Higher Categories from Type Theories

PhD thesis

Peter LeFanu Lumsdaine

Doctoral Dissertation

Department of Mathematics

Carnegie Mellon University

Higher Categories from Type Theories

Peter LeFanu Lumsdaine

20 December, 2010

Abstract

This thesis continues the programme of providing a higher-categorical analysis of the treatment of equality in Martin-Löf dependent type theory.

In particular, we construct for various type theories a *classifying weak ω-category*, with objects and 1-cells as in the standard classifying category, and higher cells being open terms of identity types between these. Weak ω-category structures (in the sense of Batanin/Leinster) on these are given by operads of syntactically definable composition laws.

Committee

Steve Awodey (doctoral advisor)

James Cummings (chair)

Richard Garner

Richard Statman

Contents

Acknowledgements vii

Introduction 1

0.1. Background 1

0.2. Overview of results 5

The fundamental weak ω-category of a type (Ch. 3) 5

The classifying weak ω-category of a theory (Ch. 4) 8

Chapter 1. Type-theoretic background 11

1.1. Syntactic presentation 11

1.2. Categorical representation 20

1.3. Constructions on dependent type theories 26

1.4. Miscellaneous facts 29

Chapter 2. Higher-categorical background 31

2.1. Weak factorisation systems 31

2.2. Strict higher categories 34

2.3. Globular operads 40

. . . à la Leinster 40

. . . à la Batanin 44

2.4. Contracting pasting diagrams 50

Chapter 3. The fundamental weak ω-groupoid of a type 53

3.1. Construction of P_{Id} 53

3.2. A proof-theoretic lemma: X is initial in $\mathbf{T}_{\mathsf{Id}}[X]$ 58

3.3. Contractibility of P_{Id} 60

3.4. Types as weak ω-categories 64

Chapter 4. The classifying weak ω-category of a theory 67

4.1. Globular structures from **DTT** 67

The type-theoretic globes 67

A variant for Π-types 74

4.2. Homotopical structures on **DTT** 76

Left and right maps in **DTT** 76

Extensions by propositional copies: the conservativity principle $\overline{J}$. 82

4.3. Contractible operads; weak ω-categories from **DTT** 91

End($\mathsf{G}_\bullet$) in theories with Id-types 91

End($\mathsf{G}_\bullet$) in theories with Π-EXT-APP 93

End(${}^{\Pi}\mathsf{G}_\bullet$) in theories with Id, Π, Π-η 96

Classifying weak ω-categories 96

Bibliography 101

Acknowledgements

This thesis would not have been possible without the support and encouragement of many people over the past few years.

The Pure and Applied Logic Group at Carnegie Mellon, straddling three departments, provided a marvellously stimulating and diverse intellectual environment within which I was able to gradually find my way—enthusiastically but quite unpremeditatedly—to categorical logic and type theory. The teaching, advice, and inspiration of James Cummings and Bob Harper were especially helpful. The unflagging institutional support of the Mathematics department (even when I decided to take an advisor in Philosophy) is also greatly appreciated.

I have also benefited enormously from innumerable discussions with, and support from, colleagues further afield—too many to list, but particularly Martin Hyland, Peter Johnstone, and the other Cambridge category theorists; Thorsten Altenkirch and the type theorists in Nottingham; Mike Shulman and others in Chicago; the Atlantic Category Theory Group at Dalhousie; and, elsewhere, Pierre-Louis Curien and Benno van den Berg.

On a personal level, I would not have made it through a long summer of writing-up without the support of my parents Nicola and David, and a few wonderfully supportive friends—they know who they are!—not to mention UCPO, AUO, the ESO, and the KCO. To all of them, I am deeply indebted.

Particularly influential on the contents of this dissertation have been Richard Garner, Michael A. Warren, and Chris Kapulkin, with whom I have greatly enjoyed working over the past few years.

I am grateful also to all the members of my thesis committee for their interest and support, and for bringing a wide range of interests and expertise to bear on this project. And above all, I owe thanks to my advisor, Steve Awodey: for five years of his unstinting support, his guidance, his teaching, generosity, and patience.

Introduction

0.1. Background

0.1.1. A fundamental construction in all flavours of type theory is the *classifying category* $\mathcal{C}\ell(\mathsf{T})$ of a theory T. Objects of $\mathcal{C}\ell(\mathsf{T})$ are types (or contexts) of T; morphisms of T are open terms of one type with free variables from another, up to syntactic equivalence of some sort.

The uses of this construction are manifold. First of all, it provides a tool for structural analysis of theories, allowing type-theoretic constructors and rules to be understood as presentations of categorical structure (products, sums, exponential objects, ...) familiar from elsewhere in mathematics.

Next, this allows us to model the type theory in any suitably structured category, via a right adjoint Lang to the functor $\mathcal{C}\ell$, giving the *internal language* of a structured category.

Putting these together, it allows us to represent type theories as *equivalent* to appropriately structured categories, giving us an alternate presentation with different strengths to the traditional syntax.

0.1.2. For some logical systems—simple type theory, for instance, and some extensional versions of Martin-Löf type theory—this correspondence is very well understood and leaves little to be desired. Unfortunately, the situation with the intensional variants of Martin-Löf type theory—ITT, henceforth—is not yet so satisfactory.

Syntactic categories for theories in ITT still work essentially as described above: they are a powerful technical tool for providing models, and for representing theories. The subtlety is that the structures corresponding to logical constructors in ITT are no longer so categorically familiar as before:

they are not quite the products, exponentials, adjoints and so on that one might expect. The problem[1] is the interaction of its two different kinds of equality.

Firstly, there is *definitional* (aka *intensional*) equality:

$$\Gamma \ \vdash \ a = a' : A$$

a decidable, quite fine syntactic relation—this covers essentially α-equivalence, β-equivalence, and sometimes η-, but no more. Its strictness reflects the extremely well-controlled computational behaviour of the system. This is the equality used in the classifying category. Crucially, however, it is a separate judgement of the system, not represented by a type, so cannot be reasoned about by e.g. induction within the system. One cannot prove in ITT, for instance, that

$$x : \mathbb{N} \ \vdash \ x + 0 = x \ : \mathbb{N}.$$

Secondly, since one does need some kind of equality which can be reasoned about internally, there is *propositional* (sometimes aka *extensional*) equality, which is represented by a dependent type, the *identity type* over each type:

$$x, y : A \ \vdash \ \mathsf{Id}_A(x, y) \ \mathsf{type}$$

An inhabitant of this type is seen as a *proof of equality* between x and y. This is a coarser equality than the definitional; it is axiomatised simply by INTRO/ELIM rules of the usual form for an inductive type, but they turn out to yield remarkably subtle consequences.

0.1.3. In the extensional theory, the two equalities coincide by fiat (the *reflection* rule). This makes reasoning about equality within the theory much simpler, but at the cost of desirable computational properties: for instance, type-checking becomes undecidable, since it depends (because of the dependency of types on terms) on definitional equality, which is now

[1] although, of course, "it's not a bug, it's a feature"!

reduced to propositional, i.e. to inhabitation of types, which is undecidable in any sufficiently rich system. Relatedly, the computational content of terms is lost: terms of different computational behaviour may become identified in this stronger definitional equality. In these and other proof-theoretic respects, the intensional theory is significantly preferable.

As mentioned above, however, categorical analysis of constructors in the intensional theory is much less clear. For instance, a unit type[2] does *not* give a terminal object of the syntactic category. It may have multiple morphisms from other objects, at least up to definitional equality. Internally, these terms will be equal, but only up to *propositional* equality—so we need to incorporate this somehow into the structure of the syntactic category.

(Of course, propositional equality is already present in the classifying category, but as terms of some other type; but we want a description which makes more apparent its rôle in the categorical analysis of the constructors.)

A naïve approach might be to simply quotient by propositional equality; but this destroys too much of the structure of the theory.

0.1.4. Instead, starting with the Hofmann-Streicher groupoid model [**HS98**], higher categories and related homotopy-theoretic structures have emerged as a natural solution to this problem: we add the propositional equality, but as extra *structure*, not just as a relation to quotient by.

In the globular approach to higher categories, a higher category has objects ("0-cells") and arrows ("1-cells") between objects, as in a category, but also 2-cells between 1-cells, and so on, with various composition operations and laws depending on the kind of category in question (strict or weak, n- or ω-, ...).

Now we see propositional equalities between terms as *2-cells* between morphisms; and also, since propositional equalities themselves are terms of a possibly non-trivial type, we treat further propositional equalities between them as higher cells, and so on *ad infinitum*. Binary composition of these

[2] as usually axiomatised in ITT, i.e. as an inductive type with one nullary constructor.

higher cells corresponds now to the transitivity of equality; higher identity cells, to reflexivity; symmetry of equality, to having some sort of inverses for cells.

There is a compelling analogy here with homotopy theory. Any topological space has a *higher fundamental groupoid* consisting of points of the space, paths between points, homotopies between paths, … Homotopies are composable, and have units, and inverses. More generally, the (1-)category **Top** underlies an ω-category of spaces, continuous maps, homotopies between maps, homotopies between homotopies, and so on.

A type, therefore, can be seen as something like a space—a category with cells of arbitrarily high dimension, in which all cells are invertible, i.e. an *ω-groupoid*—and a theory as something like a category of spaces—an ω-category in which all cells of dimension > 1 are invertible, i.e. an *$(\omega, 1)$-category* (also known as $(\infty, 1)$).

However, in the fundamental groupoid example, it is a familiar fact that associativity of composition, and other laws, hold only up to homotopy, i.e. only up to a cell of higher dimension. We find the same situation in type theory: composition is generally associative, unital etc. only up to propositional equality. (Of course, in the presence of extra axioms, or in certain models, it can sometimes be strict.) For the fully intensional theory we thus have to work not with strict but with *weak* higher categories.

Various different definitions of weak ω-categories exist; we use the *globular operadic* definition of Batanin, as modified by Leinster.

0.1.5. Since [**HS98**], ideas along these lines have been explored by various authors in various directions. For "two-dimensional ITT" and certain weak 2-categories, the full correspondence is worked out in detail in [**Gar09c**]. In the semantic direction, models of ITT in various higher categorical/homotopical settings are investigated in [**AW09**], [**War08**], [**GvdB10a**], and [**Voe**]. Conversely, the structures formed from the syntax of theories have been previously investigated in [**GG08**], [**Lum10**], [**GvdB10b**], [**AHW09**]. This dissertation continues the latter line of investigation: the main goal is to construct the *classifying weak ω-category* of a theory.

0.2. Overview of results

The fundamental weak ω-category of a type (Ch. 3).

0.2.1. As a warm-up for the main result, in this chapter we construct the "fundamental weak ω-category"[3] of a single type within a theory.

Specifically, we show that for any type A and context Δ in a type theory T with at least the Id-type rules, there is a weak ω-category in which 0-cells are terms of type A in context Δ, 1-cells are terms of type Id_A between these, 2-cells are terms of type $\mathsf{Id}_{\mathsf{Id}_A}$ between 1-cells, and so on.

(Note that for this construction the dimensions of cells are always one lower than in the classifying category sketched above. This comes from the general rule that if X, A are objects of an n-category $\mathcal{C}$, then $\mathcal{C}(X, A)$ forms an $(n-1)$-category, whose 0-cells are 1-cells of $\mathcal{C}$, and so on.)

0.2.2. Before sketching the construction, we roughly recall the globular operadic definition of weak higher categories. (The full background required on this material is set out in Ch. 2.)

An ω-category $\mathcal{C}$ has a set[4] C_n of "n-cells" for each $n > 0$. The 0- and 1-cells correspond to the objects and arrows of an ordinary category: each arrow f has source and target objects $a = s(f)$, $b = t(f)$. Similarly, the source and target of a 2-cell α are a parallel pair of 1-cells $f, g : a \rightrightarrows b$, and generally the source and target of an $(n+1)$-cell are a parallel pair of n-cells.

Cells of each dimension can be composed along a common boundary in any lower dimension, and in a *strict* ω-category, the composition satisfies

[3] named by analogy with the higher fundamental groupoids of a space; in our case the fundamental weak ω-category is again groupoidal, as shown in [**GvdB10b**], but we do not discuss this in the present work.

[4] for the present work, we consider *small* higher categories only.

