto t \in \sigma$.

We note that for $\Sigma(\mathsf{coh}\ (\Delta : A)[\sigma])$ to be well typed we must have that $\Sigma\Delta$ is a pasting diagram. This is in fact the case whenever Δ is itself a pasting diagram. One can additionally show that suspension forms a functor $\Sigma : \mathsf{Catt} \to \mathsf{Catt}$ on the category of contexts.

Example 5. The suspension of $x : \star, y : \star, f : x \to_\star y, z : \star, g : y \to_\star z$, is the following context:

$$\begin{tikzcd}
N \arrow[rr, bend left=70, "z"] \arrow[rr, "y" description] \arrow[rr, bend right=70, "x"'] \arrow[rr, phantom, bend left=35, "g\Uparrow"] \arrow[rr, phantom, bend right=35, "f\Uparrow"] & & S
\end{tikzcd}$$

and the suspension of the 1-composition operation gives the vertical composition of 2 cells.

Lemma 6. *For all n, $\Sigma D^n \equiv D^{n+1}$. Further for all pasting contexts Δ:*

$$\Sigma\mathcal{C}^n_\Delta \equiv \mathcal{C}^{n+1}_{\Sigma\Delta} \quad \Sigma\mathcal{T}^n_\Delta \equiv \mathcal{T}^{n+1}_{\Sigma\Delta} \quad \Sigma\mathcal{U}^n_\Delta \equiv \mathcal{U}^{n+1}_{\Sigma\Delta}$$

Wedge Sum. In topology, the wedge sum of two pointed spaces (X, x), (Y, y) is the disjoint union of $X \coprod Y$, with x identified with y. This may be realized as a colimit of the following diagram:

$$\begin{tikzcd}
X & & Y \\
& \bullet \arrow[ul, "x"] \arrow[ur, "y"'] &
\end{tikzcd}$$

A similar construction may be made for pasting contexts. We note first that it is a basic consequence of the their definition that every pasting context begins with a variable of type $\star$. Now, given $\Gamma = \Gamma', (x : \star), \Gamma''$ where Γ'' contains no variables of type $\star$ (i.e., the variable x is the *last* variable of this type) and $\Delta = (y : \star), \Delta'$, we may define

$$\Gamma \vee \Delta := \Gamma', (x : \star), \Gamma'', \Delta'[x/y]$$

where $\Delta'[x/y]$ denotes the result of substituting x for y in all the types which appear in Δ'. One easily checks that the result is again a pasting diagram. Moreover, this construction has an obvious extension to multiple pasting contexts, which we write as:

$$\bigvee_{i=1}^{n} \Gamma_i = \Gamma_1 \vee \Gamma_2 \vee \cdots \vee \Gamma_n$$

Example 7. We consider $D^2 \vee D^2$. This glues two 2-discs at a point and yields a context of the following form:

$$\begin{tikzcd} \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Uparrow"] & \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Uparrow"] & \bullet \end{tikzcd}$$

We note that this is a pasting diagram which can be used to define the gives horizontal composition operation on 2-cells.

To simplify definitions of substitutions between wedge sums of pasting contexts, we will write substitutions diagrammatically by specifying the individual components. Indeed given substitutions $\sigma : \Gamma \to \Gamma'$ and $\tau : \Delta \to \Delta'$ such that σ sends the last terminal $\star$-typed variable of Γ to that of Γ' and τ sends the initial variable of Δ to that of Δ', one sees easily that there is a well defined substitution $\sigma \vee \tau : \Gamma \vee \Delta \to \Gamma' \vee \Delta'$ which we will depict as:

$$\begin{tikzcd} \Gamma' & \vee & \Delta' \\ \Gamma \arrow[u, "\sigma"] & \vee & \Delta \arrow[u, "\tau"] \end{tikzcd}$$

Tree Contexts. We now have the machinery needed to define the context generated from a tree. Our definition of tree will be based on lists which we will write in square bracket notation $[x_1, \ldots, x_n]$. We also use common list notations such as $\_ \mathbin{+\!\!+} \_$ for concatenating two lists, `[]` for the empty list, and $n :: ns$ for the list with first element n and tail given by the list ns.

We these conventions, we may define trees inductive as follows:

Definition 8. A *planar rooted tree* is a (possibly empty) list of planar rooted trees.

Subtrees of a tree can be indexed by a list of natural numbers P, giving a subtree T^P, by letting $T^{[]} = T$ and $T^{k::P} = (T_k)^P$ if $T = [T_1, \ldots, T_n]$.

Each tree generates a context, using the constructions of the previous subsections.

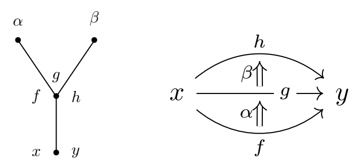
Definition 9. For a tree $T = [T_1, \ldots, T_n]$, the context $\lfloor T \rfloor$ generated from it is given by:

$$\lfloor T \rfloor := \bigvee_{i=1}^{n} \Sigma \lfloor T_i \rfloor$$

Here we understand the convention that when T is the empty list, we get a singleton context of the form $\_ : \star$. We will commonly abuse notation and omit the $\lfloor - \rfloor$ operator and use trees as contexts when it will not cause confusion.

By a simple induction using the properties of suspensions and wedge sums, we get that any context generated from a tree is a pasting context. The stronger result holds that $\lfloor - \rfloor$ is an isomorphism between trees and pasting diagrams, though we omit the proof here as it will not be needed for the definition of insertion. Next we give some simple definitions on trees that will be needed later.

Definition 10. The dimension of a tree $\dim(T)$ is 0 if T is empty or $1 + \max_k \dim(T_k)$ if $T = [T_1, \ldots, T_n]$. For a tree T, its *trunk height*, $\mathsf{th}(T)$, is $1 + \mathsf{th}(T_1)$ if $T = [T_1]$ and 0 otherwise. A tree is *linear* if its trunk height equals its dimension.



$$_{x}(f^{\alpha}g^{\beta}h)_{y}$$

Figure 5: Example tree and identity labelling.

Under the bijection between trees and pasting contexts, a substitution $\sigma : \lfloor T \rfloor \to \Gamma$ from the context associated to T to an arbitrary context Γ may be represented by an appropriate type of *labellings* of T which we now define:

Definition 11 (Tree Labelling)**.** A labelling $L : T \to \Gamma$ from a tree $T = [T_1, \ldots, T_n]$ to Γ is the following data:

$$t_0{}^{L_1}t_1 \ \cdots \ {}^{L_n}t_n$$

where each t_i is a term of Γ and each $L_i : T_i \to \Gamma$ is itself a labelling of T_i in Γ. The terms t_i label the 0-dimensional variables of the tree, and this can be graphically represented by writing each term between the two branches of the tree it sits between, as in Fig. 5. Every tree has an identity labelling $\mathrm{id}_T : T \to T$.

Example 12. Fig. 5 shows a graphical representation of the tree `[[[][]]]` and the context it generates. It graphically represents the identity labelling on this tree as well as giving its regular representation on the right.

2.5 Catt with Equality Rules

We conclude this section by extending Catt with a definitional equality determined by a set of *equality generators* $\mathcal{R}$. We denote the resulting theory by $\mathsf{Catt}_{\mathcal{R}}$. Formally, each generator $R \in \mathcal{R}$ is given by a triple $R = (\Gamma, s, t)$ where Γ is a context and $s, t \in \mathsf{Term}_\Gamma$. In what follows, we identify certain properties a set of generators $\mathcal{R}$ might enjoy which endow the resulting definitional equality with useful meta-theoretic behavior. We then give some preliminary examples.

To begin, we fix a set $\mathcal{R}$ of equality generators and add inductively defined equality judgments

$$\begin{array}{ll} \Gamma \vdash s = t & \text{Terms } s, t : \mathsf{Term}_\Gamma \text{ are equal.} \\ \Gamma \vdash A = B & \text{Types } A, B : \mathsf{Type}_\Gamma \text{ are equal.} \\ \Gamma \vdash \sigma = \tau & \text{Substitutions } \sigma, \tau : \Delta \to \Gamma \text{ are equal.} \end{array}$$

These will be defined mutually inductively alongside the typing rules. We also add the following new typing rule, named the *conversion rule*:

$$\frac{\Gamma \vdash s : A \qquad \Gamma \vdash A = B}{\Gamma \vdash s : B}$$

Finally, in addition to the structural rules given in Fig. 6, we add a family of rules:

$$\frac{x \in \Gamma}{\Gamma \vdash x = x} \qquad \frac{\Gamma \vdash s = t}{\Gamma \vdash t = s} \qquad \frac{\Gamma \vdash s = t \qquad \Gamma \vdash t = u}{\Gamma \vdash s = u}$$

$$\frac{\Delta \vdash A = B \qquad \Gamma \vdash \sigma = \tau}{\Gamma \vdash \mathsf{coh}\ (\Delta : A)[\sigma] = \mathsf{coh}\ (\Delta : B)[\tau]} \qquad \frac{}{\Gamma \vdash \star = \star}$$

$$\frac{\Gamma \vdash s = s' \qquad \Gamma \vdash t = t' \qquad \Gamma \vdash A = A'}{\Gamma \vdash s \to_A t = s' \to_{A'} t'} \qquad \frac{}{\Gamma \vdash \langle\rangle = \langle\rangle}$$

$$\frac{\Gamma \vdash \sigma = \tau \qquad \Gamma \vdash s = t}{\Gamma \vdash \langle \sigma, x \mapsto s \rangle = \langle \tau, x \mapsto t \rangle}$$

Figure 6: Structural rules for definitional equality.

$$\frac{(\Gamma, s, t) \in \mathcal{R} \qquad \Gamma \vdash s : A}{\Gamma \vdash s = t}$$

These rules are deliberately asymmetric; Only the left hand side requires a proof of validity. Preempting Section 4.1, this is because the equalities we use in our theories will take the form of a reduction, where the right hand side will be constructed from the left hand side of the equation. We refer to the equality relation = defined by these rules as *definitional equality.*

We now identify some attractive properties that the equality rules $\mathcal{R}$ can satisfy which make the resulting type theory well-behaved.

Lifting Condition. $\mathcal{R}$ has the lifting condition if for all $(\Gamma, s, t) \in \mathcal{R}$ and $A : \mathsf{Type}_\Gamma$:

$$\Gamma, A \vdash s = t$$

whenever $\Gamma, A \vdash s : B$ for some $B : \mathsf{Type}_{\Gamma, A}$. Note that even though s and t are terms over Γ, they can be lifted to terms over Γ, A.

This condition allows us to show that all equality and typing is preserved by context extension. It also gives us an easy proof that the identity substitution is well typed.

Substitution Condition. $\mathcal{R}$ has the substitution condition if for all $(\Gamma, s, t) \in \mathcal{R}$ and $\sigma : \Gamma \to \Delta$ with $\Delta \vdash \sigma : \Gamma$ and $\Delta \vdash s[\![\sigma]\!] : C$ for some $C : \mathsf{Type}_\Delta$ we have:

$$\Delta \vdash s[\![\sigma]\!] = t[\![\sigma]\!]$$

If $\mathcal{R}$ satisfies the substitution condition then more generally we have that typing and equality is preserved by substitution. We also get that given $\Delta \vdash \sigma = \tau$ and $s : \mathsf{Term}_\Gamma$ that $\Delta \vdash s[\![\sigma]\!] = s[\![\tau]\!]$, though this does not actually require the substitution condition.

If the set of rules $\mathcal{R}$ satisfies the lifting and substitution conditions, then there is a well-defined quotient category of well-typed contexts and substitutions which we will denote (as a slight abuse of notation) by $\mathsf{Catt}_\mathcal{R}$.

Suspension Condition. We say that $\mathcal{R}$ has the suspension condition if for all $(\Gamma, s, t) \in \mathcal{R}$ we have that if $\Sigma\Gamma \vdash \Sigma s : A$ for some $A : \mathsf{Type}_{\Sigma\Gamma}$ then:

$$\Sigma\Gamma \vdash \Sigma s = \Sigma t$$

This is sufficient to show that typing and equality is respected by suspension.

Definition 13. A set of equality generators $\mathcal{R}$ is *tame* is it satisfies the lifting, substitution, and suspension conditions.