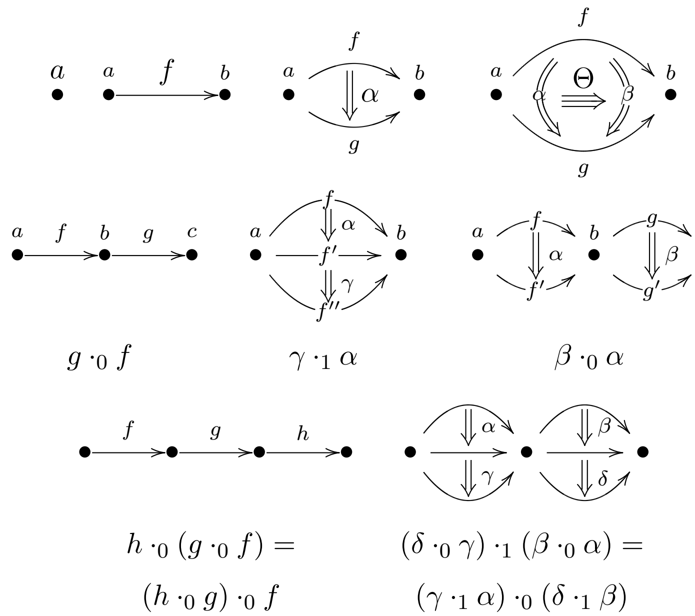


FIGURE 0.2.A. Some cells, composites, and associativities in a strict ω-category

various associativity, unit, and interchange laws, captured by the *generalised associativity* law: each labelled pasting diagram has a unique composite. (See illustrations in Fig. 0.2.A).

0.2.3. In a *weak* ω-category, we do not expect strict associativity, so may have multiple composites for a given pasting diagram, but we do demand that these composites agree up to cells of the next dimension ("up to homotopy"), and that these associativity cells satisfy certain coherence laws of their own, again up to cells of higher dimension, and so on. This is exactly the situation we find in intensional type theory. For instance, even in constructing a term witnessing the transitivity of identity—that is, a composition law for the pasting diagram ($\bullet \longrightarrow \bullet \longrightarrow \bullet$), or explicitly a term c such that

$$x, y, z : X,\ \ p : \mathsf{Id}(x, y),\ \ q : \mathsf{Id}(y, z) \vdash c(q, p) : \mathsf{Id}(x, z)$$

—one finds that there is no single canonical candidate: most obvious are the two equally natural terms c_l, c_r obtained by applying the Id-elimination rule to p or to q respectively. These are not definitionally equal, but are propositionally equal, i.e. equal up to a 2-cell: there is a term e with

$$x, y, z : X,\ p : \mathsf{Id}(x, y),\ q : \mathsf{Id}(y, z) \vdash e(q, p) : \mathsf{Id}(c_l(q, p), c_r(q, p)).$$

Similarly, for either of these operations (or any combination), we can derive a "propositional associative law":

$$x, y, z, w : X,\ p : \mathsf{Id}(x, y),\ q : \mathsf{Id}(y, z),\ r : \mathsf{Id}(z, w)\ \vdash$$
$$a(r, q, p) : \mathsf{Id}(c(r, c(q, p))\,,\ c(c(r, q), p)).$$

0.2.4. In Leinster's definition [**Lei04**], a system of composition laws of this sort is wrapped up in the algebraic structure of a *globular operad with contraction*, and a weak ω-category is given by a globular set equipped with an *action* of such an operad. We generalise this slightly, to define an *internal weak ω-category* in any suitable category $\mathcal{C}$.

Accordingly, we would like to find an operad-with-contraction P_{Id} of all such type-theoretically definable composition laws, acting on terms of any type and its identity types. Formally, we consider $\mathbf{T}_{\mathsf{Id}}[X]$, the theory axiomatised just by the structural and Id-rules plus a single generic base type X. The operad P_{Id} of definable composition laws therein may then be formally constructed as an endomorphism operad in its syntactic category $\mathcal{C}\ell(\mathbf{T}_{\mathsf{Id}}[X])$; and by some analysis of $\mathbf{T}_{\mathsf{Id}}[X]$, we show that P_{Id} is contractible.

Since X is generic, composition laws defined on it can be implemented on all other types, giving the main result of Chapter 3:

Theorem 3.4.1. *Let $\mathbf{T}$ be any type theory extending $\mathbf{T}_{\mathsf{Id}}$, and A any type of $\mathbf{T}$. Then the system of types $(A, \mathsf{Id}_A, \mathsf{Id}_{\mathsf{Id}_A}, \ldots)$ is equipped naturally with a P_{Id}-action, and hence with the structure of an internal weak ω-category in $\mathcal{C}\ell(\mathbf{T})$.*

From this it follows that the terms of these types in any fixed context form an (external) P_{Id}-algebra and weak ω-category. Indeed, one can construct, using the Id-rules, *identity contexts* Id_Γ over all contexts, satisfying rules analogous to those for identity types; so for any contexts Δ, Γ, there is an ω-category of context morphisms from Δ into Γ and its identity types.

The material of this chapter is essentially based on my earlier paper [**Lum09**]/[**Lum10**]. A very similar construction (overlapping in some parts and using different techniques in others) was discovered independently by Benno van den Berg and Richard Garner, and appears in [**GvdB10b**].

The classifying weak ω-category of a theory (Ch. 4).

0.2.5. In this chapter, the heart of the dissertation, we construct the classifying ω-category $\mathcal{C}\ell_\omega(\mathbf{T})$ of a theory T.

As intimated above, $\mathcal{C}\ell_\omega(\mathbf{T})$ should in dimensions ≤ 1 be just the usual classifying category $\mathcal{C}\ell(\mathbf{T})$ of contexts and context morphisms; its 2-cells should be morphisms into identity contexts, and higher cells should be maps into higher identity contexts. From another point of view, recall that as a category has hom-sets, a 2-category has hom-categories, and generally a $(1+n)$-category has hom-n-categories between its 0-cells. Then the hom-ω-category $\mathcal{C}\ell_\omega(\mathbf{T})(\Delta, \Gamma)$ is just the fundamental ω-category of Γ in context Δ, as sketched in the previous section.

In fact we first construct a weak ω-category $\mathcal{C}\ell_\omega^-(\mathbf{T})$, as above but with 0-cells just types, not more general contexts; then, we bump this up formally to include contexts as well.

The core technique used is analogous again to the familiar construction of the higher fundamental group(oid)s of a space: *representability*. Points, paths, homotopies, … in a space X are constructed as maps into X from the point, the interval, the 1-disc, … ; correspondingly, the cells of $\mathcal{C}\ell_\omega^-(\mathbf{T})$ can be seen as interpretations in $\mathbf{T}$ of the theories of a free type, a free open term, … —maps into T from the *type-theoretic globes*, which we denote $\mathsf{G}_\bullet$. (This is why we start with $\mathcal{C}\ell_\omega^-$ not $\mathcal{C}\ell_\omega$: there is no "theory of a free context", since contexts can have various lengths.)

Then, as usual with representable algebraic structures[5], the weak ω-category structure on $\mathcal{C}\ell_\omega(\mathsf{T})$ is induced formally from a weak ω-cocategory structure on the representing objects $\mathsf{G}_\bullet$.

To give this structure, we once again consider an endomorphism operad—in fact now the *co-endomorphism* operad of the globes, whose operations now represent composition laws that can be defined generically over multiple types and morphisms between them, and we show this is contractible by techniques extending those of Ch. 3.

0.2.6. To carry this out, we require two main additional pieces of technical machinery. Firstly, to handle the representing objects, co-endomorphism operads, and so on, we need a well-behaved category **DTT** of type theories in which to work. In fact, we consider various such categories $\mathbf{DTT}_\Phi$, given by fixing some set Φ of constructors and rules (Id-types, Π-types, etc.), and considering all theories extending these purely algebraically.

This type-theoretic background is set out in Ch. 1. In order to facilitate the categorical analysis of **DTT**, we recall the representation of theories as certain *categories with attributes*, and the equivalence of this with more traditional syntactic presentations.

On the other hand, for showing contractibility of the operads in these categories, we need analogues in $\mathbf{DTT}_\Phi$ of Id-elimination in syntactic categories. To this end, we introduce a weak factorisation system in $\mathbf{DTT}_\Phi$, of *term-extensions* and *term-contractible maps*, and a type-theoretic principle $\overline{J}$, giving (in terms of these maps) the required analogue in $\mathbf{DTT}_\Phi$ of the identity-type eliminator J.

---

[5] in fact inevitably, by the Yoneda lemma.

0.2.7. This leads us to the main restrictions on our eventual result. As an external form of J, and particularly as one concerning propositional identity of open terms, it comes as no surprise that $\overline{J}$ is intimately related to the various *functional extensionality* rules that can be imposed in ITT.

(Despite the names, these are not to be confused with the extra rules of *extensional type theory* discussed earlier, which trivialise identity types. These rules just force the Id-types over function types to be *extensional* in the sense that two functions are propositionally equal if they provably give propositionally equal outputs.)

In particular, we show that $\overline{J}$ (and hence our construction of a contractible operad) holds in the category of theories with suitably extensional Π-types, but does *not* hold in the category of theories with Π-types but without extensionality rules.

We conjecture that in fact $\overline{J}$ holds in $\mathbf{DTT}_{\mathsf{Id}}$, and hence that a contractible operad acting on $\mathcal{C}\ell_\omega$ may be constructed from Id-types alone ; but we are unable to prove this.

Our main eventual result is thus:

Theorem 4.3.8.

(1) *There is a functor* $\mathcal{C}\ell_\omega \colon \mathbf{DTT}_{\Pi\text{-}\mathsf{Id}\text{-}\textsc{elim}} \longrightarrow \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat}$*, as outlined above, giving the "classifying weak* ω*-category" of any theory with at least* Id*-types,* Π*-types, and the* Π-EXT, Π-EXT-APP *rules.*

(2) *If* $\overline{J}$ *holds for* Id*, then we moreover have* $\mathcal{C}\ell_\omega \colon \mathbf{DTT}_{\mathsf{Id}} \longrightarrow \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat}$*, giving the classifying weak* ω*-category for any theory with at least* Id*-types.*

Additionally, we give a variant construction ${}^{\Pi}\mathcal{C}\ell_\omega$, with a slightly different underlying set (using closed terms of Π-types, rather than open terms), which works for theories with just Id-types, Π-types, and the η rule for Π-types, giving an alternative candidate for the classifying ω-categories of these theories in lieu of a proof of $\overline{J}$ for $\mathbf{DTT}_{\mathsf{Id}}$.

CHAPTER 1

Type-theoretic background

1.1. Syntactic presentation

1.1.1. Several different, mostly equivalent, syntactic presentations of Martin-Löf-style dependent type theory exist in the literature; the present one is essentially based (up to notation) on those of [**Pit00**] and [**Hof97**].

A word about the range of theories we will consider is, however, in order here. All theories we consider will share the common basic syntax and structural core presented in 1.1.2 and 1.1.3. We will, however, vary the constructors we add on top of that, in two stages. Firstly, by a *type system* we will mean the extension of this structural core by some selection Φ of the constructors and rules of 1.1.4–1.1.7, and of other standard constructors and rules, within certain limitations—a precise definition will appear in 1.2.11. Secondly, over a given such system Φ, we will consider arbitrary extensions by *algebraic* axioms (1.1.8), organised into a category $\mathbf{DTT}_\Phi$ of *theories over* Φ.

1.1.2 (Basic syntax)**.** Since we will eventually work in a presentation-agnostic category of type theories, the precise formalism we use for the raw syntax will not be of importance; but for the sake of definiteness, let us suppose a simply-typed metalanguage in which the syntax of our theory is formalised, as described in e.g. [**Pit00**, 6.1], and in which we have defined notions of free variables, capture-avoiding substitution, etc.

The most essential judgements in dependent type theory are of *types*, *terms*, and *definitional equalities* between each of these:

$$\Gamma \vdash A \ \mathsf{type} \qquad\qquad \Gamma \vdash a : A$$
$$\Gamma \vdash A = A' \ \mathsf{type} \qquad\qquad \Gamma \vdash a = a' : A$$

We will moreover take as basic[1] *contexts* and *context morphisms*[2] between them:

$$\vdash \Gamma \ \mathsf{cxt} \qquad\qquad \vdash f : \Gamma \Rightarrow \Gamma'$$
$$\vdash \Gamma = \Gamma' \ \mathsf{cxt} \qquad\qquad \vdash f = f' : \Gamma \Rightarrow \Gamma'$$

(In all these judgements, we put the obvious restrictions on the free variables of the objects involved.)

We will also use derived judgements of *dependent contexts* and *dependent elements* of these:

$$\Gamma \vdash \Delta \ \mathsf{cxt} \qquad\qquad \Gamma \vdash \vec{d} : \Delta$$
$$\Gamma \vdash \Delta = \Delta' \ \mathsf{cxt} \qquad\qquad \Gamma \vdash \vec{d} = \vec{d'} : \Delta$$

which formally we will consider as syntactic sugar for $\vdash \Gamma, \Delta \ \mathsf{cxt}$ and for $(x_1, \ldots, x_k, \vec{d}) \colon \Gamma \longrightarrow \Gamma, \Delta$ respectively (where $x_1, \ldots, x_k$ are the variables of Γ).

In all the judgements above, in slight abuse of notation, we will often explicitly display free variables for emphasis or readability, writing for instance $\vec{x} : \Gamma \vdash A(\vec{x}) \ \mathsf{type}$ interchangeably with $\Gamma \vdash A \ \mathsf{type}$.

Also, we will often abbreviate multiple judgements of the same form by writing e.g. $\Gamma \vdash a, a' : A$.

Finally, beyond the initial presentation of the theory, context morphisms will usually be written not $\vdash f : \Gamma \Rightarrow \Gamma'$ as above, but $f \colon \Gamma \longrightarrow \Gamma'$, reflecting the view of them as maps of the syntactic category of a theory.

1.1.3 (Structural core)**.** Rules for contexts:

$$\frac{}{\vdash \diamond \ \mathsf{cxt}}\ \mathsf{cxt}\text{-}\textsc{empty} \qquad\qquad \frac{}{\vdash \diamond = \diamond \ \mathsf{cxt}}\ \mathsf{cxt{=}}\text{-}\textsc{empty}$$

[1] in some presentations, these are considered as derived judgements

[2] also sometimes known as *substitutions*, or as closed *telescopes*

$$\frac{\vdash \Gamma \ \mathsf{cxt} \qquad \Gamma \vdash A \ \mathsf{type}}{\vdash \Gamma, y : A \ \mathsf{cxt}}\ \mathsf{cxt}\text{-}\textsc{cons} \qquad \frac{\vdash \Gamma = \Gamma' \ \mathsf{cxt} \qquad \Gamma \vdash A = A'[\vec{x}/\vec{x}'] \ \mathsf{type}}{\vdash \Gamma, y : A = \Gamma', y' : A' \ \mathsf{cxt}}\ \mathsf{cxt}{=}\text{-}\textsc{cons}$$

Rules for types:

$$\frac{\Gamma \vdash A \ \mathsf{type} \qquad \vdash f : \Gamma' \Rightarrow \Gamma}{\Gamma' \vdash A[f/\vec{x}] \ \mathsf{type}}\ \mathsf{type}\text{-}\textsc{subst} \qquad \frac{\Gamma \vdash A = A' \ \mathsf{type} \qquad \vdash f = f' : \Delta \Rightarrow \Gamma}{\Delta \vdash A[f/\vec{x}] = A'[f'/\vec{x}] \ \mathsf{type}}\ \mathsf{type}{=}\text{-}\textsc{subst}$$

$$\frac{\Gamma \vdash A \ \mathsf{type}}{\Gamma \vdash A = A \ \mathsf{type}}\ \mathsf{type}{=}\text{-}\textsc{refl} \qquad \frac{\Gamma \vdash A = B \ \mathsf{type}}{\Gamma \vdash B = A \ \mathsf{type}}\ \mathsf{type}{=}\text{-}\textsc{sym}$$

$$\frac{\Gamma \vdash A = B \ \mathsf{type} \qquad \Gamma \vdash B = C \ \mathsf{type}}{\Gamma \vdash A = C \ \mathsf{type}}\ \mathsf{type}{=}\text{-}\textsc{trans}$$

Rules for terms:

$$\frac{\Gamma \vdash A \ \mathsf{type} \qquad \Gamma, A \vdash \Delta \ \mathsf{cxt}}{\Gamma, x : A, \Delta \vdash x : A}\ \textsc{var}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash A = A' \ \mathsf{type}}{\Gamma \vdash a : A'}\ \mathsf{term}\text{-}\textsc{coerce} \qquad \frac{\Gamma \vdash a = a' : A \qquad \Gamma \vdash A = A' \ \mathsf{type}}{\Gamma \vdash a = a' : A'}\ \mathsf{term}{=}\text{-}\textsc{coerce}$$

$$\frac{\Gamma \vdash a : A \qquad \vdash f : \Gamma' \Rightarrow \Gamma}{\Gamma' \vdash a[f/\vec{x}] : A[f/\vec{x}]}\ \mathsf{term}\text{-}\textsc{subst} \qquad \frac{\Gamma \vdash a = a' : A \qquad \vdash f = f' : \Gamma' \Rightarrow \Gamma}{\Gamma' \vdash a[f/\vec{x}] = a'[f'/\vec{x}] : A[f/\vec{x}]}\ \mathsf{term}{=}\text{-}\textsc{subst}$$

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash a = a : A}\ \mathsf{term}{=}\text{-}\textsc{refl} \qquad \frac{\Gamma \vdash a = b : A}{\Gamma \vdash b = a : A}\ \mathsf{term}{=}\text{-}\textsc{sym}$$

$$\frac{\Gamma \vdash a = b : A \qquad \Gamma \vdash b = c : A}{\Gamma \vdash a = c : A}\ \textsf{term}{=}\text{-}\textsc{trans}$$

Rules for context maps:

$$\frac{\vdash \Gamma\ \textsf{cxt}}{\vdash \diamond : \Gamma \Rightarrow \diamond}\ \textsf{cxtmap-}\textsc{empty} \qquad \frac{\vdash \Gamma\ \textsf{cxt}}{\vdash \diamond = \diamond : \Gamma \Rightarrow \diamond}\ \textsf{cxtmap}{=}\text{-}\textsc{empty}$$

$$\frac{\vdash f : \Gamma' \Rightarrow \Gamma \quad \Gamma \vdash A\ \textsf{type} \quad \Gamma' \vdash a : A[f/\vec{x}]}{\vdash f, a : \Gamma' \Rightarrow \Gamma, y : A}\ \textsf{cxtmap-}\textsc{cons} \qquad \frac{\vdash f = f' : \Gamma' \Rightarrow \Gamma \quad \Gamma \vdash A\ \textsf{type} \quad \Gamma' \vdash a = a' : A[f/\vec{x}]}{\Gamma \vdash (f, a) = (f', a') : \Delta' \Rightarrow (\Delta, A)}\ \textsf{cxtmap}{=}\text{-}\textsc{cons}$$

From these we can derive other structural rules sometimes taken as basic: exchange (EXCH), weakening (WKG), and so on.

1.1.4 (Type constructors)**.** Core rules for Id-types:

$$\frac{\Gamma \vdash A\ \textsf{type} \quad \Gamma \vdash a, b : A}{\Gamma \vdash \mathsf{Id}_A(a, b)\ \textsf{type}}\ \mathsf{Id}\text{-}\textsc{form} \qquad \frac{\Gamma \vdash A\ \textsf{type} \quad \Gamma \vdash a : A}{\Gamma \vdash r(a) : \mathsf{Id}_A(a, a)}\ \mathsf{Id}\text{-}\textsc{intro}$$

$$\frac{\begin{array}{c}\Gamma,\ x, y : A,\ u : \mathsf{Id}_A(x, y),\ \vec{w} : \Delta(x, y, u)\ \vdash\ C(x, y, u, \vec{w})\ \textsf{type} \\ \Gamma,\ z : A,\ \vec{v} : \Delta(z, z, r(z))\ \vdash\ d(z, \vec{v}) : C(z, z, r(z), \vec{v}) \\ \Gamma \vdash a, b : A \qquad \Gamma \vdash p : \mathsf{Id}_A(a, b) \qquad \Gamma \vdash \vec{c} : \Delta(a, b, p)\end{array}}{\Gamma\ \vdash\ \mathsf{J}_{(A;\ x,y,u.\,\Delta(x,y,u);\ x,y,u,\vec{w}.\,C(x,y,u,\vec{w}))}(z, \vec{v}.\ d(z, \vec{v});\ a, b, p, \vec{c})\ : C(a, b, p, \vec{c})}\ \mathsf{Id}\text{-}\textsc{elim}$$

Here all free variables are displayed, to emphasise the full formal binding that occurs in terms involving J; we will usually abbreviate the concluding term above to e.g. $\mathsf{J}_C(d;\ a, b, p, \vec{c})$, as for instance:

$$\frac{\begin{array}{c}\Gamma,\ x, y : A,\ u : \mathsf{Id}_A(x, y),\ \vec{w} : \Delta(x, y, u)\ \vdash\ C(x, y, u, \vec{w})\ \textsf{type} \\ \Gamma,\ z : A,\ \vec{v} : \Delta(z, z, r(z))\ \vdash\ d(z, \vec{v}) : C(z, z, r(z), \vec{v}) \\ \Gamma \vdash a : A \qquad \Gamma \vdash \vec{c} : \Delta(a, a, r(a))\end{array}}{\Gamma\ \vdash\ \mathsf{J}_C(d;\ a, a, r(a), \vec{c}) = d(a, \vec{c}) : C(a, a, r(a), \vec{c})}\ \mathsf{Id}\text{-}\textsc{comp}$$

Finally, two more rules are needed, asserting that the constructors Id and J respect definitional equality in all their arguments. (In the interests of economy, it is enough to specify this just for type arguments and for those term arguments in J binds variables; stability in the unbound term arguments follows from the =-SUBST rules.)

1.1.5. Core rules for Π-types:

$$\frac{\Gamma \vdash A \ \mathsf{type} \quad \Gamma,\, x : A \vdash B \ \mathsf{type}}{\Gamma \ \vdash \ \Pi_{x:A} B \ \mathsf{type}}\ \Pi\text{-}\textsc{form} \qquad \frac{\Gamma,\, x : A \vdash b : B}{\Gamma \ \vdash \ \lambda x\!:\!A.\ b : \Pi_{x:A} B}\ \Pi\text{-}\textsc{intro}$$

$$\frac{\Gamma \vdash t : \Pi_{x:A} B \quad \Gamma \vdash a : A}{\Gamma \ \vdash \ t \cdot a : B[a/x]}\ \Pi\text{-}\textsc{app} \frac{\Gamma,\, x : A \vdash b : B \quad \Gamma \vdash a : A}{\Gamma \ \vdash \ (\lambda x\!:\!A.\ b) \cdot a = b[a/x] : B[a/x]}\ \Pi\text{-}\beta$$

As with Id-types, rules are also required specifying that all the constructors respect definitional equality. Similar such rules are required for the constructors in the following section; we omit further mention of them since their statements are completely routine.

1.1.6 (Variant rules for Id-types)**.** Two rules which will not be part of our core concern but which are worth mentioning here are the stronger eliminators for identity types: the *reflection principle* of extensional type theory (which collapses propositional and definitional equality), and Thomas Streicher's eliminator K (introduced in [**Str93**]):

$$\frac{\Gamma \vdash e : \mathsf{Id}_A(a, b)}{\Gamma \vdash a = b : A}\ \textsc{reflection}$$

$$\frac{\begin{array}{c} \Gamma, x : A, u : \mathsf{Id}_A(x,x), \vec{w} : \Delta(x,u) \vdash C(x,u,\vec{w}) \ \mathsf{type} \\ \Gamma, z : A, \vec{v} : \Delta(z, r(z)) \vdash d(z,\vec{v}) : C(z, r(z), \vec{v}) \\ \Gamma \vdash a : A \qquad \Gamma \vdash p : \mathsf{Id}_A(a,a) \qquad \Gamma \vdash \vec{c} : \Delta(a,p) \end{array}}{\Gamma \vdash \mathsf{K}_{A;\ x,u.\,\Delta;\ x,u,\vec{w}.\,C}(z,\vec{v}.\,d(z,\vec{v});\ a, p, \vec{c}) : C(a,p,\vec{c})}\ \mathsf{K}$$

$$\frac{\begin{array}{c}\Gamma, x : A, u : \mathsf{Id}_A(x,x), \vec{w} : \Delta(x,u) \vdash C(x,u,\vec{w}) \ \mathsf{type} \\ \Gamma, z : A, \vec{v} : \Delta(z, r(z)) \vdash d(z,\vec{v}) : C(z, r(z), \vec{v}) \\ \Gamma \vdash a : A \qquad \Gamma \vdash \vec{c} : \Delta(a, r(a))\end{array}}{\Gamma \vdash \mathsf{K}_C(d;\ a, r(a), \vec{c}) = d(a) : C(a, r(a), \vec{c})}\ \mathsf{K}\text{-}\textsc{comp}$$

Both these eliminators essentially trivialise the higher-categorical structure with which we are principally concerned, so we will mainly consider theories without them; however, they will provide interesting comparisons at times. In the case of the reflection rule, this trivialisation is immediate. In the case of K, it is slightly less obvious; but it turns out that from K we can derive the "(propositional) uniqueness of identity proofs" principle, asserting that all elements of any identity type are equal ([**Str93**], [**War08**]). See 4.2.20 for a "topological" point of view on the relationship between J and K.