In any tame theory, it can be shown that tree labellings and substitutions between wedge sums can be well typed, and it can also be shown that the unbiased constructions (type, term, and coherence) are valid.

Support Condition. $\mathcal{R}$ has the support condition if all $(\Gamma, s, t) \in \mathcal{R}$ with $\Gamma \vdash s : A$ we have $\mathsf{supp}(s) = \mathsf{supp}(t)$. Unsurprisingly, this condition being true implies all equalities preserve support.

While this rule may at first appear obvious to show, it turns out to be not quite so trivial. Despite knowing that s is valid, we have no guarantee that it is well behaved with respect to support, as it could contain equalities that do not preserve support. We therefore give the following lemma and proof strategy, with follows the method used in [5] to show preservation of support. We first define a set:

$$\mathcal{R}_\mathsf{s} = \{(\Gamma, s, t) \in \mathcal{R} \mid \mathsf{supp}(s) = \mathsf{supp}(t)\}$$

This generates a new type theory $\mathsf{Catt}_{\mathcal{R}_\mathsf{s}}$. For clarity we let this type theory have judgments of the form $\vdash_\mathsf{s}$.

Lemma 14. *Suppose for all $(\Gamma, s, t) \in \mathcal{R}$ such that $\Gamma \vdash_\mathsf{s} s : A$ for some $A : \mathsf{Type}_\Gamma$ we have that $\mathsf{supp}(s) = \mathsf{supp}(t)$. Then $\mathcal{R}$ satisfies the support condition.*

Using this lemma, we can make any constructions in $\mathcal{R}$ in $\mathsf{Catt}_{\mathcal{R}_\mathsf{s}}$ which will automatically give us that certain equalities preserve support. We will see this later with insertion, where we will show that the insertion operation is valid in any theory satisfying the appropriate conditions.

Conversion Condition. The last condition is the conversion condition that states that if $(\Gamma, s, t) \in \mathcal{R}$ then $\Gamma \vdash s : A$ implies $\Gamma \vdash t : A$. This along with the support condition is enough to show that equality preserves typing.

We can immediately see that $\mathsf{Catt} = \mathsf{Catt}_\varnothing$, and since $\varnothing$ trivially satisfies the above conditions, all the results above hold for Catt itself.

Disc Removal. We give two examples of equality rules from $\mathsf{Catt}_{\mathsf{su}}$ [5] which will be reused for $\mathsf{Catt}_{\mathsf{sua}}$. The first trivialises unary compositions.

Definition 15 (Disc removal)**.** Recall that any substitution from a disc is of the form $\{A, t\}$ for some term t and type A. Disc removal equates the terms $\mathcal{C}^n_{D^n}[\![\{A, s\}]\!]$ and s in context Γ for any n, $s : \mathsf{Term}_\Gamma$, and $A : \mathsf{Type}_\Gamma$. In other words, disc removal gives us equalities of the following form, after unwrapping the constructions above:

$$\Gamma \vdash \mathsf{coh}\ (D^n : U^n_{D^n})[\{A, s\}] = s$$

This can be intuitively understood as saying "the term s is equal to (s), the unary composite of s".

Disc removal gives the property that $\mathcal{C}^n_\Delta = \mathcal{T}^n_\Delta$ for $n > 0$, effectively removing the need to differentiate between the two when working up to definitional equality.

Endo-Coherence Removal. The second equality rule simplifies a class of terms called "endo-coherences". These are terms of the following form:

$$\mathsf{coh}\ (\Delta : s \to_A s)[\sigma]$$

These are "coherence laws" that can be understood intuitively as saying "the term $s[\sigma]$ is equal to the term $s[\sigma]$". But we already have a canonical way to express that — the identity on $s[\sigma]$. This inspires the following reduction.

Definition 16 (Endo-coherence removal)**.** Endo-coherence removal is the following class of equalities:

$$\Gamma \vdash \mathsf{coh}\ (\Delta : s \to_A s)[\sigma] = \mathbb{1}[\![\{A[\![\sigma]\!], s[\![\sigma]\!]\}]\!]$$

for all Δ, s, A, and σ such that $\mathsf{coh}\ (\Delta : s \to_A s)[\sigma]$ is not already an identity.

It can be checked (using proofs from [5]) that both disc removal and endo-coherence removal satisfy all the conditions listed above. By defining the rule-set $\mathsf{su} = \{$disc removal $\cup$ endo-coherence removal $\cup$ pruning$\}$, the type theory $\mathsf{Catt}_{\mathsf{su}}$ is recovered. This establishes that $\mathsf{Catt}_{\mathsf{su}}$ is part of the general schema presented in this section.

3 Insertion

The semistrict behaviour of our type theory $\mathsf{Catt}_{\mathsf{sua}}$, our adaptation of Catt with strict units and associators, is principally driven by a new equality rule called "insertion". This equality rule incorporates part of the structure of an argument context into the head context, simplifying the overall syntax of the term.

To be a candidate for insertion, an argument must not occur as the source or target of another argument of the term, and we call such arguments *locally maximal.* Consider the composite $f \cdot (g \cdot h)$. This term has two locally maximal arguments, f and $g \cdot h$, the second of which is an (unbiased) coherence. Insertion allows us to merge these two composites into one by "inserting" the pasting diagram of the inner coherence into the pasting diagram of the outer coherence. In the case above we will get that the term $f \cdot (g \cdot h)$ is equal to the ternary composite $f \cdot g \cdot h$, a term with a single coherence. As the term $(f \cdot g) \cdot h$ also reduces by insertion to the ternary composite, we see that both sides of the associator become equal under insertion. The action of insertion on these contexts is shown in Fig. 7.

$$\begin{array}{ccc} \boxed{x' \xrightarrow{f'} y' \xrightarrow{g'} z'} & & \\ \downarrow & \rightsquigarrow & \\ x \xrightarrow{f} y \xrightarrow{g} z & & x \xrightarrow{f} x' \xrightarrow{f'} y' \xrightarrow{g'} z' \end{array}$$

Figure 7: Insertion acting on the composite $f \cdot (g \cdot h)$

More generally we consider a coherence term $\mathsf{coh}\ (\Delta : A)[\sigma] : \mathsf{Term}_\Gamma$, where there is some locally maximal variable $x : A \in \Delta$ such that $x⟦\sigma⟧$ is itself an unbiased coherence $\mathcal{C}^n_\Theta⟦\tau⟧$. Under certain conditions on the shape of Γ and Δ (which will be specified in Section 3.1) we will construct the following data as part of the insertion operation:

- The *inserted context* $\Delta \ll_x \Theta$, obtained by inserting Θ into Δ along x. The inserted context is a pasting diagram.
- The *interior substitution* $\iota : \Theta \to \Delta \ll_x \Theta$, the inclusion of Θ into a copy of Θ living in the inserted context.
- The *exterior substitution* $\kappa : \Delta \to \Delta \ll_x \Theta$, which maps x to unbiased coherence over the copy of Θ, or more specifically $\mathcal{C}^n_\Theta⟦\iota⟧$, and other locally maximal variables to their copy in the inserted context.
- The *inserted substitution* $\sigma \ll_x \tau : \Delta \ll_x \Theta \to \Gamma$, which collects the appropriate parts of σ and τ.

Using this notation, insertion yields the following reduction:

$$\mathsf{coh}\ (\Delta : A)[\sigma] \rightsquigarrow \mathsf{coh}\ (\Delta \ll_x \Theta : A⟦\kappa⟧)[\sigma \ll_x \tau]$$

These constructions can be assembled into the following diagram:

$$\begin{tikzcd} D^n \arrow[r, "\{A,x\}"] \arrow[d, "\{\mathcal{U}^n_\Theta, \mathcal{C}^n_\Theta\}"'] & \Delta \arrow[d, "\kappa"] \arrow[ddr, bend left, "\sigma"] & \\ \Theta \arrow[r, "\iota"'] \arrow[drr, bend right, "\tau"'] & \Delta \ll_x \Theta \arrow[ul, phantom, very near start, "\ulcorner"] \arrow[dr, dashed, "\sigma \ll_x \tau"] & \\ & & \Gamma \end{tikzcd}$$

The commutativity of the outer boundary is the equation $x⟦\sigma⟧ = \mathcal{C}_\Theta^n⟦\tau⟧$, one of the conditions for the construction. The commutativity of the inner square is a property of the external substitution as stated above. Furthermore, as suggested by the diagram, $\Delta \ll_x \Theta$ is a pushout, and $\sigma \ll_x \tau$ is unique map to Γ determined by the universal property of the pushout. This gives a different way to think of the insertion, and gives the intuition that insertion is the result of taking the disjoint union of the two contexts, and gluing together x in the first with the unbiased coherence over the second.

3.1 The Insertion Construction

We have stated the existence of inserted contexts and associated maps, and claimed that they satisfy a universal property. In this section we give a direct construction of these objects. All constructions will be done by induction over the tree structure of pasting diagrams, which were introduced in Section 2.4. Trees are more convenient for our technical development than contexts, and so we will work with trees throughout.

To allow us to proceed with inductive definitions we need an inductive version of a locally maximal variable, which we will call a *branch*. We define some properties of branches as follows, illustrated in Figure 8.

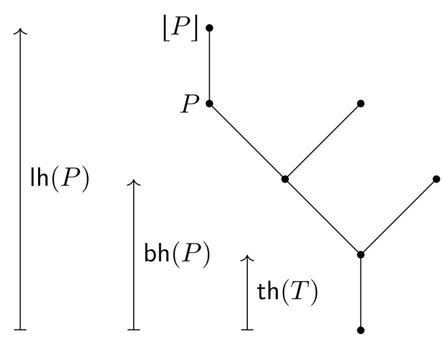


Figure 8: Illustrating leaf height, branch height and trunk height.

Definition 17. A *branch* P of a tree T is a non-empty indexing list for a subtree of T which is linear. A branch P of T gives a locally maximal variable $\lfloor P \rfloor$ of $\lfloor T \rfloor$ by taking the unique locally maximal variable of $\lfloor T_P \rfloor$. Define the *branch height* of P, denoted $\mathsf{bh}(P)$, to be one less that the length of P (note that P is always non-empty). Finally define the *leaf height* $\mathsf{lh}(P)$ of a path P as the dimension of $\lfloor P \rfloor$. As with trees, we will omit the $\lfloor - \rfloor$ notation and use a branch as a variable when it is clear.

For every locally maximal variable, there is some branch representing it, though not necessarily a unique one. Recall Definition 10 of trunk height for a tree. We now give one of the central definitions of the paper, which was also given an informal exposition in the introduction, and illustrated with Fig. 1.

Definition 18 (Inserted Tree)**.** Given trees S and T, and a branch P of S such that $\mathsf{th}(T) \geqslant \mathsf{bh}(P)$, we define the *inserted tree* $S \ll_P T$ by induction on the length of P:

- Suppose $P = [k]$ and $S = [S_1, \dots, S_k, \dots, S_n]$. Then:

$$S \ll_P T = [S_1, \dots, S_{k-1}] \mathbin{+\!\!+} T \mathbin{+\!\!+} [S_{k+1}, \dots, S_n]$$

- Suppose P has length greater than 1 so that $P = k :: P'$ and again $S = [S_1, \dots, S_k, \dots, S_n]$. We note that P' is a branch of S_k and by the condition on trunk height of T we have $T = [T_1]$. Then:

$$S \ll_P T = [S_1, \dots, S_{k-1}, S_k \ll_{P'} T_1, S_{k+1}, \dots, S_n]$$

We draw attention to the condition of the trunk height of T being at least the branch height of P, which is necessary for the induction to proceed.

We now proceed to define the interior and exterior substitutions, which will be done using the diagrammatic notation introduced in Section 2.4.

Definition 19 (Interior Substitution)**.** Given S, T trees, with P a branch of S with $\mathsf{th}(T) \geqslant \mathsf{bh}(P)$ we define the interior substitution $\iota_{S,P,T} : T \to S \ll_P T$ by induction on P.

- When $P = [k]$, $S = [S_1, \dots, S_k, \dots, S_n]$ we get:

$$\begin{tikzcd}[S_1, \dots, S_{k-1}] & \vee & T & \vee & [S_{k+1}, \dots, S_n] \\ & & T \arrow[u, "\mathrm{id}"] & & \end{tikzcd}$$

- When $P = k :: P'$, $S = [S_1, \dots, S_k, \dots, S_n]$ we get:

$$\begin{tikzcd}[S_1, \dots, S_{k-1}] & \vee & \Sigma S_k \ll_{P'} T_1 & \vee & [S_{k+1}, \dots, S_n] \\ & & \Sigma T_1 \arrow[u, "\Sigma\iota_{S_k,P',T_1}"] & & \end{tikzcd}$$

We may drop the subscripts on ι when they are easily inferred.

Definition 20 (Exterior Substitution)**.** Given S, T trees, with P a branch of S with $\mathsf{th}(T) \geqslant \mathsf{bh}(P)$ we define the exterior substitution $\kappa_{S,P,T} : S \to S \ll_P T$ by induction on P.

- When $P = [k]$, $S = [S_1, \dots, S_k, \dots, S_n]$ we get:

$$\begin{tikzcd}[S_1, \dots, S_{k-1}] & \vee & T & \vee & [S_{k+1}, \dots, S_n] \\ {[S_1, \dots, S_{k-1}]} \arrow[u, "\mathrm{id}"] & \vee & \Sigma S_k \arrow[u, "\{\mathcal{U}_T^n, \mathcal{C}_T^n\}"] & \vee & {[S_{k+1}, \dots, S_n]} \arrow[u, "\mathrm{id}"] \end{tikzcd}$$

  Where we note that by the condition of P being a branch we have that S_k is linear and so $\Sigma[S_k]$ is a disc.

- When $P = k :: P'$, $S = [S_1, \dots, S_k, \dots, S_n]$ we get:

$$\begin{tikzcd}[S_1, \dots, S_{k-1}] & \vee & \Sigma S_k \ll_{P'} T_1 & \vee & [S_{k+1}, \dots, S_n] \\ {[S_1, \dots, S_{k-1}]} \arrow[u, "\mathrm{id}"] & \vee & \Sigma S_k \arrow[u, "\Sigma\kappa_{S_k,P',T_1}"] & \vee & {[S_{k+1}, \dots, S_n]} \arrow[u, "\mathrm{id}"] \end{tikzcd}$$

Again the subscripts on κ may be dropped where they can be inferred.

Lastly we define the inserted substitution.

Definition 21 (Inserted Substitution)**.** Given S, T trees, with P a branch of S with $\mathsf{th}(T) \geqslant \mathsf{bh}(P)$ and $\sigma : S \to \Gamma$, $\tau : T \to \Gamma$, we define the *inserted substitution* $\sigma \ll_P \tau : S \ll_P T \to \Gamma$. Without loss of generality, we can assume that σ and τ are given by labellings L, M of S and T, and that we need to provide a labelling $L \ll_P M : S \ll_P T \to \Gamma$. Let

$$S = [S_1, \dots, S_n] \qquad L = {}_{s_0}{L_1}_{s_1} \; \dots \; {L_n}_{s_n}$$

and then proceed by induction on P.

- Let $P = [k]$, and

  $$T = [T_1, \dots, T_m] \qquad M = {}_{t_0}{M_1}_{t_1} \; \dots \; {M_m}_{t_m}$$

  Then define $L \ll_{[k]} M$ to be:

  $${}_{s_0}{L_1}_{s_1} \; \dots \; {L_{k-1}}_{t_0}{M_1}_{t_1} \; \dots \; {M_m}_{t_m}{L_{k+1}}_{s_{k+1}} \; \dots \; {L_n}_{s_n}$$

- Suppose $P = k :: P'$ so that

  $$T = [T_1] \qquad M = {}_{t_0}{M_1}_{t_1}$$

  Define $L \ll_P M$ as:

  $${}_{s_0}{L_1}_{s_1} \; \dots \; {L_{k-1}}_{t_0}(L_k \ll_{P'} M_1)_{t_1}{L_{k+1}}_{s_{k+1}} \; \dots \; {L_n}_{s_n}$$

The inserted substitution is then defined as the substitution corresponding to this labelling.

As we need a lot of data to perform an insertion, we will package it up to avoid repetition.

Definition 22. An *insertion point* is a triple (S, P, T) where S and T are trees and P is a branch of S with $\mathsf{bh}(P) \leqslant \mathsf{th}(T)$. An *insertion redex* is a sextuple $(S, P, T, \Gamma, \sigma, \tau)$ where (S, P, T) is an insertion point, $\sigma : S \to \Gamma$ and $\tau : T \to \Gamma$ are substitutions, and $P[\![\sigma]\!] \equiv \mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!]$.

3.2 Properties of Insertion

Here all typing judgements are taken with respect to a tame set of rules $\mathcal{R}$ which are taken as implicit. We will state various properties of the constructions in the previous section in any such $\mathsf{Catt}_{\mathcal{R}}$.

Proposition 23. *If (S, P, T) is an insertion point then:*

$$S \ll_P T \vdash \iota : T \quad S \ll_P T \vdash \kappa : S$$

If further $(S, P, T, \Gamma, \sigma, \tau)$ is an insertion redex then:

$$\Gamma \vdash \sigma \ll_P \tau : S \ll_P T$$

Proof. All substitutions are built from standard constructions, and their typing follows from the typing of these constructions, working by induction on P. The equality $P[\![\sigma]\!] \equiv \mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!]$ is needed to show that the generated labelling is valid. □

Lemma 24. *For any insertion redex $(S, P, T, \Gamma, \sigma, \tau)$:*

$$\Sigma(\sigma \ll_P \tau) \equiv \Sigma\sigma \ll_{0::P} \Sigma\tau$$

Given another substitution $\mu : \Gamma \to \Delta$, we have:

$$(\sigma \ll_P \tau) \circ \mu \equiv (\sigma \circ \mu) \ll_P (\tau \circ \mu)$$

and the constructions above are well defined.

We next state the required conditions for the universal property of insertion. All of these can be proven by induction on P.

Lemma 25. *For all insertion points (S, P, T), the terms $P[\![\kappa]\!]$ and $\mathcal{C}_T^{\mathsf{lh}(P)}[\![\iota]\!]$ are syntactically equal. If we extend to an insertion redex $(S, P, T, \Gamma, \sigma, \tau)$ then the following hold:*

$$\iota_{S,P,T} \circ (\sigma \ll_P \tau) = \tau$$
$$\kappa_{S,P,T} \circ (\sigma \ll_P \tau) = \sigma$$

This leads us to the following theorem, proving that insertion arises as a pushout.

Theorem 26. *The following diagram is a pushout in $\mathsf{Catt}_{\mathcal{R}}$ for any insertion point (S, P, T), where A is the type of $\lfloor P \rfloor$:*

$$\begin{tikzcd} D^n \arrow[r, "\{A,P\}"] \arrow[d, "\{\mathcal{U}_\Theta^{\mathsf{lh}(P)}, \mathcal{C}_\Theta^{\mathsf{lh}(P)}\}"'] \arrow[dr, phantom, "\ulcorner"] & S \arrow[d, "\kappa"] \\ T \arrow[r, "\iota"'] & S \ll_P T \end{tikzcd}$$

Proof. All that is left to show after Lemma 25 is that the inserted substitution is the unique substitution satisfying the commutativity conditions. This is done by realising that each variable of $S \ll_P T$ is either the image of a variable in S or T. □

3.3 The Type Theory $\mathsf{Catt}_{\mathsf{sua}}$

All the ingredients are now in place to define our principal type theory $\mathsf{Catt}_{\mathsf{sua}}$. We first formally define the insertion rule.

Definition 27 (Insertion)**.** The *insertion rule* says that the following equation holds:

$$\Gamma \vdash \mathsf{coh}\ (S : A)[\sigma] = \mathsf{coh}\ (S \ll_P T : A[\![\kappa]\!])[\sigma \ll_P \tau]$$

for all insertion redexes $(S, P, T, \Gamma, \sigma, \tau)$ and types $A : \mathsf{Type}_S$, where $\mathcal{C}_T^{\mathsf{lh}(P)}$ is either an unbiased composite or an identity, and $\mathsf{coh}\ (S : A)[\sigma]$ is not an identity.

We now define the set of rules sua to be the union of insertion, disc removal, and endo-coherence removal, and let $\mathsf{Catt}_{\mathsf{sua}}$ be the type theory generated from these rules. As disc removal and endo-coherence removal satisfy all conditions, it remains to check that insertion also does. The lifting condition is trivial, as insertion does not interact with the ambient context the terms exist in. Substitution and suspension conditions follow quickly from Lemma 24 and some computation. This leaves support and conversion.

For support we appeal to the proof strategy detailed in Section 2.5. Suppose

$$\Gamma \vdash \mathsf{coh}\ (S : A)[\sigma] = \mathsf{coh}\ (S \ll_P T : A[\![\kappa]\!])[\sigma \ll_P \tau]$$

is a valid insertion under the insertion rule, and further that $\Gamma \vdash_{\mathsf{s}} \mathsf{coh}\ (S : A)[\sigma] : B$ (the left hand side of the rule is well typed in $\mathsf{Catt}_{\mathcal{R}_\mathsf{s}}$). Then, Lemma 25 holds in $\mathsf{Catt}_{\mathcal{R}_\mathsf{s}}$ which implies $\mathsf{supp}(\kappa \circ (\sigma \ll_P \tau)) = \mathsf{supp}(\sigma)$, and since κ is full (its support is the entire context), it follows that $\mathsf{supp}(\sigma \ll_P \tau) = \mathsf{supp}(\sigma)$, as required.

Lastly, conversion follows from Proposition 23. It is also necessary to show that the support conditions hold in the generated term, which is done in the formalisation.

The following two properties hold in $\mathsf{Catt}_{\mathsf{sua}}$. Their proofs appear later in the paper in Section 4.1, after more properties of insertion have been proved.

Theorem 28. *The following equality holds for any insertion redex $(S, P, T, \Gamma, \sigma, \tau)$:*

$$\Gamma \vdash \mathsf{coh}\ (S : A)[\sigma] = \mathsf{coh}\ (S \ll_P T : A[\![\kappa]\!])[\sigma \ll_P \tau]$$

even when $P[\![\sigma]\!] \equiv \mathcal{C}_T^{\mathsf{lh}(P)}$ is not an unbiased composite or identity.

Proof. Follows immediately from Lemma 60. $\square$

Theorem 29. *An insertion into an unbiased coherence is equal to an unbiased coherence. More specifically:*

$$\Gamma \vdash \mathsf{coh}\ (S \ll_P T : \mathcal{U}_S^n[\![\kappa]\!])[\sigma \ll_P \tau] = \mathcal{C}^n_{S \ll_P T}[\![\sigma \ll_P \tau]\!]$$

for any insertion redex $(S, P, T, \Gamma, \sigma, \tau)$ and $n \geqslant \dim(T)$.

Proof. Immediately follows from Lemma 59. $\square$

4 A Decision Procedure for $\mathsf{Catt}_{\mathsf{sua}}$

We show that equality for $\mathsf{Catt}_{\mathsf{sua}}$ is decidable and hence type checking is also decidable. This gives an algorithm for checking validity of $\mathsf{Catt}_{\mathsf{sua}}$ terms, which we have implemented and discuss in Section 5.

Decidability is shown providing a reduction relation $\rightsquigarrow$, such that the symmetric transitive reflexive closure of $\rightsquigarrow$ agrees with equality on the set of valid terms. It is also shown that $\rightsquigarrow$ is terminating, meaning for each term t we can generate a normal form $N(t)$, and confluent which implies uniqueness of normal forms. Therefore, equality of two terms s and t can be checking syntactic equality of their normal forms $N(s)$ and $N(t)$.