Two less destructive variations on the Id-ELIM rule are also worth mentioning; we will not use these as rules per se, but in Section 4.2 will discuss analogous variations on the principle $\overline{J}$.

Firstly, the dependent context $\Delta(x, y, u)$ which we have included in the premises is often omitted. (Its inclusion is, in categorical terms, a Frobenius condition.) In the presence of Π-types, this is interderivable with our version (Δ can simply be curried over to the right-hand side). In the absence of Π-types, the non-Frobenius version of the rule is simply not strong enough to be of much use at all: one cannot even derive, for instance, the transitivity of propositional equality.

Secondly, there is a "one-ended" form of Id-ELIM (in contrast to which the version above may be seen as "two-ended"):

$$\dfrac{\begin{array}{ll} \Gamma \vdash a : A & \Gamma,\, y : A,\, u : \mathsf{Id}_A(a, y),\, \vec{w} : \Delta(y, u) \;\vdash\; C(y, u, \vec{w})\ \mathsf{type} \\ & \Gamma,\, \vec{v} : \Delta(a, r(a)) \;\vdash\; d(\vec{v}) : C(a, r(a), \vec{v}) \\ \multicolumn{2}{c}{\Gamma \vdash b : A \qquad \Gamma \vdash p : \mathsf{Id}_A(a, b) \qquad \Gamma \vdash \vec{c} : \Delta(b, p)} \end{array}}{\Gamma \;\vdash\; \mathsf{J}^1_C(d;\, b, p, \vec{c}) \;:\; C(b, p, \vec{c})}\ \mathsf{Id}\text{-}\textsc{elim}^1$$

and with computation rule concluding $\mathsf{J}^1_C(d;\, a, r(a), \vec{c}) = d(\vec{c})$.

This easily implies the original form; and with a little more effort (using the identity contexts of 1.3.1 below), they are in fact inter-derivable. This is originally due to Christine Paulin-Mohring, as discussed in [**Str93**].

1.1.7 (Functional extensionality and η-rules)**.** It turns out that with just the core rules of 1.1.4, propositional equality on Π-types is a rather more exclusive relation than one might desire. Various different rules have been considered relaxing it, and in particular relating to the principle of "functional extensionality": that two functions are equal if they are equal on values. A systematic comparison is given in [**Gar09b**], the terminology of which we follow here. Several rules asserting equalities may be given in both propositional and definitional flavours.

$$\dfrac{\Gamma \vdash f : \Pi_{x:A} B}{\Gamma \vdash f = (\lambda x.\, f \cdot x) : \Pi_{x:A} B}\ \Pi\text{-}\eta \qquad \dfrac{\Gamma \vdash f : \Pi_{x:A} B}{\Gamma \vdash \eta(f) : \mathsf{Id}_{\Pi_{x:A} B}(f, (\lambda x.\ f \cdot x))}\ \Pi\text{-}\textsc{prop}\text{-}\eta$$

$$\dfrac{\Gamma,\, x : A \vdash b(x) : B(x)}{\Gamma \vdash \eta(\lambda x.\, b(x)) = r(\lambda x.\, b(x)) : \mathsf{Id}_{\Pi_{x:A} B}(\lambda x.\, b(x), \lambda x.\, b(x))}\ \Pi\text{-}\textsc{prop}\text{-}\eta\text{-}\textsc{comp}$$

$$\dfrac{\Gamma \vdash f, g : \Pi_{x:A} B \qquad \Gamma \vdash k : \Pi_{x:A} \mathsf{Id}_B(f \cdot x, g \cdot x)}{\Gamma \vdash \mathsf{ext}(f, g, k) : \mathsf{Id}_{\Pi_A B}(f, g)}\ \Pi\text{-}\textsc{ext}$$

$$\frac{\Gamma,\, x : A \vdash b : B}{\Gamma \vdash \mathsf{ext}(\lambda x.\, b, \lambda x.\, b, \lambda x.\, r(b)) = r(\lambda x.\, b) : \mathsf{Id}(\lambda x.\, b, \lambda x.\, b)}\ \Pi\text{-}\textsc{ext-comp}$$

$$\frac{\Gamma \vdash f, g : \Pi_{x:A} B \qquad \Gamma \vdash k : \Pi_{x:A} \mathsf{Id}_B(f \cdot x, g \cdot x) \qquad \Gamma \vdash a : A}{\Gamma \vdash \mu(f, g, k, a) : \mathsf{Id}(\mathsf{ext}(f, g, k) \star a, k \cdot a)}\ \Pi\text{-}\textsc{ext-app}$$

$$\frac{\Gamma,\, x : A \vdash b(x) : B(x) \qquad \Gamma \vdash a : A}{\Gamma \vdash \mu(\lambda x. b(x), \lambda x. b(x), \lambda x. r b(x), a) = r(r(b(a))) : \mathsf{Id}(r(b(a)), r(b(a)))}\ \Pi\text{-}\textsc{ext-app-comp}$$

$$\frac{\begin{array}{c}\Gamma \vdash f, g : \Pi_{x:A} B \\ \Gamma \vdash k : \Pi_{x:A} \mathsf{Id}_B(f \cdot x, g \cdot x) \qquad \Gamma \vdash a : A\end{array}}{\Gamma \vdash \mathsf{ext}(f, g, k) \star a = k \cdot a : \mathsf{Id}_B(f \cdot a, g \cdot a)}\ \Pi\text{-}\textsc{ext-app-def}$$

When we consider theories with any of these propositional rules, the corresponding COMP rule will always be included, even if not explicitly mentioned.

1.1.8 (Algebraic rules)**.** Beyond these specific constructors, we also consider extensions by arbitrary rules of a simpler form: *algebraic* axioms of each of the basic judgements.

An *algebraic type-forming axiom* is specified by a basic type-former T, unique to the axiom, together with a pre-context Γ_{T}; it then has introductory rule

$$\frac{\vdash \Gamma_{\mathsf{T}}\ \mathsf{cxt}}{\vec{x} : \Gamma_{\mathsf{T}} \vdash \mathsf{T}(\vec{x})\ \mathsf{type}}.$$

An *algebraic term-forming axiom* is similarly given by a basic term-former t, and pre-context Γ_{t} and pre-type A_{t}; its rule is then

$$\frac{\vec{x} : \Gamma_{\mathsf{T}} \vdash A_{\mathsf{t}}(\vec{x})\ \mathsf{type}}{\vec{x} : \Gamma_{\mathsf{T}} \vdash \mathsf{t}(\vec{x}) : A_{\mathsf{t}}(\vec{x})}.$$

And *algebraic type-* and *term-equality axioms*, similarly, are of the form:

$$\frac{\Gamma \vdash A,\ A' \ \mathsf{type}}{\vec{x} : \Gamma \vdash A = A' \ \mathsf{type}} \qquad\qquad \frac{\Gamma \vdash a, a' : A}{\Gamma \vdash a = a' : A}$$

for some given Γ, A, A' or Γ, A, a, a'.

1.1.9 (Logical frameworks)**.** An alternate presentation of the syntax is via a *logical framework*: rather than starting with raw syntax and then defining the judgements of well-formedness on top of that, we define well-formed types, terms etc. directly from the start; the logistical cost of this is using a dependently-typed metalanguage, the *logical framework*, assumed to have Π-types and their η-rule, plus one "universe" in which we encode our object language. [**Hof97**] gives a useful overview of the two presentations, together with a crucial comparison of their formal strengths, which we recall in Example 4.2.10; see also [**NPS90**] for further discussion.

We will briefly have cause to work in the logical framework presentation, and in particular to discuss Garner's rule Π-Id-ELIM (and its associated computation rule Π-Id-COMP), from [**Gar09b**], which involves second-order quantification in its premises and so cannot be directly expressed in the earlier presentation. This rule is a strong extensionality principle, asserting that the type $u, v : \Pi_{x:A} B(x) \ \vdash \ \Pi_{x:A} \ \mathsf{Id}_{B(x)}(u \cdot x, v \cdot x)$ is generated by canonical elements of the form $\lambda x.r(b(x))$:

$$\frac{\begin{array}{c}\Gamma,\ u, v : \Pi_{x:A} B(x),\ w : \Pi_{x:A} \ \mathsf{Id}_{B(x)}(u \cdot x, v \cdot x) \ \vdash \ C(u, v, w) \ \mathsf{type} \\ \Gamma,\ f : (x\!:\!A)B(x) \ \vdash \ d(f) : C(\lambda f, \lambda f, \lambda(r \circ f)) \\ \Gamma \ \vdash \ k, k' : \Pi_{x:A} B(x) \qquad \Gamma \ \vdash \ l : \Pi_{x:A} \ \mathsf{Id}_{B(x)}(k \cdot x, k' \cdot x)\end{array}}{\Gamma \ \vdash \ \mathsf{L}(C, d, k, k', l) : C(k, k', l)} \ \Pi\text{-}\mathsf{Id}\text{-}\textsc{elim}$$

Its computation rule concludes from appropriate premises that

$$\mathsf{L}(C, d, \lambda h, \lambda h, \lambda(r \circ h)) = d(h) : C(\lambda h, \lambda h, \lambda(r \circ h)). \quad \Pi\text{-}\mathsf{Id}\text{-}\textsc{elim-comp}$$

(Here and in the sequel, $(x\!:\!A)B(x)$ and $[x\!:\!A]b(x)$ respectively denote type and term abstraction in the metalanguage.)

[**Gar09b**, 5.11] shows that (over the core Π-type rules) the rule Π-Id-ELIM is inter-derivable with the conjunction of the first-order rules Π-EXT and Π-EXT-APP.

1.2. Categorical representation

1.2.1. The syntax as given so far is an excellent tool for intuitive and computationally tractable presentations of theories; however, for semantic/categorical purposes its formal complexity can be a hindrance. To this end, various categorical structures have been introduced which correspond, more or less closely, to dependent type theories as presented above[3].

Of these, the most categorically flexible are probably Jacobs' *comprehension categories*, which admit many useful variations, and connect well with other categorical structures; since in the present work we will stick closely to the type theory, however, we will use Cartmell's *categories with attributes* as our main model, as these are very elementarily presented.

Definition 1.2.2. A *category with attributes* consists of:

- a category $\mathcal{C}$, with a distinguished terminal object $\diamond$;
- a functor $\mathrm{Ty}\colon \mathcal{C}^{\mathrm{op}} \longrightarrow \mathbf{Sets}$; that is, for each object Γ, a set $\mathrm{Ty}(\Gamma)$, and for $f\colon \Delta \longrightarrow \Gamma$, actions $f^*\colon \mathrm{Ty}(\Gamma) \longrightarrow \mathrm{Ty}(\Delta)$, functorial in Γ;
- for each $A \in \mathrm{Ty}(\Gamma)$, an object $\Gamma.A$ and map $\pi_{\Gamma;A}\colon \Gamma.A \longrightarrow \Gamma$;
- for each $A \in \mathrm{Ty}(\Gamma)$ and $f\colon \Delta \longrightarrow \Gamma$, a pullback square

$$\begin{tikzcd} \Delta.f^*A \arrow[r, "q(f,A)"] \arrow[d, "\pi_{\Delta;f^*A}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & \Gamma.A \arrow[d, "\pi_{\Gamma;A}"] \\ \Delta \arrow[r, "f"] & \Gamma \end{tikzcd}$$

  again functorial in f, in that $q(1_\Gamma, A) = 1_{\Gamma.A}$, $q(f \circ g, A) = q(f, A) \circ q(g, f^*A)$.