4.1 Reduction for $\mathsf{Catt}_{\mathsf{sua}}$

To define a reduction for $\mathsf{Catt}_{\mathsf{sua}}$, we define a reduction relation on terms, types and substitutions by mutual induction, and write these reductions $\Gamma \vdash \_ \rightsquigarrow \_$ with appropriate pieces of syntax replacing the underscores. The rules for the single step reduction are given in Fig. 9. We write the reflexive transitive closure $\Gamma \vdash \_ \rightsquigarrow^* \_$. It is a simple manipulation to show that definitional equality on $\mathsf{Catt}_{\mathsf{sua}}$ is the symmetric transitive reflexive closure of single step reduction.

$$\frac{\Delta \vdash A \rightsquigarrow B}{\Gamma \vdash \mathsf{coh}\ (\Delta : A)[\sigma] \rightsquigarrow \mathsf{coh}\ (\Delta : B)[\sigma]}\ \textsc{Cell} \qquad \frac{\Gamma \vdash \sigma \rightsquigarrow \tau}{\Gamma \vdash \mathsf{coh}\ (\Delta : A)[\sigma] = \mathsf{coh}\ (\Delta : A)[\tau]}\ \textsc{Arg}$$

$$\frac{\Gamma \vdash s \rightsquigarrow s'}{\Gamma \vdash s \to_A t \rightsquigarrow s' \to_A t} \qquad \frac{\Gamma \vdash A \rightsquigarrow A'}{\Gamma \vdash s \to_A t \rightsquigarrow s \to_{A'} t} \qquad \frac{\Gamma \vdash t \rightsquigarrow t'}{\Gamma \vdash s \to_A t \rightsquigarrow s \to_A t'}$$

$$\frac{\Gamma \vdash \sigma \rightsquigarrow \tau}{\Gamma \vdash \langle \sigma, x \mapsto s\rangle \rightsquigarrow \langle \tau, x \mapsto s\rangle} \qquad \frac{\Gamma \vdash s \rightsquigarrow t}{\Gamma \vdash \langle \sigma, x \mapsto s\rangle \rightsquigarrow \langle \sigma, x \mapsto t\rangle} \qquad \frac{(\Gamma, s, t) \in \mathsf{sua}}{\Gamma \vdash s \rightsquigarrow t}$$

Figure 9: One-step Reduction Rules

Due to the conversion condition, if we start with a well typed term, then any term arising as a reduction of it will also be well typed. In practice this means that we rarely need to check typing conditions when reducing a term, and as such we will omit the context Γ and just write $s \rightsquigarrow t$ or $s \rightsquigarrow^* t$, when we know s is well-typed. By inspecting the proof that sua is tame, in particular noting that the proofs do not use the symmetry of equality, we can deduce that $\rightsquigarrow^*$ respects context extension and substitution.

Termination. We show strong termination for the reduction, demonstrating that there are no infinite reduction sequences. Our strategy is to assign an ordinal number to each term, show that each single step reduction reduces the associated ordinal number, and therefore deduce that any infinite reduction sequence of the form above would imply the existence of an infinite chain of ordinials, which can not exist due to the well-foundedness of ordinal numbers. We call the ordinal number associated to each term its *syntactic complexity.*

To define syntactic complexity, we will need to use ordinal numbers up to ω^ω. We will also need a construction known as the natural sum of ordinals, $\alpha \# \beta$, which is associative, commutative, and strictly monotone in both of its arguments [11].

Definition 30. For all terms t and substitutions σ, the *syntactic complexity* $\mathsf{sc}(t)$ and $\mathsf{sc}(\sigma)$ are mutually defined as follows:

- For substitutions we have:

$$\mathsf{sc}(\langle t_0, \ldots, t_n\rangle) = \mathop{\#}_{i=0}^{n} t_i$$

- For terms, we have $\mathsf{sc}(x) = 0$ for variables x.

If $\mathsf{coh}\ (\Delta : A)[\sigma]$ is an identity then:

$$\mathsf{sc}(\mathsf{coh}\ (\Delta : A)[\sigma]) = \omega^{\dim(A)} \mathbin{\#} \mathsf{sc}(\sigma)$$

Otherwise:

$$\mathsf{sc}(\mathsf{coh}\ (\Delta : A)[\sigma]) = 2\omega^{\dim(A)} \mathbin{\#} \mathsf{sc}(\sigma)$$

The motivation for syntactic complexity is as follows. We would like to show that each reduction reduces the depth of the syntax tree, but this doesn't quite work, as reductions like insertion can add new constructions into the reduced term. The necessary insight is that these constructions only add complexity in a lower dimension than the term being reduced. Syntactic complexity measures the depth of the syntax at each dimension, where removing a coherence of dimension n reduces the complexity, even if we add arbitrary complexity at lower dimensions. Syntactic complexity also treats identities in a special way, as these play a special role in blocking reduction in the theory.

The syntactic complexity does not account for the type in a coherence, as this is difficult to encode. Instead of showing that all reductions reduce syntactic complexity, we instead show that all reduction which are not "cell reductions" (reductions that have the rule marked "Cell" in their derivation) reduce syntactic complexity and deduce that a hypothetical infinite reduction sequence must only consist of cell reductions after a finite number of steps, and then appeal to an induction on dimension.

Lemma 31. *The following inequality:*

$$\mathsf{sc}(\sigma \ll_P \tau) < \mathsf{sc}(\sigma)$$

holds for any insertion redex $(S, P, T, \Gamma, \sigma, \tau)$*.*

Proof. We begin by noting that:

$$\begin{aligned}
\mathsf{sc}(\sigma) &= \mathop{\#}_{x \neq \lfloor P \rfloor} \mathsf{sc}(x[\![\sigma]\!]) \mathbin{\#} \mathsf{sc}(P[\![\sigma]\!]) \\
&= \mathop{\#}_{x \neq \lfloor P \rfloor} \mathsf{sc}(x[\![\sigma]\!]) \mathbin{\#} \mathsf{sc}(\mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!]) \\
&> \mathop{\#}_{x \neq \lfloor P \rfloor} \mathsf{sc}(x[\![\sigma]\!]) \mathbin{\#} \mathsf{sc}(\tau)
\end{aligned}$$

Further we extend the notion of syntactic depth to labels in the obvious way and therefore show that for all labels L and M with appropriate conditions that:

$$\mathsf{sc}(L \ll_P M) \leqslant \mathop{\#}_{x \neq \lfloor P \rfloor} \mathsf{sc}(x[\![L]\!]) \mathbin{\#} \mathsf{sc}(M)$$

which we do by induction on P. If $P = [k]$ then it is clear that $L \ll_P M$ contains all the terms of M and some of the terms of L, and crucially not $\lfloor P \rfloor[\![L]\!]$. If instead $P = k :: P'$ then by induction hypothesis we get that:

$$\mathsf{sc}(L_k \ll_{P'} M_1) \leqslant \mathop{\#}_{x \neq \lfloor P' \rfloor} \mathsf{sc}(x[\![L']\!]) \mathbin{\#} \mathsf{sc}(M_1)$$

It is then clear again that $L \ll_P M$ contains terms from M and terms of L which are not $P[\![L]\!]$, and so the inequality holds. $\square$

Theorem 32. *One-step reductions that do not use the cell rule reduce syntactic complexity. Those that do use the cell rule do not change the complexity.*

Proof. We wish to show that for all reductions $s \rightsquigarrow t$, and $\sigma \rightsquigarrow \tau$ that $\mathsf{sc}(t) < \mathsf{sc}(s)$, and $\mathsf{sc}(\tau) < \mathsf{sc}(\sigma)$ respectively. We proceed by induction on the derivation of the reduction, noting that all cases for structural rules follow from strict monotonicity of the natural sum. This leaves us with the following cases.

Disc removal. Suppose $\mathcal{C}^n_{D^n}[\![\{A, s\}]\!] \rightsquigarrow s$ is by disc removal. By a simple induction, $\mathsf{sc}(\{A, s\}) \geqslant \mathsf{sc}(s)$ and so:

$$\mathsf{sc}(s) \leqslant \mathsf{sc}(\{A, s\}) < \mathsf{sc}(\mathcal{C}^n_{D^n}[\![\{A, s\}]\!])$$

Endo-coherence removal. Let $\mathsf{coh}\ (\Delta : s \to_A s)[\sigma] \rightsquigarrow \mathbb{1}_{\{A[\![\sigma]\!], s[\![\sigma]\!]\}}$ be a reduction by endo-coherence removal. Then:

$$\begin{aligned}\mathsf{sc}(\mathbb{1}_{\{A[\![\sigma]\!], s[\![\sigma]\!]\}}) &= \omega^{1+\dim(A)} \# \mathsf{sc}(\{A[\![\sigma]\!], s[\![\sigma]\!]\}) \\ &< \omega^{1+\dim(A)} \# \omega^{1+\dim(A)} \\ &\leqslant \mathsf{sc}(\mathsf{coh}\ (\Delta : s \to_A s)[\sigma])\end{aligned}$$

Here the second line holds as $\dim(s) = 1 + \dim(A)$ and the last line holds as $\mathsf{coh}\ (\Delta : s \to_A s)[\sigma]$ cannot be an identity by assumption.

Insertion. Let $(S, P, T, \Gamma, \sigma, \tau)$ be an insertion redex so that:

$$\mathsf{coh}\ (S : A)[\sigma] \rightsquigarrow \mathsf{coh}\ (S \ll_P T : A[\![\kappa]\!])[\sigma \ll_P \tau]$$

by insertion. This implies that $\mathsf{coh}\ (S : A)[\sigma]$ is not an identity. Then:

$$\begin{aligned}&\mathsf{sc}(\mathsf{coh}\ (S \ll_P T : A[\![\kappa]\!])[\sigma \ll_P \tau]) \\ &\leqslant 2\omega^{\dim(A)} \# \mathsf{sc}(\sigma \ll_P \tau) \\ &< 2\omega^{\dim(A)} \# \mathsf{sc}(\sigma) \\ &\leqslant \mathsf{coh}\ (S : A)[\sigma]\end{aligned}$$

A simple induction shows that reductions using the cell rule do not modify the complexity, as when the cell rule is used, we modify a type that does not contribute to the syntactic complexity. $\square$

Corollary 33. *Reduction for* $\mathsf{Catt}_{\mathsf{sua}}$ *is strongly terminating.*

Proof. We proceed by induction on the dimension. Suppose there is an infinite reduction sequence, starting with a dimension k term:

$$s_0 \rightsquigarrow s_1 \rightsquigarrow s_2 \rightsquigarrow \cdots$$

Then by Theorem 32, only finitely many of these reductions do not use the cell rule, and so there is an n such that:

$$s_n \rightsquigarrow s_{n+1} \rightsquigarrow \cdots$$

are all cell reductions. Each of these reductions reduces one of finitely many subterms of s_n, and each of these subterms has dimension less than k, so by inductive hypothesis, none of these subterms can be reduced an infinitely often, contradicting the existence of an infinite reduction sequence. $\square$

Confluence. To prove confluence, we take the standard approach of proving local confluence, which says that all single step reductions of a term can be reduced (in any number of steps) to a common reduct. This implies full confluence (that multi-step reduction has the diamond property) and uniqueness of normal forms when combined with strong termination.

Theorem 34. *Reduction is locally confluent: if a is valid with $a \rightsquigarrow b$ and $a \rightsquigarrow c$, then there exists some d with $b \rightsquigarrow^* d$ and $c \rightsquigarrow^* d$.*

Before beginning the proof of this theorem, we give a higher level explanation of some of the difficulties of the proof, and the strategies we employ to overcome them.

When we examine simpler reductions such as disc removal, endo-coherence removal, or even reductions like pruning from $\mathsf{Catt}_{\mathsf{su}}$, we see that if there is a redex for that reduction, then other reductions do not "break" the redex. For example, a redex for pruning is a coherence term with an identity as a locally maximal argument. In $\mathsf{Catt}_{\mathsf{su}}$, identities are head normal forms, and so no matter how many reductions we apply to this argument, the argument remains an identity, and the pruning redex remains valid. Similarly, cases for disc removal tend to be easy to handle, as discs are also head normal forms. An endo-coherence removal redex can be broken, however in every case this can be easily fixed by application of further reductions.

With insertion the situation is very different. Insertion can occur when a locally maximal argument is an unbiased composite, and in $\mathsf{Catt}_{\mathsf{sua}}$ unbiased composites are not in general normal forms. Indeed, a biased composite such as $(f \cdot g) \cdot h$ has an unbiased composite at its heads. Furthermore, unbiased composites can reduce to terms which are no longer unbiased composites and are therefore no longer insertable, meaning that more complicated arguments are required to establish confluence.

The critical pair that demonstrates this most clearly is a coherence with an insertable argument, where this argument is itself a coherence with an insertable argument. The immediate problem is that performing the insertion on the inner term can cause it to no longer to be an unbiased coherence, as applying the exterior substitution to an unbiased type does not in general return an unbiased type. In order to recover the redex for the outer insertion, we need to use a directed version of Theorem 29, which is not a syntactic equality, and can involve applying more insertion reductions to return the term to an unbiased coherence.