[3]Most relevantly for our purposes, *comprehension categories* ([**Jac93**]), *categories with attributes* ([**Car78**], [**Mog91**], [**Pit00**]), *categories with families* ([**Dyb96**], [**Hof97**]), *contextual categories* ([**Car86**], [**Str91**]); see [**Jac93**], [**Hof97**] for useful overviews.

1.2.3. We will use a few pieces of obvious terminology for working in CwA's. Objects and maps of $\mathcal{C}$ we call *contexts*, and *context maps*. Elements of $\mathrm{Ty}(\Gamma)$ we call *types over* Γ.

For a type $A \in \mathrm{Ty}(\Gamma)$, the map $\pi_{\Gamma;A} \colon \Gamma.A \longrightarrow \Gamma$ is called a *basic dependent projection*. Compositions of such maps are called a *dependent projections*, and are denoted in diagrams as $\Gamma' \longrightarrow\!\!\!\!\!\rightarrow \Gamma$. A *term of type A in context* Γ is a section $a : \Gamma \longrightarrow \Gamma.A$ of the dependent projection $\pi_{\Gamma;A}$; we write $\mathrm{Tm}_\Gamma(A)$ for the set of these.

For an object $\Gamma \in \mathcal{C}$, a *dependent context* over Γ is a sequence $A_1 \in \mathrm{Ty}(\Gamma)$, $A_2 \in \mathrm{Ty}(\Gamma.A_1)$, $\ldots A_l \in \mathrm{Ty}(\Gamma.A_1.\ldots.A_{l-1})$, for some $l \geq 0$; we write $\mathrm{Cxt}(\Gamma)$ for the set of these. For any $\Delta \in \mathrm{Cxt}(\Gamma)$, there is an evident context extension $\Gamma.\Delta$, and dependent projection $\pi_{\Gamma;\Delta} \colon \Gamma.\Delta \longrightarrow\!\!\!\!\!\rightarrow \Gamma$.

Definition 1.2.4. A category with attributes $\mathcal{C}$ is *contextual* if the natural map $\mathrm{Cxt}(\diamond) \longrightarrow \mathrm{ob}\,\mathcal{C}$ is a bijection, and *accessible* [**Pit00**] when this map is a surjection.

Equivalently, $\mathcal{C}$ is contextual if there is a *length* function $l \colon \mathrm{ob}\,\mathcal{C} \longrightarrow \mathbb{N}$, such that

- $\diamond$ is the unique context of length 0;
- for any context Γ and type A, $l(\Gamma.A) = l(\Gamma) + 1$; and
- for any context Δ of length $n > 0$, there are unique Γ, A such that $\Delta = \Gamma.A$.

Our contextual CwA's correspond precisely to the *contextual categories* of Cartmell [**Car86**] and Streicher [**Str91**], and hence, crucially, to dependent type theories without constructors:

Proposition 1.2.5 ([**Car78**],[**Car86**])**.** *The category of small CwA's is* equivalent *to the category of type theories presented syntactically by (a set of) purely algebraic axioms (i.e. dependent terms and types, and equality axioms between them) and interpretations between such theories.*

Specifically, there is an adjoint equivalence

$$\begin{tikzcd} \mathbf{SynThy} \arrow[r, shift left, "\mathcal{C}\ell"] \arrow[r, phantom, "\simeq"] & \mathbf{CwA}_{cxl} \arrow[l, shift left, "\mathsf{Lang}"] \end{tikzcd}$$

taking a syntactically presented type theory T *to its* classifying category $\mathcal{C}\ell(\mathsf{T})$*, and a CwA* $\mathcal{C}$ *to its* internal language $\mathsf{Lang}(\mathcal{C})$.

(Note that this really is an honest equivalence of categories, not just a 2-equivalence or anything similarly weak; $\mathbf{CwA}_{cxl}$ and **SynThy** do carry natural 2-category structures, but for the present we study them purely 1-categorically. This is our main reason for working specifically with contextual CwA's, rather than just e.g. accessible)

1.2.6. This equivalence justifies working with a *presentation-agnostic* category of type theories, which we denote **DTT**, defined up to equivalence as either **SynThy** or $\mathbf{CwA}_{cxl}$: we will construct and work with objects of **DTT** (*theories*) sometimes as syntactic presentations, sometimes as categories with attributes. By abuse of notation we will use $\mathcal{C}\ell$ when we wish to emphasise the forgetful functor $\mathbf{DTT} \longrightarrow \mathbf{Cat}$.

Given any construction either on syntactically presented theories or on contextual CwA's, we will transfer it without comment to **DTT**, and so forth. In subsequent chapters, we will work more often with syntactic presentations, but for the constructions of this section it will be convenient to work primarily via $\mathbf{CwA}_{cxl}$, for the sake of its connections to other categories of CwA's.

The major power of the $\mathbf{CwA}_{cxl}$ presentation, however, is that it displays **DTT** as the category of models of a (small) *essentially algebraic theory* [**Joh02**, D1.3.4(a)]; this implies many useful categorical properties, including in particular that it is complete, cocomplete, and moreover locally presentable—in fact, finitely presentable, since contextual CwA's are a finitary theory.

(It is clear from our definition that CwA's admit an essentially algebraic presentation; to give such a presentation of contextual CwA's, we must break down the "objects" sort into $\mathbb{N}$-many sorts, one for objects of each length l, and split up the other sorts and operations similarly.)

The construction of the equivalence also gives us a useful universal property for theories presented syntactically: a map out of such a theory is just an interpretation, and hence is determined precisely by the interpretations of the axioms.

1.2.7. Of course, we want categories not just of purely algebraic dependent type theories, but of type theories with constructors; in particular, Id- and Π-types. These too can be succinctly and profitably defined in terms of CwA's:

Definition 1.2.8. An *elim-structure* e on a map $i \colon \Gamma \longrightarrow \Theta$ is a function assigning, to every type $C \in \mathrm{Ty}(\Theta)$ and every term $d \colon \Gamma \longrightarrow \Gamma.i^*C$ of type i^*C (equivalently, every map $\hat{d} \colon \Gamma \longrightarrow \Theta.C$ over Θ) a term $e_{C,d}$ of type C such that $e_{C,d} \cdot i = \hat{d}$. Diagramatically, we indicate elim-structures as $i \colon \Gamma \rhd\!\!\longrightarrow \Theta$.

$$\begin{tikzcd} \Gamma \arrow[rr, "\hat{d}"] \arrow[dr, "i"'] & & \Theta.C \arrow[dl, bend left] \\ & \Theta \arrow[ur, dotted, "e_{C,d}"] & \end{tikzcd}$$

A *Frobenius elim-structure* on $i \colon \Gamma \longrightarrow \Theta$ is an elim-structure e_Δ on $i.\Delta$ for each dependent context Δ over Θ.

This axiomatises the structure provided by the elimination/computation rules for an inductive type with just a single introduction form i. (It can be nicely generalised to deal with multiple introduction forms, but we will not need that.) In particular:

Definition 1.2.9. A *CwA with* Id-*types* is a CwA $\mathcal{C}$, together with:

- for each context $\Gamma \in \mathcal{C}$ and type $A \in \mathrm{Ty}(\Gamma)$, a type $\mathsf{Id}_A \in \mathrm{Ty}(\Gamma.A.A)$, and a morphism $r_A \colon \Gamma.A \longrightarrow \Gamma.A.A.\mathsf{Id}_A$ over $\Delta_A \colon \Gamma.A \longrightarrow \Gamma.A.A$ with a Frobenius elim-structure J_A,
- all stable in Γ, in that for $f \colon \Gamma' \longrightarrow \Gamma$ and $A \in \mathrm{Ty}(\Gamma)$, we have $(f.A.A)^*\mathsf{Id}_A = \mathsf{Id}_{f^*A} \in \mathrm{Ty}(\Gamma'.f^*A.f^*A)$, and so on.

(Note that $\Gamma.A.A$ and so on are a slight abuse of notation: formally we should write $\Gamma.A.\pi^*_{\Gamma,A}A$ and the like.)

Write $\mathbf{CwA}^{\mathsf{Id}}$, $\mathbf{CwA}^{\mathsf{Id}}_{cxl}$, etc. for the various categories of CwA's with Id-types.

Proposition 1.2.10 ([**Hof97**], [**Pit00**]). *This structure really does correspond precisely to the Id-type rules: the equivalence of Proposition 1.2.5 lifts to an equivalence between $\mathbf{CwA}^{\mathsf{Id}}_{cxl}$ and a category $\mathbf{SynThy}^{\mathsf{Id}}$, of theories presented syntactically by the core Id-type rules plus algebraic axioms.*

1.2.11. Similarly, we may define structure on the categorical side corresponding to the core Π-type rules, or any of the other rules of 1.1.6, 1.1.7 above (with obvious dependencies: the structure for EXT rules refers to the Π- and Id-types structure, and so on).

In particular, if Φ is any appropriate collection of the above rules, then we can obtain a category $\mathbf{CwA}^{\Phi}_{cxl}$ and an equivalence $\mathbf{CwA}^{\Phi}_{cxl} \simeq \mathbf{SynThy}^{\Phi}$ as before, and justified by this we introduce a presentation-agnostic category $\mathbf{DTT}_{\Phi}$ of theories over Φ.

In the case of all the rules above, theories over Φ are again models of an essentially algebraic theory extending the base theory of contextual CwA's. This motivates the level of generality we work at: when we speak informally of a *collection of rules/constructors* Φ, what we mean formally is an algebraic theory extending the theory of contextual CwA's, with the same sorts.

(This is not a terribly natural definition. It allows type-theoretically unnatural constructions, e.g. rules which hold only over contexts of length 17, and the like; and, worse, important constructions such as the logical framework embedding cannot defined in this generality. However, as a crude demarcation, it will suffice for the scope of the present work.)

This immediately gives us comparison functors between different categories of theories. If Φ' is any collection of rules/constructors extending Φ, then we have a map of algebraic theories, and hence an adjunction:

$$\begin{tikzcd} \mathbf{DTT}_\Phi \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{DTT}_{\Phi'} \arrow[l, bend left, "U"] \end{tikzcd}$$

where, in syntactic terms, F adds to a theory the extra rules/constructors of Φ, while U forgets this structure. In particular, if T is a theory over Φ presented by some collection of axioms, then $F(\mathsf{T})$ is the theory presented by the same axioms over Φ'.

This is the main reason why we consider second-order rules such as Π-Id-ELIM separately. They do *not* (at least a priori) correspond to extra essentially algebraic structure over contextual CwA's; in the present categorical setup, they can be discussed only within their logical framework embeddings.

1.2.12 (Left and right maps in a CwA)**.** The above presentation of Id-types via elim-structures is based on ideas of Gambino and Garner ([**GG08**]) which are implicit in much of the present work: that any CwA has important classes of left and right maps, which in the presence of Id-types is moreover a weak factorisation system. (These notions are recalled and discussed in Section 2.1 below).

The right maps are just retracts of (compositions of basic) dependent projections. The left maps are maps admitting an (or algebraically: with a chosen) elim-structure.

Note that categorically, an elim-structure on $i \colon \Gamma \longrightarrow \Theta$ just provides fillers for all triangles from i to a basic dependent projection; or equivalently, fillers for all squares from i to basic dependent projections, commuting with the canonical pullback squares between dependent projections. In other words, an elim-structure is exactly a $\mathcal{J}^{\boxslash}$-structure as defined in 2.1.3, where $\mathcal{J}^{\boxslash}$ is the category of basic dependent projections and canonical pullback squares between them.

$$
\begin{tikzcd}
\Gamma \arrow[rr, "d"] \arrow[dr, "i"'] & & \Theta.C \arrow[dl, bend left] \\
& \Theta \arrow[ur, dotted, "e_{C,d}"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
\Gamma \arrow[r, "d"] \arrow[d, "i"'] & \Xi.C \arrow[d] \\
\Theta \arrow[r, "f"'] \arrow[ur, dotted, "e_{f^*C,f^*d}"] & \Xi
\end{tikzcd}
$$

([**GG08**] moreover gives an alternative type-theoretic characterisation of each class of maps, and uses Id-types to construct $\mathcal{L}, \mathcal{R}$ factorisations, but we will not need these.)