This critical pair raises a second problem. Once we perform the inner insertion and then reduce the resulting term to yield an unbiased coherence once again, there is no guarantee that the resulting term is either a composite or identity. To proceed in this situation we use a directed version of Theorem 28, which shows that in the case that the term does not become a composite or identity, it must become an endo-coherence, thus allowing an endo-coherence removal step, followed by insertion of the resulting identity.

One may consider restructuring the theory to avoid dealing with this case, for example by allowing insertions of all unbiased coherences. However, taking this path would introduce a new critical pair given by an insertable argument that can reduce by endo-coherence removal, which presents new problems which we believe are at least as hard to solve.

Even after proving that these terms admit these reductions, we still need to show that the resulting terms, which can be by complex after applying multiple reductions, are indeed equal, closing the reduction square. This is an involved procedure, and as a result the confluence proof is rather lengthy.

We now start to state the various lemmas and definitions needed to show confluence. All lemmas are formalized in the accompanying Agda formalisation, which is linked in the introduction, and for this reason we omit proofs here of some straightforward lemmas which consist of a large case analysis.

We begin by defining some refined versions of equality, which help to prove confluence.

Definition 35. Define the n-bounded equality relation as follows: Let $\Gamma \vdash s =_n t$ when $\Gamma \vdash s = t$ with a derivation that only uses rules $(\Delta, s', t') \in \mathcal{R}$ when $\dim(s') < n$. We further define maximal equality by letting $\Gamma \vdash \sigma \equiv^{\mathsf{max}} \tau$ when substitutions σ and τ are syntactically equal when applied to locally maximal variables, and $\Gamma \vdash \sigma =^{\mathsf{max}} \tau$ when they are definitionally equal on all locally maximal variables.

It is clear that bounded equality implies equality. It is also true that maximal equality (of either variety) between valid substitutions implies equality due to the conversion. Further, we have that any equal terms of dimension n are n-bounded equal and so if $\Gamma \vdash \sigma \equiv^{\mathsf{max}} \tau$ then it follows that $\Gamma \vdash \sigma =_{\dim(\sigma)} \tau$. Lastly, if $\Gamma \vdash \sigma \equiv^{\mathsf{max}} \tau$ and both σ and τ are valid in $\mathsf{Catt}_{\varnothing}$ then it follows that $\sigma \equiv \tau$.

Lemma 36. *The terms $P[\![\kappa_{S,P,T}]\!]$ and $\mathcal{C}_T^{\mathsf{lh}(P)}[\![\iota]\!]$ are syntactically equal for any insertion point (S, P, T).*

Lemma 37. *For insertion redex $(S, P, T, \Gamma, \sigma, \tau)$, the following hold:*

$$\iota_{S,P,T} \circ (\sigma \ll_P \tau) \equiv \tau$$

$$\kappa_{S,P,T} \circ (\sigma \ll_P \tau) \equiv^{\mathsf{max}} \sigma$$

These imply the equality results from Section 3.2.

Lemma 38. *Suppose (S, P, T) and (S, Q, T) are insertion points with $\lfloor P \rfloor \equiv \lfloor Q \rfloor$. Then $S \ll_P T = S \ll_Q T$ and $\kappa_{S,P,T} \equiv^{\mathsf{max}} \kappa_{S,Q,T}$. If we further have $\sigma : S \to \Gamma$ and $\tau : T \to \Gamma$, then $\sigma \ll_P \tau \equiv^{\mathsf{max}} \sigma \ll_Q \tau$.*

Definition 39. Let (S, P, T) be an insertion point and Q be a branch of S such that $\lfloor P \rfloor \neq \lfloor Q \rfloor$. Then we can define a new branch $Q \ll_P T$ of $S \ll_P T$ with $\mathsf{bh}(Q \ll_P T) = \mathsf{bh}(Q)$ and $\lfloor Q \ll_P T \rfloor \equiv Q[\![\kappa_{S,P,T}]\!]$. Intuitively this branch refers to the same part of S, and is unaffected by T being inserted in.

Definition 40. We define a variant of the inserted substitution, and write it $\sigma \ll'_P \tau$. Whereas the original uses as many terms from τ as possible, the variant uses as many terms from σ as possible. More precisely, we define $L \ll'_{[k]} M$ to be:

$${s_0}{L_1}{s_1} \;\ldots\; {L_{k-1}}{\boldsymbol{s_k}}{M_1}{t_2} \;\ldots\; {M_m}{\boldsymbol{s_{k+1}}}{L_{k+1}}{s_{k+1}} \;\ldots\; {L_n}{s_n}$$

and $L \ll'_{k::P'} M$ as:

$${s_0}{L_1}{s_1} \;\ldots\; {L_{k-1}}{\boldsymbol{s_k}}(L_k \ll_{P'} M_1){\boldsymbol{s_{k+1}}}{L_{k+1}}{s_{k+1}} \;\ldots\; {L_n}{s_n}$$

where the terms in bold have been modified from the original definition. In the edge case where $M = \texttt{[]}$, we arbitrarily use s_k instead of s_{k+1} for the definition of $L \ll'_{[k]} M$.

Lemma 41. *The following equality holds*

$$\sigma \ll_P \tau =_{\dim(S)} \sigma \ll'_P \tau$$

for any insertion redex $(S, P, T, \Gamma, \sigma, \tau)$.

Lemma 42. *Let* (S, P, T) *and* (S, Q, U) *be insertion points such that* $\lfloor P \rfloor \neq \lfloor Q \rfloor$. *Then we have:*

$$(S \ll_P T) \ll_{Q \ll_P T} U = (S \ll_Q U) \ll_{P \ll_Q U} T$$

$$\kappa_{S,P,T} \circ \kappa_{S \ll_P T, Q \ll_P T, U} \equiv^{\mathsf{max}} \kappa_{S,Q,U} \circ \kappa_{S \ll_Q U, P \ll_Q U, T}$$

Further

$$(\sigma \ll_P \tau) \ll'_{Q \ll_P T} \mu \equiv^{\mathsf{max}} (\sigma \ll_Q \mu) \ll'_{P \ll_Q U} \tau$$

for any insertion redexes $(S, P, T, \Gamma, \sigma, \tau)$ *and* $(S, P, T, \Gamma, \sigma, \mu)$.

Lemma 43. *The following reduction holds, even when the left-hand side is an identity:*

$$\mathsf{coh}\ (\Gamma : s \to_A s)[\sigma] \rightsquigarrow^* \mathbb{1}[\![\{s, A\} \circ \sigma]\!]$$

Proof. If $\mathsf{coh}\ (\Gamma : s \to_A s)[\sigma]$ is not an identity then we can reduce by endo-coherence removal. Otherwise we have $\Gamma = D^n$ for some n, $s \equiv d_n$, and $A \equiv \mathcal{U}^n_{D^n}$, and so:

$$\mathbb{1}[\![\{s, A\} \circ \sigma]\!] \equiv \mathbb{1}[\![\{d_n, \mathcal{U}^n_{D^n}\} \circ \sigma]\!] \equiv \mathbb{1}[\![\sigma]\!]$$

It follows that the reduction is trivial. □

Lemma 44. *Let* T *be a tree,* $n \geqslant \dim(T)$, *and* P *a branch of* D^n *with* $\mathsf{bh}(P) \leqslant \mathsf{th}(T)$. *Then* $D^n \ll_P T = T$ *and* $\iota_{D^n,P,T} \equiv \mathrm{id}$. *Suppose further that* $\sigma : D^n \to \Gamma$ *and* $\tau : T \to \Gamma$. *Then* $\sigma \ll_P \tau \equiv^{\mathsf{max}} \tau$.

Lemma 45. *Let* S *be a tree, and* P *a branch of* S. *Then we get that* $S \ll_P D^{\mathsf{lh}(P)} = S$ *and* $\kappa_{S,P,D^{\mathsf{lh}(P)}} =^{\mathsf{max}} \mathrm{id}$. *Further*

$$\sigma \ll_P \tau \equiv^{\mathsf{max}} \sigma$$

if $(S, P, D^{\mathsf{lh}(P)}, \Gamma, \sigma, \tau)$ *is an insertion redex.*

Lemma 46. *Let* $n \in \mathbb{N}$ *and suppose* S *and* T *are trees, and* P *a branch of* S *such that:*

- $\mathsf{bh}(P) \leqslant \mathsf{th}(T)$
- $n < \mathsf{lh}(P)$
- $n \leqslant \mathsf{th}(T)$
- $\mathsf{lh}(P) \geqslant \dim(T)$

Then $\partial_n(S) = \partial_n(S \ll_P T)$ *and for* $\epsilon \in \{-, +\}$:

$$\delta^\epsilon_n(S) \circ \kappa_{S,P,T} \equiv^{\mathsf{max}} \delta^\epsilon_d(S \ll_P T)$$

Definition 47. Let $n \in \mathbb{N}$, and suppose S is a tree with branch P with $n > \mathsf{bh}(P)$. Then we can define a new branch ∂^n_P of ∂^n_S given by the same list as P.

Lemma 48. *Let $n \in \mathbb{N}$ and suppose S and T are trees, and P a branch of S such that:*

- $\mathsf{bh}(P) \leqslant \mathsf{th}(T)$
- $\mathsf{lh}(P) \geqslant \dim(T)$
- *one of the following holds:*
  1. $n > \mathsf{th}(T)$ *and* $n \leqslant \mathsf{lh}(P)$
  2. $n \geqslant \mathsf{lh}(P)$

Then $\partial_S^n \ll_{\partial_P^n} \partial_T^n = \partial_{S \ll_P T}^n$ *and:*

$$\delta_S^{\epsilon,n} \circ \kappa_{S,P,T} \equiv^{\mathsf{max}} \kappa_{\partial_S^n,\partial_P^n,\partial_T^n} \circ \delta_{S \ll_P T}^{\epsilon,n}$$

for $\epsilon \in \{-,+\}$.

Lemma 49. *For all n and S, $\mathcal{C}_S^n \rightsquigarrow^* \mathcal{T}_S^n$.*

Proof. The only case in which $\mathcal{C}_S^n \neq \mathcal{T}_S^n$ is when $S = D^n$, in which case a single disc removal gives the required reduction. □

Lemma 50. *Given $\sigma \rightsquigarrow^* \sigma'$ and $\tau \rightsquigarrow^* \tau'$, then if $\sigma \ll_P \tau$ is defined, we have:*

$$\sigma \ll_P \tau \rightsquigarrow^* \sigma' \ll_P \tau'$$

Lemma 51. *Let (S, P, T) be an insertion point. Then:*

$$\kappa_{S,P,T} \ll_P \iota_{S,P,T} \equiv \mathrm{id}$$

Lemma 52. *If P is a branch of S, and $\sigma, \sigma' : S \to \Gamma$ are substitutions differing only on $\lfloor P \rfloor$, then the following holds for insertion redex $(S, P, T, \Gamma, \sigma, \tau)$:*

$$\sigma \ll_P \tau \equiv \sigma' \ll_P \tau$$

Lemma 53. *Let (S, P, T) be an insertion point. Further assume S is not linear. Then* $\mathsf{th}(S \ll_P T) \geqslant \mathsf{th}(S)$.

Definition 54. Let (S, P, T) be an insertion point. Further assume T is not linear and has a branch Q. Then there is a branch $S \ll_P Q$ of $S \ll_P T$ with the same height as Q. The new branch points to the same place as Q, except in the copy of T which exists in $S \ll_P T$.

Lemma 55. *Let (S, P, T) be an insertion point. Further assume T is not linear and has a branch Q. Then $\lfloor S \ll_P Q \rfloor \equiv Q[\![\iota_{S,P,T}]\!]$. Further if (T, Q, U) is an insertion point, then*

$$S \ll_P (T \ll_Q U) = (S \ll_P T) \ll_{S \ll_P Q} U$$

and:

$$\kappa_{S,P,T \ll_Q U} =^{\mathsf{max}} \kappa_{S,P,T} \circ \kappa_{S \ll_P T, S \ll_P Q, U}$$

Further

$$\sigma \ll_P (\tau \ll_Q \mu) \equiv^{\mathsf{max}} (\sigma \ll_P \tau) \ll_{S \ll_P Q} \mu$$

for any $\sigma : S \to \Gamma$, $\tau : T \to \Gamma$, and $\mu : U \to \Gamma$

Definition 56. For tree S and branch P, let

$$S \mathbin{/\!\!/} P = S \ll_P D^{\mathsf{lh}(P)-1}$$

and let $\pi_P = \kappa_{S,P,D^{\mathsf{lh}(P)-1}}$.