1.3. Constructions on dependent type theories

There are several interesting and important ways to construct new type theories from old:

1.3.1. First, the *dependent contexts monad* on **CwA** over **Cat**, sending $\mathcal{C} = (\mathcal{C}, \mathrm{Ty})$ to $\mathcal{C}^{\mathrm{Cxt}} := (\mathcal{C}, \mathrm{Cxt})$ as defined in 1.2.3. So the base category is unchanged; but types of the new theory are dependent contexts of the old, and context extension is just by concatenation.[4]

Moreover, given Id-types structure on $\mathcal{C}$, we can extend this to an Id-types structure on $\mathcal{C}^{\mathrm{Cxt}}$: this is the "identity contexts" of [**Str93**] or [**GG08**]. So we have an endofunctor $(-)^{\mathrm{Cxt}}$ on $\mathbf{CwA}^{\mathsf{Id}}$. However, the monad structure does *not* lift to $\mathbf{CwA}^{\mathsf{Id}}$: the unit is fine, but the multiplication turns out not to preserve the Id-structure strictly. (One might hope that this could be accommodated by a theory of pseudo-maps of $\mathbf{CwA}^{\mathsf{Id}}$'s.)

Similar techniques let us lift Π-types from $\mathcal{C}$ to $\mathcal{C}^{\mathrm{Cxt}}$, and hence to lift $(-)^{\mathrm{Cxt}}$ to an endofunctor of $\mathbf{CwA}_{\Pi}$; likewise for Σ-types, the Π-η rules, and most other standard type-constructors.

---

[4] This can also be seen as a monad for "very strong, strictly associative Σ-types".

Matters are slightly subtler for the functional extensionality rules. Π-EXT alone does not seem to lift from $\mathcal{C}$ to $\mathcal{C}^{\mathrm{Cxt}}$; however, Π-EXT combined with either Π-EXT-APP or Π-EXT-APP-DEF does lift.

So for various sets of rules Φ, we have an endofunctor $(-)^{\mathrm{Cxt}}$ on $\mathbf{CwA}^{\Phi}$. However, this does not restrict to one on $\mathbf{CwA}_{cxl}$: its result is almost never contextual, since adjoining two types A and B to a context in succession has the same result as adjoining (A, B) in one step, so (as long as the original theory had any types at all) there can be no well-defined notion of length.

1.3.2. The *slice* construction is one of the fundamental tools of the category–type theory correspondence; however, in terms of CwA's, it is not exactly the ordinary categorical slice.

For $\mathcal{C}$ any CwA, and Γ any object of $\mathcal{C}$, the *(type-theoretic) slice* $\mathcal{C}/\!/\Gamma$ has as objects dependent contexts over Γ, and morphisms and attributes structure induced by pullback along the map $\mathrm{ob}(\mathcal{C}/\!/\Gamma) \longrightarrow \mathrm{ob}\,\mathcal{C}$ sending $\Delta \in \mathrm{Cxt}_{\mathcal{C}}(\Gamma)$ to Γ, Δ.

In syntactic terms, slicing corresponds to taking variables into the context: judgements $\Delta \vdash \mathcal{J}$ in $\mathsf{T}/\!/\Gamma$ correspond exactly to judgements $\Gamma, \Delta \vdash \mathcal{J}$ in T.

Slices are always contextual. In particular, by slicing any $\mathcal{C} \in \mathbf{CwA}$ over its "empty context" $\diamond$, we obtain the *contextual core* of $\mathcal{C}$: this gives a right adjoint $-/\!/\diamond \colon \mathbf{CwA} \longrightarrow \mathbf{CwA}_{cxl}$ to the inclusion $\mathbf{CwA}_{cxl} \rightarrowtail \mathbf{CwA}$.

The slice construction also extends to *all* rules and constructors we have considered.

(There is also a natural CwA structure on the *categorical slice* $\mathcal{C}/\Gamma$, but this will not be of importance to us; it is rarely contextual or even accessible, and its contextual core is just $\mathcal{C}/\!/\Gamma$.)

1.3.3. Combining the type-theoretic slice with the dependent contexts construction gives us an endofunctor $(-)^{\mathrm{Cxt}}/\!/\diamond$ of $\mathbf{CwA}_{cxl}$, the "types to contexts" construction:

$$\mathbf{CwA}_{cxl} \rightarrowtail \mathbf{CwA} \xrightarrow{(-)^{\mathrm{Cxt}}} \mathbf{CwA} \xrightarrow{-/\!/\diamond} \mathbf{CwA}_{cxl}$$

This composite endofunctor is essentially just a "contextualised" version of $(-)^{\mathrm{Cxt}}$: the base category of contexts is left unchanged, the types of the new theory are the dependent contexts of the old, and context extension is by concatenation.

Like $(-)^{\mathrm{Cxt}}$, this endofunctor lifts to $\mathbf{DTT}^{\Phi}$ for various important sets of rules Φ; again like $(-)^{\mathrm{Cxt}}$, it is a monad in the constructor-free case (since it is just a monad wrapped in an adjunction), but fails to be one in the presence of Id-types.

This construction will be briefly but crucially useful to us, in 3.4.3 and 4.1.7 below.

1.3.4. With some of our constructors, we can also construct *co-slice theories.* In a co-slice $\Theta\backslash\mathcal{C}$, an object is a map $g\colon \Theta \longrightarrow \Gamma$ of $\mathcal{C}$ (a Θ-*pointed object of* $\mathcal{C}$); a map $(g, \Gamma) \longrightarrow (d, \Delta)$ is a map $f\colon \Gamma \longrightarrow \Delta$ of $\mathcal{C}$ with $fg = d$ (i.e. preserving the "point"); and a type over (g, Γ) is a type $A \in \mathrm{Ty}(\Gamma)$ together with a term a of g^*A, or equivalently a point $(g, a)\colon \Theta \longrightarrow \Gamma, A$ for which $\pi_{\Gamma;A}(g, a) = g$.

There is an obvious functor $\Theta\backslash\mathcal{C} \longrightarrow \mathcal{C}$, forgetting the points.

This construction preserves contextuality; it also extends to act on theories with Id-types, Σ-types, $\mathsf{1}$, and more generally inductive types with a single unary constructor; but it does *not* extend to Π-types, nor to Bool, $\mathsf{0}$, or most other type-formers. This is familiar categorically: co-slices retain e.g. binary products and terminal objects, but not coproducts or exponentials.

1.4. Miscellaneous facts

1.4.1 (Normalisation results)**.** It is a fundamental fact, going back to [**ML75**], that the basic structural theory together with any subset of the standard constructors (Id-, Σ-, Π-types, and also Nat and Bool) is strongly normalising.

Moreover, it is clear from most proofs that this result extends to theories including algebraic type- and term-forming axioms. (It can fail, however, under the addition of algebraic definitional equality axioms.)

Normalisation is especially informative for theories also possessing the *canonicity* property, that all normal forms are "canonical", i.e. are terms formed from the INTRO-rule constructors. Unfortunately, this will fail in many theories of interest to us—in particular, to theories with terms of identity types adjoined. We will discuss canonicity of particular theories as and when we require it.

It is currently somewhat unclear to what extent one can retain strong normalisation in conjunction with the functional extensionality rules; in any account, the forms considered here certainly break canonicity.

The Observational Type Theory of Altenkirch and collaborators ([**AM06**] , [**AMS07**]) reconciles these, but has *defined* rather than axiomatic identity types (and therefore does automatically permit extension by further type axioms), and moreover forces these to be trivial: any two terms of an identity type are equal (the axiom of *Uniqueness of Identity Proofs*). However, the OTT system is an encouraging step towards the development of a fully intensional system with both functional extensionality and normalisation/canonicity. (The difficulty lies essentially in defining the computational behaviour of the extensionality combinator; this seems to be related to the admissibility of the principle $\overline{J}$ of Section 4.2, and is of course intimately connected with the investigations of [**Gar09b**].)

We will also need in Chapter 3 a small proposition on limits in syntactic categories:

Proposition 1.4.2. *Suppose* $\Gamma = \bigwedge_{i \in I} x_i : A_i$ *is a context in* $\mathbf{T}$*, and* $\mathcal{F} \subseteq \mathcal{P}(I)$ *a set of subsets of* I*, closed under binary intersection and with* $\bigcup \mathcal{F} = I$*, such that for each* $J \in \mathcal{F}$*,* $\Gamma_J = \bigwedge_{i \in J} x_i : A_i$ *is also a well-formed context.*

(Here $\bigwedge_{i \in P} x_i : A_i$ *denotes the context* $x_{i_1} : A_{i_1}, \ldots\ x_{i_k} : A_{i_k}$*, for any finite linear order* $P = \{i_1 < \ldots < i_k\}$*.)*

Then the contexts Γ_J *and the dependent projections between them give a diagram*

$$\Gamma_{-} \colon (\mathcal{F}, \subseteq)^{\mathrm{op}} \longrightarrow \mathcal{C}\ell(\mathbf{T}),$$

and the dependent projections $\Gamma \longrightarrow \Gamma_J$ *express* Γ *as its limit:*

$$\Gamma = \varprojlim_{J \in \mathcal{F}} \Gamma_J.$$

Moreover, for a translation $F \colon \mathbf{T} \longrightarrow \mathbf{T}'$*, the functor* $\mathcal{C}\ell(F)$ *preserves such limits.* $\square$

A familiar special case asserts that if $\Gamma \vdash A$ type and $\Gamma \vdash B$ type, then the following square of projections is a pullback:

$$\begin{tikzcd}
\Gamma,\, x{:}A,\, y{:}B \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \Gamma, y{:}B \arrow[d] \\
\Gamma,\, x{:}A \arrow[r] & \Gamma
\end{tikzcd}$$

The proof of the general proposition is essentially the same.

CHAPTER 2

Higher-categorical background

2.1. Weak factorisation systems

The study of *weak factorisation systems* originated in homotopy theory, and has since become important in higher category theory, especially with recent developments in *algebraic*[1] *weak factorisation systems.* We will not require any of the more sophisticated results of this theory; but there are a couple of basic constructions from it which we will meet repeatedly, so which we now recall here. See [**Hov99**, Ch. 1] for further background on the classical theory, and [**Gar09d**] for the algebraic case.

Definition 2.1.1. For maps f, g in a category $\mathcal{C}$, we say f is (weakly)[2] *(left) orthogonal to* g, written $f \boxslash g$, if every commutative square from f to g has a filler:

$$\begin{tikzcd} D \arrow[r] \arrow[d, "f"'] & Y \arrow[d, "g"] \\ C \arrow[r] \arrow[ur, dotted, "\exists"] & X \end{tikzcd}$$

There are several synonyms: "f has the (weak) *left lifting property* against g", and corresponding right-handed versions phrasing it as a property of of g.

If $\mathcal{L}$, $\mathcal{R}$ are classes of maps in $\mathcal{C}$, we say $\mathcal{L} \boxslash \mathcal{R}$ if $f \boxslash g$ for every $f \in \mathcal{L}$, $g \in \mathcal{R}$.

Simple examples of such classes are given in **Sets** by $\mathcal{L}$ = surjections, $\mathcal{R}$ = injections; or $\mathcal{L}$ = injections, $\mathcal{R}$ = surjections. (In a constructive

[1] often (and originally) called *natural*

[2] We will never require the strong counterparts of these properties, so will assume weak by default.

foundation, the latter system has to be restricted to (*decidable* injections; *split* surjections).)

2.1.2. A more powerful example, in **Top**, takes $\mathcal{L}$ to be retracts of relative cell complexes, and $\mathcal{R}$ to be trivial fibrations in the Quillen model structure, sometimes called weakly contractible maps.

The definition of these is most clearly given as an example of a very general construction: if $\mathcal{J}$ is any class of maps, we define $\mathcal{J}^{\boxslash}$ to be the class of maps weakly right orthogonal to all maps in $\mathcal{J}$; and dually we define ${}^{\boxslash}\mathcal{J}$.