Definition 57. Given S and branch P, if $2 + \mathsf{bh}(P) \leqslant \mathsf{lh}(P)$, then there is a branch P' of $S \mathbin{/\!\!/} P$, given by the same list as P.

Lemma 58. *If S has branch P with $2+\mathsf{bh}(P) \leqslant \mathsf{lh}(P)$, then $\lfloor P' \rfloor \equiv d_{\mathsf{lh}(P)-1}[\![\iota_{S,P,D^{\mathsf{lh}(P)-1}}]\!]$. If (S,P,T) is an insertion point, we further get that $(S \mathbin{/\!\!/} P) \ll_{P'} T = S \ll_P T$ and $\pi_P \circ \kappa_{S /\!\!/ P,P',T} =^{\mathsf{max}} \kappa_{S,P,T}$. If we are also given $\sigma : S \to \Gamma$ and $\tau : T \to \Gamma$ then:*

$$(\sigma \ll_P (\{\mathcal{T}_T^{\mathsf{lh}(P)-1}, \mathcal{U}_T^{\mathsf{lh}(P)-1}\} \circ \tau)) \ll_{P'} \tau \equiv^{\mathsf{max}} \sigma \ll_P \tau$$

Now, two larger metatheorems about $\mathsf{Catt}_{\mathsf{sua}}$ can be proven. The first implies that the term obtained by inserting into an unbiased coherence reduces back to an unbiased coherence on the inserted context. Despite the restriction in the theory of only allowing unbiased composites and identities to be inserted, the second proves that the insertion of any unbiased coherence can be simulated.

Lemma 59. *Let (S,P,T) be an insertion point. Then for $n \leqslant \dim(S)+1$, $\mathcal{U}_S^n[\![\kappa_{S,P,T}]\!] \rightsquigarrow^* \mathcal{U}^n_{S \ll_P T}$ and if $\dim(S) = n$ then $\mathcal{T}_S^n[\![\kappa_{S,P,T}]\!] \rightsquigarrow^* \mathcal{T}_{S \ll_P T}$.*

Proof. We proceed by induction on n, starting with the statement for types. If $n = 0$ then both unbiased types are $\star$, so we are done. Otherwise we have:

$$\begin{aligned}\mathcal{U}_S^{1+n}[\![\kappa_{S,P,T}]\!] \equiv{}& \mathcal{T}^n_{\partial^n_S}[\![\delta_S^{-,n} \circ \kappa_{S,P,T}]\!] \\ &\to_{\mathcal{U}_S^n[\![\kappa_{S,P,T}]\!]} \\ &\mathcal{T}^n_{\partial^n_S}[\![\delta_S^{+,n} \circ \kappa_{S,P,T}]\!]\end{aligned}$$

and

$$\begin{aligned}\mathcal{U}^{1+n}_{S \ll_P T} \equiv{}& \mathcal{T}^n_{\partial^n_{S \ll_P T}}[\![\delta^{-,n}_{S \ll_P T}]\!] \\ &\to_{\mathcal{U}^n_{S \ll_P T}} \\ &\mathcal{T}^n_{\partial^n_{S \ll_P T}}[\![\delta^{+,n}_{S \ll_P T}]\!]\end{aligned}$$

By inductive hypothesis: $\mathcal{U}_S^n[\![\kappa_{S,P,T}]\!] \rightsquigarrow^* \mathcal{U}^n_{S \ll_P T}$, and so we need to show that:

$$\mathcal{T}^n_{\partial^n_S}[\![\delta_S^{\epsilon,n} \circ \kappa_{S,P,T}]\!] \rightsquigarrow^* \mathcal{T}^n_{\partial^n_{S \ll_P T}}[\![\delta^{\epsilon,n}_{S \ll_P T}]\!]$$

We now note that either the conditions for Lemma 46 or Lemma 48 must hold. If conditions for Lemma 46 hold then (as everything is well typed in $\mathsf{Catt}_\varnothing$) we get that the required reduction is trivial. Therefore we focus on the second case. Here we get from Lemma 48 that:

$$\mathcal{T}^n_{\partial^n_S}[\![\delta_S^{\epsilon,n} \circ \kappa_{S,P,T}]\!] \equiv \mathcal{T}^n_{\partial^n_S}[\![\kappa_{\partial^n_S,\partial^n_P,\partial^n_T} \circ \delta^{\epsilon,n}_{S \ll_P T}]\!]$$

Then by the induction hypothesis for terms since $\dim(\partial^n_S) = n$ when $n \leqslant \dim(S)$, we get the required reduction.

Now we move on to the case for terms. If $\mathcal{T}_S^n$ is a variable, then we must have that S is linear and so $S = D^n$. We must also have in this case that $\mathcal{T}_S^n = \lfloor P \rfloor$. Then by Lemma 36, $\mathcal{T}_S^n[\![\kappa_{S,P,T}]\!] \equiv \mathcal{C}_T^n[\![\iota_{S,P,T}]\!]$ and then by Lemmas 44 and 49 this reduces to $\mathcal{T}_{S \ll_P T}^n$ as required. If $\mathcal{T}_S^n$ is not a variable, then $\mathcal{T}_S^n \equiv \mathcal{C}_S^n$ and as $n = \dim(S)$, $\mathcal{C}_S^n$ is not an identity. By Lemma 36 and other assumptions we get that $\mathcal{C}_S^n[\![\kappa_{S,P,T}]\!]$ admits an insertion along branching point P and so:

$$\begin{aligned}
&\mathcal{T}_S^n[\![\kappa_{S,P,T}]\!] \\
\equiv\ &\mathcal{C}_S^n[\![\kappa_{S,P,T}]\!] \\
\rightsquigarrow\ &\mathsf{coh}\ (S \ll_P T : \mathcal{U}_S^n[\![\kappa_{S,P,T}]\!])[\kappa_{S,P,T} \ll_P \iota_{S,P,T}] \\
\equiv\ &\mathsf{coh}\ (S \ll_P T : \mathcal{U}_S^n[\![\kappa_{S,P,T}]\!])[\mathrm{id}] \\
\rightsquigarrow^*\ &\mathsf{coh}\ (S \ll_P T : \mathcal{U}_{S \ll_P T}^n)[\mathrm{id}] \\
\equiv\ &\mathcal{C}_{S \ll_P T}^n \\
\rightsquigarrow^*\ &\mathcal{T}_{S \ll_P T}^n
\end{aligned}$$

With the second equivalence coming from Lemma 51, the second reduction coming from inductive hypothesis (which is well founded as the proof for types only uses the proof for terms on strictly lower values of n), and the last reduction coming from Lemma 49. $\square$

Lemma 60. *Let (S, P, T) be an insertion point and let $\mathsf{lh}(P) \geqslant \dim(T)$. Further suppose that $a \equiv \mathsf{coh}\ (S : A)[\sigma]$ is not an identity and $P[\![\sigma]\!] \equiv \mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!]$ (which may or may not be an unbiased composite or identity). Then there exists a term s with:*

$$a \rightsquigarrow^* s =_{\dim a} \mathsf{coh}\ (S \ll_P T : A[\![\kappa_{S,P,T}]\!])[\sigma \ll_P \tau]$$

In other words it is possible to insert terms that are not unbiased composites or identities, up to dimension bounded equality.

Proof. We proceed by induction on $\mathsf{lh}(P) - \dim(T)$. If $\mathsf{lh}(P) - \dim(T) = 0$ then $\mathcal{C}_T^{\mathsf{lh}(P)}$ is a composite and so we can perform the usual insertion. We now assume that $\mathsf{lh}(P) > \dim(T)$. We may also assume without loss of generality that $\mathcal{C}_T^{\mathsf{lh}(P)}$ is not an identity, as otherwise it would be immediately insertable. This allows us to perform endo-coherence removal to get:

$$\mathcal{C}_T^{\mathsf{lh}(P)} \rightsquigarrow \mathbb{1}[\![\{\mathcal{T}_T^{\mathsf{lh}(P)-1}, \mathcal{U}_T^{\mathsf{lh}(P)-1}\} \circ \tau]\!]$$

Now suppose $a \equiv \mathsf{coh}\ (S : A)[\sigma]$ and $b \equiv \mathsf{coh}\ (S : A)[\sigma']$ where σ' is the result of applying the above reduction to the appropriate term of σ. Since $P[\![\sigma']\!]$ is now an identity it can be inserted to get $b \rightsquigarrow c$ where:

$$c \equiv \mathsf{coh}\ (S /\!/ P : A[\![\pi_P]\!])[\sigma' \ll_P (\{\mathcal{T}_T^{\mathsf{lh}(P)-1}, \mathcal{U}_T^{\mathsf{lh}(P)-1}\} \circ \tau)]$$

We now wish to show that $2 + \mathsf{bh}(P) \leqslant \mathsf{lh}(P)$ so that P' exists as a branch of $S /\!/ P$. Since we always have $1 + \mathsf{bh}(P) \leqslant \mathsf{lh}(P)$, we consider the case where $1 + \mathsf{bh}(P) = \mathsf{lh}(P)$. We know that $\mathsf{bh}(P) \leqslant \dim(T) \leqslant \mathsf{lh}(P)$ and so must be equal to one of the two. If $\dim(T) = \mathsf{lh}(P)$ then $\mathcal{C}_T^{\mathsf{lh}(P)}$ is an unbiased composite. If $\dim(T) = \mathsf{bh}(P)$ then $\mathsf{th}(T) = \dim(T)$ and so T is linear. However this makes $\mathcal{C}_T^{\mathsf{lh}(P)}$ an identity. Either case is a contradiction and so $2 + \mathsf{bh}(P) \leqslant \mathsf{lh}(P)$ and so P' is a branch of $S /\!/ P$.

By Lemmas 25 and 58, we now have:

$$\begin{aligned}
&P'[\![\sigma' \ll_P (\{\mathcal{T}_T^{\mathsf{lh}(P)-1}, \mathcal{U}_T^{\mathsf{lh}(P)-1}\} \circ \tau)]\!] \\
&\equiv d_{\mathsf{lh}(P)-1}[\![\iota_{S,P,D^{\mathsf{lh}(P)-1}} \circ (\sigma' \ll_P (\{\mathcal{T}_T^{\mathsf{lh}(P)-1}, \mathcal{U}_T^{\mathsf{lh}(P)-1}\} \circ \tau))]\!] \\
&\equiv d_{\mathsf{lh}(P)-1}[\![\{\mathcal{T}_T^{\mathsf{lh}(P)-1}, \mathcal{U}_T^{\mathsf{lh}(P)-1}\} \circ \tau]\!] \\
&\equiv \mathcal{T}_T^{\mathsf{lh}(P)-1}[\![\tau]\!] \\
&\equiv \mathcal{C}_T^{\mathsf{lh}(P)-1}[\![\tau]\!]
\end{aligned}$$

with the last equivalence holding as if $\mathcal{T}_T^{\mathsf{lh}(P)-1}$ was a variable then $\mathcal{C}_T^{\mathsf{lh}(P)}$ would be an identity. As $\mathsf{lh}(P') - \dim(T) = \mathsf{lh}(P) - \dim(T) - 1$ we can use the induction hypothesis to get that $c \rightsquigarrow d$ and:

$$\begin{aligned}
d =_{\dim(a)} \mathsf{coh}\ ((S \mathbin{/\!/} P) \ll_{P'} T : A[\![\pi_P \circ \kappa_{S /\!/ P, P', T}]\!])[\\
(\sigma' \ll_P (\{\mathcal{T}_T^{\mathsf{lh}(P)-1}, \mathcal{U}_T^{\mathsf{lh}(P)-1}\} \circ \tau)) \ll_{P'} \tau]
\end{aligned}$$

By Lemmas 52 and 58,

$$d =_{\dim(a)} \mathsf{coh}\ (S \ll_P T : A[\![\kappa_{S,P,T}]\!])[\sigma \ll_P \tau]$$

which completes the proof as $a \rightsquigarrow^* d$. □

With the above the lemmas we can now recall Theorem 34 and give a proof.

Theorem 34. *Reduction is locally confluent: if a is valid with $a \rightsquigarrow b$ and $a \rightsquigarrow c$, then there exists some d with $b \rightsquigarrow^* d$ and $c \rightsquigarrow^* d$.*

Proof. We proceed by simultaneous induction on subterms and dimension. Suppose $a \rightsquigarrow b$ and $a \rightsquigarrow c$. It is sufficient to show that $b \rightsquigarrow^* b'$, $c \rightsquigarrow^* c'$, and that $b' =_{\dim(a)} c'$, as then by induction on dimension we have that b' and c' have a common reduct, which we can obtain for example by reducing both terms to normal form. We now split on cases on the reductions. We will ignore cases where both reductions are the same and those which are symmetric to a case which has already been covered. We split first on $a \rightsquigarrow b$.

Insertion. Let $a \equiv \mathsf{coh}\ (S : A)[\sigma]$ be not an identity, and $(S, P, T, \Gamma, \sigma, \tau)$ be an insertion redex with $\mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!]$ an unbiased composite or identity. We then have:

$$b \equiv \mathsf{coh}\ (S \ll_P T : A[\![\kappa_{S,P,T}]\!])[\sigma \ll_P \tau]$$

We consider the possible cases for the second reduction.

Insertion. Suppose $a \rightsquigarrow c$ is also an insertion, along a branch Q of S. We now split on whether $\lfloor P \rfloor = \lfloor Q \rfloor$. First suppose $\lfloor P \rfloor = \lfloor Q \rfloor$; then by Lemma 38, we have $b =_{\dim(a)} c$. Suppose now that $\lfloor P \rfloor \neq \lfloor Q \rfloor$, and that $Q[\![\sigma]\!] \equiv \mathcal{C}_U^{\mathsf{lh}(Q)}[\![\mu]\!]$, such that

$$c \equiv \mathsf{coh}\ (S \ll_Q U : A[\![\kappa_{S,Q,U}]\!])[\sigma \ll_Q \mu]$$

We now want b and c to further reduce as follows:

$$b \rightsquigarrow b' \equiv \mathsf{coh}\ ((S \ll_P T) \ll_{Q \ll_P T} U :$$

$$
\begin{aligned}
&A[\![\kappa_{S,P,T} \circ \kappa_{S \ll_P T, Q \ll_P T, U}]\!])[(\sigma \ll_P \tau) \ll_{Q \ll_P T} \mu] \\
c \rightsquigarrow c' \equiv \mathsf{coh}\ &((S \ll_Q U) \ll_{P \ll_Q U} T : \\
&A[\![\kappa_{S,Q,U} \circ \kappa_{S \ll_Q U, P \ll_Q U, T}]\!])[(\sigma \ll_Q \mu) \ll_{P \ll_Q U} \tau]
\end{aligned}
$$

We will show that the first reduction is valid and the other will hold by symmetry. We need b to not be an identity. First suppose that a is typed by the coherence rule. Then if $A \equiv s \to_B t$ we must have $\mathsf{supp}(s) = \mathsf{supp}(S)$. However S is certainly not linear, as it has two distinct leaves. Therefore s cannot be a variable and so $s[\![\kappa_{S,P,T}]\!]$ is also not a variable, making b not an identity. Now suppose instead that a is a composite. Then, if $\partial^-(S)$ is not linear, we can apply the same logic as before, and so S must have linear height one less than its dimension. However, this means that both $\lfloor P \rfloor$ and $\lfloor Q \rfloor$ are distinct variables with the same dimension as the tree, and so $\dim(S \ll_P T) = \dim(S) = \dim(a)$ and so b cannot be an identity.

We also note:

$$
\begin{aligned}
Q \ll_P T[\![\sigma \ll_P \tau]\!] &\equiv Q[\![\kappa]\!][\![\sigma \ll_P \tau]\!] \\
&\equiv Q[\![\sigma]\!] \\
&\equiv \mathcal{C}_U^{\mathsf{lh}(Q)}[\![\mu]\!] \\
&\equiv \mathcal{C}_U^{\mathsf{lh}(Q \ll_P T)}[\![\mu]\!]
\end{aligned}
$$

as required for the insertion, with the third equality coming from Lemma 37. Lastly the trunk height condition is satisfied as $\mathsf{bh}(Q) = \mathsf{bh}(Q \ll_P T)$.

Therefore both reductions are valid insertions and by Lemmas 41 and 42, $b' =_{\dim(a)} c'$.

Cell reduction. If $A \rightsquigarrow B$ and $c \equiv \mathsf{coh}\ (S : B)[\sigma]$ from cell reduction, then if c is not an identity then it admits insertion to reduce to:

$$
c' \equiv \mathsf{coh}\ (S \ll_P T : B[\![\kappa_{S,P,T}]\!])[\sigma \ll_P \tau]
$$

As reduction is compatible with substitution, b also reduces to c'. If instead c was an identity then

$$
\begin{aligned}
b &\equiv \mathsf{coh}\ (D^n \ll_P T : A[\![\kappa_{S,P,T}]\!])[\sigma \ll_P \tau] \\
&\rightsquigarrow \mathsf{coh}\ (D^n \ll_P T : \mathcal{U}_{D^n}^{n+1}[\![\kappa_{S,P,T}]\!])[\sigma \ll_P \tau] \\
&\rightsquigarrow^* \mathbb{1}[\![\{\mathcal{U}_{D^n}^n, d_n\} \circ \kappa_{S,P,T} \circ \sigma \ll_P \tau]\!] \\
&=_{n+1} \mathbb{1}[\![\sigma]\!] \\
&\equiv c
\end{aligned}
$$

Where the second reduction is due to Lemma 43 and the equality is due to Lemma 25 and $\{\mathcal{U}_{D^n}^n, d_n\}$ being the identity substitution.

Disc removal. Suppose $\lfloor S \rfloor = D^n$ and $c \equiv d_n[\![\sigma]\!]$ with $a \rightsquigarrow c$ by disc removal. Then certainly $\lfloor P \rfloor = d_n$ and so $c \equiv d_n[\![\sigma]\!] \equiv \mathcal{C}_T^n[\![\tau]\!]$ and in this case:

$$
b \equiv \mathsf{coh}\ (D^n \ll_P T : \mathcal{U}_{D^n}^n[\![\kappa_{D^n,P,T}]\!])[\sigma \ll_P \tau]
$$

By Lemmas 44 and 59, $b =_{\dim(a)} c$.

Endo-coherence removal:.Suppose $A \equiv s \to_B s$ and $a \rightsquigarrow c$ by endo-coherence removal. In this case $c \equiv \mathbb{1}[\![\{A, s\} \circ \sigma]\!]$ and

$$b \equiv \mathsf{coh}\ (S \ll_P T : (s \to_B s)[\![\kappa_{S,P,T}]\!])[\sigma \ll_P \tau]$$

which reduces by endo-coherence removal to:

$$b' \equiv \mathbb{1}[\![\{s, A\} \circ \kappa_{S,P,T} \circ (\sigma \ll_P \tau)]\!]$$

Finally, by Lemma 25, we have that $\kappa_{S,P,T} \circ (\sigma \ll_P \tau) =_{\dim(S)} \sigma$ and so $b' =_{\dim(S)} c$ and since $\dim(S) \leqslant \dim(a)$, we get $b' =_{\dim(a)} c$ as required.

Reduction of non-inserted argument. Suppose $\sigma \rightsquigarrow \sigma'$ along an argument which is not $\lfloor P \rfloor$ and $c \equiv \mathsf{coh}\ (S : A)[\sigma']$. Then as $P[\![\sigma']\!] \equiv \mathcal{C}_T^{\mathsf{lh}(P)}$, an insertion can still be performed on c to get:

$$c \rightsquigarrow c' \equiv \mathsf{coh}\ (S \ll_P T : A[\![\kappa_{S,P,T}]\!])[\sigma' \ll_P \tau]$$

By Lemma 50, $b \rightsquigarrow^* c'$.

Argument reduction on inserted argument. Suppose $\tau \rightsquigarrow \tau'$, and σ' is σ but with the argument for $\lfloor P \rfloor$ replaced by $\mathsf{coh}\ (T : \mathcal{U}_T^{\mathsf{lh}(P)})[\tau']$, such that $\sigma \rightsquigarrow \sigma'$ and $a \rightsquigarrow c \equiv \mathsf{coh}\ (S : A)[\sigma']$. Then c admits an insertion and reduces as follows:

$$c \rightsquigarrow c' \equiv \mathsf{coh}\ (S \ll_P T : A[\![\kappa_{S,P,T}]\!])[\sigma' \ll_P \tau']$$

By Lemma 50 we then have $b \rightsquigarrow^* c'$.

Disc removal on inserted argument. If $a \rightsquigarrow c$ is the result of reducing $P[\![\sigma]\!]$ by disc removal, then T must equal D^n (with $n = \mathsf{lh}(P)$) and $c \equiv \mathsf{coh}\ (S : A)[\sigma']$ where σ' is σ with the argument for $\lfloor P \rfloor$ replaced with $d_n[\![\tau]\!]$. Further:

$$b \equiv \mathsf{coh}\ (S \ll_P D^n : A[\![\kappa_{S,P,D^n}]\!])[\sigma \ll_P \tau]$$

By Lemma 52 we have $\sigma \ll_P \tau \equiv \sigma' \ll_P \tau$, and furthermore by Lemma 45, $S \ll_P D^n = S$, $\sigma' \ll_P \tau \equiv^{\mathsf{max}} \sigma'$ and $\kappa_{S,P,D^n} = \mathrm{id}$. As $\dim(A[\![\kappa_{S,P,D^d}]\!]) \leqslant \dim(a)$:

$$\begin{aligned} b &\equiv \mathsf{coh}\ (S : A[\![\kappa_{S,P,D^d}]\!])[\sigma' \ll_P \tau] \\ &=_{\dim(a)} \mathsf{coh}\ (S : A)[\sigma'] \\ &\equiv c \end{aligned}$$

Endo-coherence removal on inserted argument. The inserted argument must already be an unbiased composite or identity, so cannot reduce by endo-coherence removal, hence this case is vacuous.

Insertion on inserted argument. Suppose $\mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!]$ admits an insertion, so that there is branch Q of T with $Q[\![\tau]\!] \equiv \mathcal{C}_U^{\mathsf{lh}(Q)}[\![\mu]\!]$ and $\mathsf{th}(U) \geqslant \mathsf{bh}(Q)$. Then:

$$\mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!] \rightsquigarrow \mathsf{coh}\ (T \ll_Q U : \mathcal{U}_T^{\mathsf{lh}(P)}[\![\kappa_{T,Q,U}]\!])[\tau \ll_Q \mu]$$

We then have $c \equiv \mathsf{coh}\ (S : A)[\sigma']$ where σ' is σ with the reduction above applied. We can conclude that $\mathcal{C}_T^{\mathsf{lh}(P)}$ must be a composite (i.e. not an identity) as otherwise the second insertion would not be possible. We now split on whether T is linear.

If T is linear then it must equal $D^{\mathsf{lh}(P)}$. Following the disc removal/insertion confluence case we see that the inner insertion is the same (up to bounded equality) as disc removal. Further, following the insertion/argument disc removal case we get that performing a disc removal on the insertable argument is the same (up to bounded equality) as performing the outer insertion. We can therefore conclude that in this case $b =_{\dim(a)} c$ as required.