It is easy to check that a class $\mathcal{J}^{\boxslash}$ must contain all identities, and be closed under composition, retracts, and pullback along arbitrary maps. Dually, ${}^{\boxslash}\mathcal{J}$ is closed under identities, composition, retracts, and pushouts. A little more thought shows that the classes are closed under (co-)transfinite composition (i.e. under colimits of colimit-preserving diagrams indexed by ordinals, or under dually defined limits).

Most typically we start with a *set* of maps $\mathcal{J}$ in a category $\mathcal{C}$, then form $\mathcal{R} = \mathcal{J}^{\boxslash}$ ("$\mathcal{J}$-fibrations"), $\mathcal{L} = {}^{\boxslash}(\mathcal{J}^{\boxslash})$ ("$\mathcal{J}$-cofibrations").

For the case of Serre trivial fibrations, we start with the inclusions of spheres into discs (*cells*), as their boundaries:

$$\mathcal{J} = \{\, S^{i-1} \rightarrowtail D^i \mid i \in \mathbb{N} \,\}.$$

Then a map $p \colon Y \longrightarrow X$ is in $\mathcal{J}^{\boxslash}$ just if whenever we are given a cell in X and a lift of its boundary to Y, we can lift the whole cell to Y, with the given boundary. So any loop in Y that is null-homotopic in X must be null-homotopic in Y, and so on: p is forced in particular to be a weak homotopy equivalence.

But now by the closure properties of ${}^{\boxslash}(\mathcal{J}^{\boxslash})$, the generating boundary inclusions are not the only maps we can lift against trivial fibrations. We can lift any *relative cell complex*, i.e. any transfinite composite of pushouts of the boundary inclusions, essentially by lifting it one cell at a time; and indeed we can also lift retracts of these.

This exemplifies one of the two main flavours of classes of orthogonal maps: the left maps are *cofibrations* of some sort, the right maps are *trivial fibrations*. The other flavour is dual: *trivial cofibrations* vs. *fibrations*.

Things work similarly from any basic set $\mathcal{J}$ in a co-complete category $\mathcal{C}$; *$\mathcal{J}$-cell complexes* (transfinite compositions of pushouts of maps in $\mathcal{J}$) will always be $\mathcal{J}$-cofibrations. Under good circumstances (co-completeness and smallness conditions), all $\mathcal{J}$-cofibrations will be retracts of these, and every map of $\mathcal{C}$ will have a factorisation into a J-cell complex followed by a $\mathcal{J}$-fibration, giving a *weak factorisation system*; but this is beyond what we will require.

2.1.3. The above theory can be improved and refined by changing the objects of study from mere *properties* of maps to *structures* on maps. This both strengthens the resulting theory—constructions become more natural, functorial, etc.—and more naturally describes mathematical practice: demonstrating "fibrationhood" of a map typically involves giving some kind of structure on it. This is also a more naturally constructive approach—an existence property should always be given by exhibiting witnesses—and, as such, turns out to develop much better in a constructive setting than the classical theory does.

We recall only the barest rudiments of this theory. Given a category $\mathcal{J}$ with a functor $U \colon \mathcal{J} \longrightarrow \mathrm{Arr}(\mathcal{C})$, a $\mathcal{J}^{\boxslash}$-structure on a map p is a function *choosing*, for each $j \in \mathcal{J}$ and commutative square from $U(j)$ to p, a diagonal filler, such that these fillers commute with squares coming from the maps of $\mathcal{J}$.

$$\begin{tikzcd}
\bullet \arrow[r] \arrow[d, "Uj'"'] & \bullet \arrow[r] \arrow[d, "Uj"] & X \arrow[d, "p"] \\
\bullet \arrow[r] \arrow[urr, dotted] & \bullet \arrow[r] \arrow[ur, dotted] & Y
\end{tikzcd}$$

If $\mathcal{J}$ is just a discrete set, then assuming choice, the maps admitting a $\mathcal{J}^{\boxslash}$-structure are exactly the fibrations (in the sense of 2.1.2) with respect to the image of U. $\mathcal{J}^{\boxslash}$-structured maps, together with commutative squares between them preserving the structure (in a suitable sense), form a category themselves, again with a forgetful functor to $\mathrm{Arr}(\mathcal{C})$.

Structured versions of the closure properties above can be formulated: any identity carries a natural $\mathcal{J}^{\boxslash}$-structure, as does any (finite or transfinite) composite of maps with $\mathcal{J}^{\boxslash}$-structures; any pullback of a map with $\mathcal{J}^{\boxslash}$-structure again carries a natural such structure, as does any retract of such a map. When we speak of some category of structured maps being *closed under composition*, and similar, we will mean statements of this form.

Again analogously to the classical theory, any map exhibited as (a retract of) a *formal $\mathcal{J}$-cell complex* lifts against any map with $\mathcal{J}^{\boxslash}$-structure; these liftings are now moreover *natural* in the appropriate structure preserving maps. Again, under good circumstances, we have factorisations of every map into (the realisation of) a formal $\mathcal{J}$-cell complex, followed by (the underlying map of) a $\mathcal{J}^{\boxslash}$-structured map, and this is now moreover functorial [**Gar09d**].

The constructions of $\mathcal{J}^{\boxslash}$ and $\mathcal{J}$-cell complexes are the ones which will recur in the present work: for elim-structures on context maps (1.2.8), for contractions on maps of globular sets and operads (2.3.6), and for contractions on maps of theories (4.2.5).

2.2. Strict higher categories

2.2.1. There are various approaches to higher category theory, and in particular many definitions of weak higher categories [**Lei02**], [**CL04**]. We will use the *globular operadic* approach, as set out originally by Michael Batanin in [**Bat98**], and later re-presented and slightly modified by Tom Leinster in [**Lei04**]. These two sources are the authoritative references for most of this material; [**Lei02**] also provides an excellent, rather more streamlined, introduction.

In the globular approach, the underlying substance of an ω-category $\mathcal{C}$ consists of a set C_n of "n-cells" for each $n > 0$. The 0- and 1-cells correspond

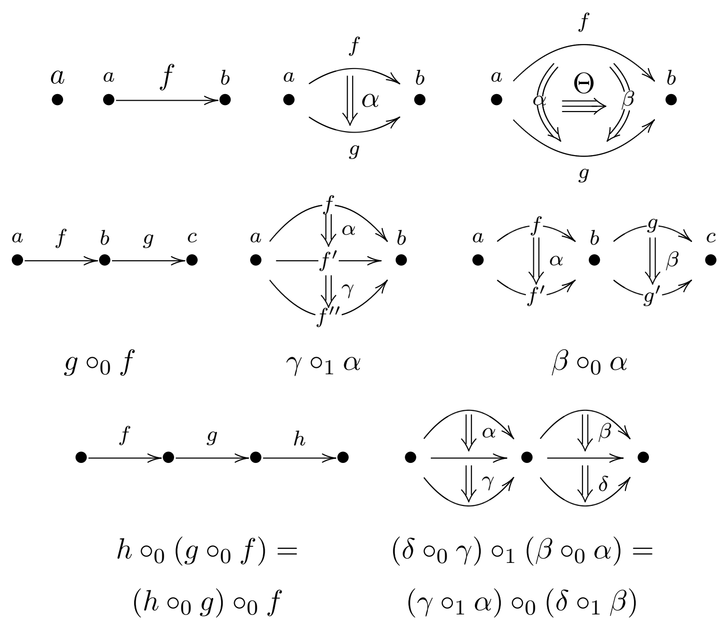


FIGURE 2.2.A. Some cells, composites, and associativities in a strict higher category

to the objects and arrows of an ordinary category: each arrow f has source and target objects $a = s(f)$, $b = t(f)$. Similarly, the source and target of a 2-cell α are a parallel pair of 1-cells $f, g : a \rightrightarrows b$, and generally the source and target of an $(n+1)$-cell are a parallel pair of n-cells. Summing this up we arrive at:

Definition 2.2.2. A *globular set* $A_\bullet$ is a diagram of sets and functions

$$A_0 \underset{t_0}{\overset{s_0}{\leftleftarrows}} A_1 \underset{t_1}{\overset{s_1}{\leftleftarrows}} A_2 \underset{t_2}{\overset{s_2}{\leftleftarrows}} A_3 \leftleftarrows \ \ldots$$

such that $s_i \cdot s_{i+1} = s_i \cdot t_{i+1}$ and $t_i \cdot s_{i+1} = t_i \cdot t_{i+1}$—the *globularity* conditions, asserting that the source and target of any cell are parallel.

Some notation we will use throughout the sequel: we will often drop the subscripts on arrows as far as possible, writing just s and t; s^i, t^i will denote i-fold iterations of these operations, as usual; and for $c \in A_n$ and $k \leq n$, we take $s_k(c) := s^{n-k}(c)$, $t_k(c) := t^{n-k}(c)$, the k-dimensional source and target of c. (Of course when $n = k+1$, this agrees with the original usage of s_k, t_k.)

A map of globular sets $f_\bullet \colon A_\bullet \longrightarrow B_\bullet$ consists of a sequence of functions $f_n \colon A_n \longrightarrow B_n$, preserving the globular structure, in that $s_i \cdot f_{i+1} = f_i \cdot s_i$ and $t_i \cdot f_{i+1} = f_i \cdot t_i$, or more compactly, $sf = fs$, $tf = ft$.

2.2.3. More generally, let $\mathbb{G}$ be the category with objects $\mathbb{N}$ and arrows generated by

$$0 \underset{t_0}{\overset{s_0}{\rightrightarrows}} 1 \underset{t_1}{\overset{s_1}{\rightrightarrows}} 2 \underset{t_2}{\overset{s_2}{\rightrightarrows}} 3 \rightrightarrows \ldots$$

subject to the equations $ss = ts$, $st = tt$. Then a *globular object* in a category $\mathcal{E}$ is a functor $A_\bullet \colon \mathbb{G}^{\mathrm{op}} \longrightarrow \mathcal{E}$, and a map of these is a natural transformation between such functors.

Now the category of globular sets is just the category $\widehat{\mathbb{G}}$ of presheaves on $\mathbb{G}$. The Yoneda embedding $\mathbf{y} \colon \mathbb{G} \longrightarrow \widehat{\mathbb{G}}$ yields some specific globular sets which will be of particular use and importance to us: the *basic n-cells* or *n-globes* $\mathbf{y}(n)$, and their boundaries $\partial(n) \subseteq \mathbf{y}(n)$, the maximal proper subobject of $\mathbf{y}(n)$, consisting of all *non-identity* maps into n. Note that for $n > 0$, we have $\partial(n) \cong \mathbf{y}(n-1) +_{\partial(n-1)} \mathbf{y}(n-1)$, a *parallel pair* of $(n-1)$-cells.

(For the finite-dimensional versions of all the above, and all that follows, $\mathbb{G}$ is replaced by the category $\mathbb{G}_n$, defined just as $\mathbb{G}$ except with no objects or arrows above n; an n-globular set is a presheaf $A_\bullet \colon \mathbb{G}_n^{\mathrm{op}} \longrightarrow \mathbf{Sets}$, and so on.)

To complete the definition of strict ω-categories, we simply add the structure of composition on top of this basic data. As illustrated in Figure 2.2.A, we want to compose cells whenever the target of one is the source of another in any lower dimension. Specifically, for any $k < n$, the set of k-composable

n-cells is the pullback

$$\begin{tikzcd} A_n \times_k A_n \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & A_n \arrow[d, "t_k"] \\ A_n \arrow[r, "s_k"] & A_k \end{tikzcd} .$$

Definition 2.2.4. [3] A *strict ω-category* $\mathcal{C}$ is a globular set $C_\bullet$ together with composition operations for each $k < n$

$$\circ_k \colon C_n \times_k C_n \longrightarrow C_n$$

and unit maps

$$r_n \colon C_{n-1} \longrightarrow C_n$$

(for which we use index conventions analogous to those for s, t), such that for each $j < k < l$, $(C_j, C_k, C_l, \circ_j, r_k, \circ_k, r_l)$ forms a strict 2-category. Unwinding a bit further, this says that:

- for each $k < n$, $(C_k, C_n, \circ_k, r_n)$ forms a category, i.e. the source and target of composites and units are "what one would expect", and the associativity and unit laws

$$c \circ_k (b \circ_k a) = (c \circ_k b) \circ_k a$$

$$a \circ_k (r_n s_k a) = a = (r_n t_k a) \circ_k a$$

hold (for appropriately composable $a, b, c \in C_n$); and additionally,

- for each $j < k < n$, and suitable $a, b, c, d \in C_n$, we have $s_k(b \circ_j a) = s_k(b) \circ_j s_k(a)$, and similarly for t_k; and the *interchange law* holds:

$$(d \circ_j c) \circ_k (b \circ_j a) = (d \circ_k b) \circ_j (c \circ_k a)$$

(also as illustrated in Figure 2.2.A).