We now turn to the case where T is not linear. By Lemma 59, $\mathcal{U}_T^{\mathsf{lh}(P)}[\![\kappa_{T,Q,U}]\!] \rightsquigarrow^* \mathcal{U}_{T \ll_P Q}^{\mathsf{lh}(P)}$ and so $\mathcal{C}_T^{\mathsf{lh}(P)}[\![\tau]\!] \rightsquigarrow^* \mathcal{C}_{T \ll_Q U}^{\mathsf{lh}(P)}[\![\tau \ll_Q \mu]\!]$. Let c' be the term obtained by applying this further reduction to the appropriate argument. Now by Lemma 53, we have that $\mathsf{th}(T \ll_Q U) \geqslant \mathsf{th}(T)$ and so by Lemma 60, there is $c' \rightsquigarrow^* c''$ with:

$$\begin{aligned} c'' =_{\dim(a)} \mathsf{coh}\ (S \ll_P (T \ll_Q U) :\\ A[\![\kappa_{S,P,T \ll_Q U}]\!])[\sigma \ll_P (\tau \ll_Q \mu)] \end{aligned}$$

We now examine how b reduces. As T is not linear, there is a branch $S \ll_P Q$ of $S \ll_P T$ and we get the following by Lemmas 25 and 55:

$$\begin{aligned} S \ll_P Q[\![\sigma \ll_P \tau]\!] &\equiv Q[\![\iota_{S,P,T} \circ (\sigma \ll_P \tau)]\!] \\ &\equiv Q[\![\tau]\!] \\ &\equiv \mathcal{C}_U^{\mathsf{lh}(Q)}[\![\mu]\!] \end{aligned}$$

Since $\mathsf{th}(U) \geqslant \mathsf{bh}(Q) = \mathsf{bh}(S \ll_P Q)$ we can reduce b to b' by insertion as follows:

$$\begin{aligned} b' \equiv \mathsf{coh}\ ((S \ll_P T) \ll_{S \ll_P Q} U :\\ A[\![\kappa_{S,P,T} \circ \kappa_{S \ll_P T, S \ll_P Q, U}]\!])[(\sigma \ll_P \tau) \ll_{S \ll_P Q} \mu] \end{aligned}$$

and then by Lemma 55 we get $b' =_{\dim(a)-1} c''$ as required.

Cell reduction. If $a \rightsquigarrow b$ is an instance of a cell reduction, then $a \equiv \mathsf{coh}\ (\Gamma : A)[\sigma]$, $A \rightsquigarrow B$, and $b \equiv \mathsf{coh}\ (\Gamma : B)[\sigma]$. We now split on the reduction $a \rightsquigarrow c$.

*Cell reduction.*If $a \rightsquigarrow c$ is the result of another cell reduction $A \rightsquigarrow C$ then either the reductions target different parts of the type in which case there is a common reduct D. Otherwise we can appeal to the inductive hypothesis on subterms to find a common reduct.

Disc removal. A is not in normal form, so a cannot have a disc as its head, hence $a \rightsquigarrow c$ cannot be a disc removal.

Endo-coherence removal. Suppose $A \equiv s \to_{A'} s$, and $c \equiv \mathbb{1}[\![\{A', s\} \circ \sigma]\!]$. If the reduction $A \rightsquigarrow B$ arises from $A' \rightsquigarrow B'$ then b immediately admits endo-coherence removal and reduces as follows:

$$b' \equiv \mathbb{1}[\![\{B', s\} \circ \sigma]\!]$$

Then $\{s, A'\} \rightsquigarrow \{s, B'\}$ and so $c \rightsquigarrow^* b'$.

Otherwise we either have the reduction $s \to_{A'} s \rightsquigarrow s' \to_{A'} s$ or $s \to_{A'} s \rightsquigarrow s \to_{A'} s'$. In either case we have $b \rightsquigarrow \mathsf{coh}\ (\Gamma : s' \to_{A'} s')[\sigma]$ and so by endo-coherence removal we get the following:

$$b \rightsquigarrow^* b' \equiv \mathbb{1}[\![\{A', s'\} \circ \sigma]\!]$$

Hence we conclude $c \rightsquigarrow^* b'$ as required.

Argument reduction. If $c \equiv \mathsf{coh}\ (\Gamma : A)[\sigma']$ arises from argument reduction $\sigma \rightsquigarrow \sigma'$ then both b and c reduce to $\mathsf{coh}\ (\Gamma : B)[\sigma']$ by an argument reduction or cell reduction.

Disc removal. If $a \rightsquigarrow b$ is a disc removal then $a \equiv \mathcal{C}^n_{D^n}[\![\sigma]\!]$ for some n and $b \equiv d_n[\![\sigma]\!]$. Now a can not reduce by endo-coherence removal and disc removal is unique so the only remaining case is an argument reduction. Suppose $\sigma \rightsquigarrow \sigma'$ and $c \equiv \mathcal{C}^n_{D^n}[\![\sigma']\!]$, which reduces to $c' \equiv d_n[\![\sigma]\!]'$ by disc removal. If the reduction $\sigma \rightsquigarrow \sigma'$ is along d_n then by definition $c \rightsquigarrow c'$, and otherwise $c \equiv c'$.

Endo-coherence removal. For $a \rightsquigarrow b$ to be an instance of Endo-coherence removal we must have $a \equiv \mathsf{coh}\ (\Gamma : s \to_A s)[\sigma]$ and $b \equiv \mathbb{1}[\![\{A, s\} \circ \sigma]\!]$. The only case remaining for the second reduction is argument reduction, as the rest follow by symmetry or the uniqueness of endo-coherence removal. Therefore let $\sigma \rightsquigarrow \sigma'$ and $c \equiv \mathsf{coh}\ (\Gamma : s \to_A s)[\sigma']$ which reduces to $c' \equiv \mathbb{1}[\![\{A, s\} \circ \sigma']\!]$ by endo-coherence removal. As the transitive closure of reduction respects substitution, $\{A, s\} \circ \sigma \rightsquigarrow^* \{A, s\} \circ \sigma'$ and so $b \rightsquigarrow^* c'$.

Argument reduction. We suppose $a \equiv \mathsf{coh}\ (\Gamma : A)[\sigma]$ and $b \equiv \mathsf{coh}\ (\Gamma : A)[\sigma']$ where σ' is the result of reducing one argument x of σ. The only case left is that $a \rightsquigarrow c$ is also an argument insertion, and so $c \equiv \mathsf{coh}\ (\Gamma : A)[\sigma'']$ with σ'' the result of reducing an argument y of σ. If x does not equal y, then both σ' and σ'' reduce to the substitution where we apply both reductions. Otherwise if $x = y$ then by induction on subterms, σ' and σ'' have a common reduct. $\square$

5 Implementation

We have provided an OCaml implementation of our type theory $\mathsf{Catt}_{\mathsf{sua}}$, and here we show some example use-cases. In each case we indicate the file name where the example can be found. The source for the implementation can be found at:

`https://github.com/ericfinster/catt.io/releases/tag/arxiv-sua`

Triangle Equation. `examples/monoidal.catt`
In a monoidal category, the triangle equation expresses compatibility of the left unitor, right unitor and associator:

$$\begin{tikzcd}
(f \cdot \mathrm{id}) \cdot g \arrow[rr, "\alpha_{f,\mathrm{id},g}"] \arrow[dr, "\rho_f \cdot g"'] & \Downarrow & f \cdot (\mathrm{id} \cdot g) \arrow[dl, "f \cdot \lambda_g"] \\
& f \cdot g &
\end{tikzcd}$$

We express this in our implementation as follows:

```
coh triangle C (x(f)y(g)z) :
     vert (assoc f (id y) g) (horiz (id1 f) (unitor-l g))
   => horiz (unitor-r f) (id1 g)
```

We then normalize it in the appropriate context:

```
normalize {C : Cat} {x :: C} {y :: C} (f :: x => y) {z :: C} (g :: y => z)
   | triangle f g
```

Since the triangle equation is entirely expressed in terms of associator and unitor structure, we would expect it to normalize to the identity in our type theory, and this is what the implementation returns.

Pentagon Equation. `examples/monoidal.catt`
We next consider the pentagon constraint, the second axiom family of a monoidal category:

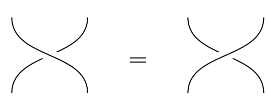
$$\begin{tikzcd}
 & & (f \cdot g) \cdot (h \cdot i) \arrow[drr, "\alpha_{f,g,h \cdot i}"] & & \\
((f \cdot g) \cdot h) \cdot i \arrow[urr, "\alpha_{f \cdot g,h,i}"] \arrow[dr, "\alpha_{f,g,h} \cdot i"'] & & \Downarrow & & f \cdot (g \cdot (h \cdot i)) \\
 & (f \cdot (g \cdot h)) \cdot i \arrow[rr, "\alpha_{f,g \cdot h,i}"'] & & f \cdot ((g \cdot h) \cdot i) \arrow[ur, "f \cdot \alpha_{g,h,i}"'] &
\end{tikzcd}$$

We define this as follows in our implementation:

```
coh pentagon C (v(f)w(g)x(h)y(i)z) :
     vert (assoc (comp f g) h i) (assoc f g (comp h i))
  => vert (horiz (assoc f g h) (id1 i))
          (vert (assoc f (comp g h) i) (horiz (id1 f) (assoc g h i)))
```

Again employing the normalize command, we show that it reduces to the identity as expected.

Syllepsis. `examples/syllepsis.catt`
The syllepsis is a 5-dimensional homotopy which expresses the fact that the overcrossing and undercrossing are equivalent in 4-dimensional space:



It is a fundamental move from low-dimensional topology, and plays an essential role in the homotopy groups of spheres. The bureaucracy of weak higher structures means that it has long been recognized as difficult to describe directly in a formal way, given the extensive use of interchangers, unitors and associators that are required to build it.

Two formal models for the syllepsis were presented at LICS 2022, one using homotopy type theory [16], and an alternative using the type theory $\mathsf{Catt}_{\mathsf{su}}$ [5]. The theory $\mathsf{Catt}_{\mathsf{sua}}$ allows an even shorter representation of the syllepsis, purely in terms of interchanger coherences.

References

[1] Michael Atiyah. Topological quantum field theories. *Publications mathématiques de l'IHÉS*, 68(1):175–186, January 1988. `doi:10.1007/bf02698547`.

[2] Thibaut Benjamin and Samuel Mimram. Suspension et fonctorialité: Deux opérations implicites utiles en catt. In *Journées Francophones des Langages Applicatifs*, 2019. URL: `https://hal.inria.fr/hal-01985195/`, `arXiv:hal-01985195`.

[3] James Dolan. Private communication, 1996.

[4] 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*, 2017. `arXiv:1706.02866`, `doi:10.1109/lics.2017.8005124`.

[5] Eric Finster, David Reutter, Jamie Vicary, and Alex Rice. A type theory for strictly unital ∞-categories. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, 2022. `arXiv:2007.08307`, `doi:10.1145/3531130.3533363`.

[6] R. Gordon, A. J. Power, and Ross Street. Coherence for tricategories. *Mem. Amer. Math. Soc.*, 117(558):vi+81, 1995. `doi:10.1090/memo/0558`.

[7] John W. Gray. *Formal Category Theory: Adjointness for 2-Categories.* Springer Berlin Heidelberg, 1974. `doi:10.1007/bfb0061280`.

[8] Alexander Grothendieck. Pursuing stacks. Unpublished letter to Daniel Quillen, 1983. URL: `https://thescrivener.github.io/PursuingStacks/`.

[9] Martin Hofmann and Thomas Streicher. The groupoid model refutes uniqueness of identity proofs. In *Proceedings of the 9th Annual IEEE Symposium on Logic in Computer Science*, 1994. `doi:10.1109/lics.1994.316071`.

[10] Tom Leinster. *Higher operads, higher categories.* Number 298. Cambridge University Press, 2004. `arXiv:0305049`.

[11] Paolo Lipparini. An infinite natural sum. *Mathematical Logic Quarterly*, 62(3):249–257, 2016. `doi:10.1002/malq.201500017`.

[12] Jacob Lurie. *Higher Topos Theory.* Princeton University Press, 2009. `arXiv:math/0608040`, `doi:10.1515/9781400830558`.

[13] Georges Maltsiniotis. Grothendieck ∞-groupoids, and still another definition of ∞-categories. 2010. `arXiv:1009.2331`.

[14] Christopher Schommer-Pries. *The Classification of Two-Dimensional Extended Topological Field Theories.* PhD thesis, University of California, Berkeley, 2009. `arXiv:1112.1000`.

[15] Carlos Simpson. Homotopy types of strict 3-groupoids. 1998. `arXiv:math/9810059`.

[16] Kristina Sojakova and GA Kavvos. Syllepsis in homotopy type theory. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, 2022. `arXiv:2107.14283`.

[17] Ross Street. Oberwolfach notes on descent theory. 1995. URL: `http://science.mq.edu.au/~street/Descent.pdf`.

[18] Ross Street. The petit topos of globular sets. *Journal of Pure and Applied Algebra*, 154(1-3):299–315, 2000.

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