---

[3]Strict ω-categories may alternatively, and completely equivalently, be defined via iterated enrichment.

2.2.5. This presentation exhibits strict ω-categories explicitly as models of an essentially algebraic theory, monadic over $\widehat{\mathbb{G}}$:

$$\begin{tikzcd} \widehat{\mathbb{G}} \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathbf{str}\text{-}\omega\text{-}\mathbf{Cat} \arrow[l, bend left, "U"] \end{tikzcd}$$

and it is shown in [**Str00**], [**Lei04**] that the resulting monad $T = FU$ in $\widehat{\mathbb{G}}$ is familially representable (that is, naturally isomorphic to a coproduct of representable functors), and indeed cartesian—both its functor part, and its monad structure η, μ.

(Recall that a functor is cartesian if it preserves pullbacks, and a natural transformation is cartesian if all its naturality squares are pullbacks.)

The cells of TX may thus be seen as *formal composites*, or *labelled pasting diagrams*, of cells of X. In particular, taking 1 to be the terminal globular set, with a single cell of each dimension, we define:

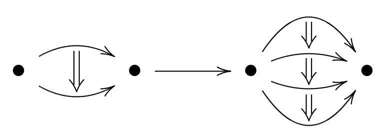
Definition 2.2.6. A *pasting diagram* is a cell of $T1$, the free strict ω-category on the terminal globular set. We will often write **pd** for $T1$. The source and target operations in **pd** coincide; we will often denote their common value by d.

There are several useful combinatorial representations for pasting diagrams. [**Lei04**, 8.1] provides an inductive description in terms of free monoids:

$$\mathbf{pd}_0 := T_{\mathbf{Mon}}(\emptyset) = \{\, ()\,\}$$

$$\mathbf{pd}_{n+1} := T_{\mathbf{Mon}}(\mathbf{pd}_m) = \{\, (\pi_1, \ldots, \pi_l) \mid l \geq 0,\ \pi_i \in \mathbf{pd}_n \,\}$$

Here, for instance, the element $(((),(),()),(),(()))\in \mathbf{pd}_2$ represents the 2-cell $(c_2 \circ_1 c_2 \circ_1 c_2) \circ_0 (r_2 c_1) \circ_0 (c_2)$ of $T1$ (where c_i is the unique i-cell of 1):



This representation may be seen syntactically as a normal form theorem: every formal expression in the operations r_n, $\circ_k$ may be put into normal form by eliminating identities wherever possible and moving lowest-dimensional composition outermost.

The k-dimensional source or target of a pasting diagram is given (in this representation) by simply discarding everything nested more than k levels deep.

2.2.7. With this representation, we can construct for each n-dimensional pasting diagram π an n-dimensional globular set $\widehat{\pi}$ (justifying the ubiquitous illustration of pasting diagrams):

$$(\widehat{\pi_1, \ldots, \pi_l})_i \ := \ \begin{cases} \sum_{1 \leq j \leq l} (\widehat{\pi}_j)_{i-1} & i > 1 \\ \{0, \ldots, l\} & i = 0 \end{cases}$$

These form a functor $(\int_{\mathbb{G}^{\mathrm{op}}} \mathbf{pd})^{\mathrm{op}} \longrightarrow \widehat{\mathbb{G}}$; that is, there are co-source and co-target embeddings $s, t \colon \widehat{d\pi} \rightarrowtail \widehat{\pi}$, satisfying the usual equations.

We will often write $\widehat{s\pi}$, $\widehat{t\pi}$ to distinguish which of these embeddings is implicitly assumed, as for instance in defining the *boundaries*
delπ of pasting diagrams: the unique 0-dimensional pasting diagram has empty boundary, and for $n > 0$, $\pi \in \mathbf{pd}_n$,

$$\partial\widehat{\pi} \ := \ \widehat{s\pi} +_{\partial\widehat{d\pi}} \widehat{t\pi}$$

There is an obvious natural map $\partial\widehat{\pi} \longrightarrow widehat\pi$; note that it may fail to be a monomorphism, if π is (wholly or partially) degenerate.

The realisations $\widehat{\pi}$ now provide the promised familial representation of T, decomposing TX into the fibers of the map $T! \colon TX \longrightarrow T1$:

$$TX_n = \sum_{\pi \in T1_n} \widehat{\mathbb{G}}(\widehat{\pi}, X).$$

Here and in the next few constructions, it is highly instructive to compare this with the analogous presentation of the "free monoid" monad on **Sets**:

$$T_{\mathbf{Mon}} X = \sum_{n \in \mathbb{N}} \mathbf{Sets}([n], X).$$

2.3. Globular operads

. . . à la Leinster. A weak higher category, as outlined in the introduction, should again consist of a globular set together with some composition operations, similar to those in a strict higher category; but now there may be multiple composition operations for each shape of pasting diagram. To present appropriate algebraic theories of such structures, we introduce the definition:

Definition 2.3.1 ("global version"). A *globular operad* P is a monad T_P on $\widehat{\mathbb{G}}$, together with a cartesian monad map $\alpha\colon T_P \longrightarrow T$. (This moreover implies that T_P is itself cartesian.)

Thus a globular operad yields again an algebraic theory over globular sets, via its monad part T_P; the natural transformation $\alpha\colon T_P A_\bullet \longrightarrow T A_\bullet$ ensures that all the operations of T_P can be viewed as composition operations for pasting diagrams, while the cartesianness of α ensures that the set of such operations for any pasting diagram is uniform in X. Precisely,

Definition 2.3.2 ("local version"). A *globular operad* may equivalently be specified by a globular set P with maps $a\colon P \longrightarrow T1$ ("arity"), $e\colon 1 \longrightarrow P$ ("units"), $m\colon P \times_{T1} TP \longrightarrow P$ ("composition"), such that the diagrams

$$
\begin{tikzcd}
1 \arrow[r, "e"] \arrow[dr, "\eta"'] & P \arrow[d, "a"] \\
& T1
\end{tikzcd}
\qquad
\begin{tikzcd}
& P \times_{T1} TP \arrow[dl] \arrow[dr] \arrow[rrr, "m"] \arrow[dd, phantom, very near start, "\vee"] & & & P \arrow[dddl, "a"] \\
P \arrow[dr, "a"] & & TP \arrow[dl, "T!"'] \arrow[dr, "Ta"] & & \\
& T1 & & T^2 1 \arrow[d, "\mu"'] & \\
& & & T1 &
\end{tikzcd}
$$

commute (i.e. e and m are maps over $T1$), and satisfying the axioms

$$m \cdot (1_P \times \eta \cdot e) = 1_P = m \cdot (e \times \eta)\colon P \longrightarrow P,$$

$$m \cdot (m \times \mu) = m \cdot (1_P \times Tm)\colon P \times_{T1} TP \times_{T^2 1} T^2 P \longrightarrow P.$$

Given an operad in the original form T_P, α, we take $P := T_P 1$, $a := \alpha_1$, $m := \mu_1$, $e := \eta_1$.

Conversely given P, a, e, m, we recover the full monad as $T_P X := TX \times_{T1} P$ (forced by the cartesianness of α), and likewise recover η, μ, α as appropriate pullbacks of e, m, a. In particular, T_P must again be familially representable:

$$(T_P X)_n \cong \sum_{\pi \in T1_n} P(\pi) \times \widehat{\mathbb{G}}(\widehat{\pi}, X),$$

where $P(\pi)$ denotes the fiber of a over $\pi \in T1_n$, the set of "π-ary operations" of P. The map e then gives us a unary n-cell "identity" operation for each n, while the map m allows us to compose these operations appropriately.

As groups have actions, rings have modules, theories have models, so operads have algebras:

Definition 2.3.3. An *algebra* for an operad P is an algebra for the monad T_P; or equivalently, a globular set A together with a map $c\colon P \times_{T1} TA \longrightarrow A$, such that

$$c \cdot (e \times \eta) = 1_A \colon A \longrightarrow A,$$

$$c \cdot (m \times \mu) = c \cdot (1_P \times Tc) \colon P \times_{T1} TP \times_{T^2 1} T^2 A \longrightarrow A.$$

This structure is also called an *action* of P on A.

Actions may also be reformulated in terms of endomorphism operads:

Definition 2.3.4. The *endomorphism operad* $\mathrm{End}_{\widehat{\mathbb{G}}}(A)$ of a globular set A has underlying globular set $\sum_{\pi : T1} [TX_\pi, X]$ (interpreted in the internal language of $\widehat{\mathbb{G}}$, i.e. computed as the internal hom $[TX, X \times T1]$ in the slice $\widehat{\mathbb{G}}/T1$), and structure maps constructed using the universal property of the internal hom, together with the monad structure of T.

Actions of an operad P on a globular set A then correspond to operad maps $P \longrightarrow \mathrm{End}_{\widehat{\mathbb{G}}}(A)$. Via either definition, there is an evident category P-**Alg** of P-algebras.

Example 2.3.5. The object $T1$ itself carries a natural operad structure, making it the terminal operad; its associated monad is just T, and its algebras are strict ω-categories.

We would like to define weak ω-categories as algebras for some operad. However, to give a reasonable theory of some sort of ω-categories, an operad needs to have enough operations to implement the usual compositions and identities; and given any two parallel operations of the same arity in P, they should be an operation of the next dimension connecting them, witnessing that they are "equal up to homotopy". In Leinster's definition, these two conditions are imposed by a single piece of structure:[4]

Definition 2.3.6. A *contraction* on a map of globular sets $f\colon Y \longrightarrow X$ is a diagonal filler for every square

$$\begin{tikzcd} \partial(n) \arrow[r] \arrow[d, tail] & Y \arrow[d, "f"] \\ \mathbf{y}(n) \arrow[r] \arrow[ur, dotted] & X \end{tikzcd}$$

A map is *contractible* if it admits a contraction. Contractible maps are distinguished diagrammatically as $f\colon Y \longrightarrow\!\!\triangleright X$.

So contractibility of $f\colon Y \longrightarrow\!\!\triangleright X$ asserts that given any cell in X and a lifting of its boundary to Y, we may lift the cell, with the given boundary. Or, seen slightly differently, given any boundary in Y and a cell filling its image in X, we may lift the filler to Y.

Recalling the results of 2.1.3, we see that maps-with-contraction are closed under identities, (transfinite) composition, pullback along arbitrary maps, and retraction.

Definition 2.3.7. A *contraction* on an operad P is a contraction on its arity map $a\colon P \longrightarrow\!\!\triangleright T1$, or equivalently, a natural contraction on each component of the natural transformation $\alpha\colon T_P \longrightarrow T$.

[4]This is the sole point where Batanin's original definition differs in more than just presentation.

An *operad-with-contraction* is an operad equipped with a contraction. An operad is *contractible* if it admits a contraction. The essential distinction between these notions is in their maps: a map of operads-with-contraction must commute with their specified contractions.

So roughly, this asserts that T_P is homotopically equivalent to T (see [**Gar09a**] for a precise exploration of this idea). Concretely, contractibility provides us with composition operations of all arities in P, and also with witnesses that these are unique up to coherent higher cells. Any contractible operad may thus be taken as giving a reasonable theory of ω-categories. In particular, for "maximal weakness", let L be the *initial* operad-with-contraction:

Definition 2.3.8 ([**Lei04**])**.** A *weak ω-category* is an L-algebra:

$$\mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat} := L\text{-}\mathbf{Alg}.$$

The universal property of L ensures that if P is any other contractible operad, choosing a contraction χ on P determines a map $f_\chi \colon L \longrightarrow P$ and hence a functor $f_\chi^* \colon P\text{-}\mathbf{Alg} \longrightarrow \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat}$. However, the subtleties of contractible vs. -with-contraction confront us here: this functor will depend upon the choice of contraction (though it seems reasonable to expect that the functors induced by different contractions will differ only up to some notion of weak equivalence).

2.3.9. A further condition imposed by some authors ([**Bat98**], [**GvdB10b**]) is that there are no non-trivial 0-operations: we say an operad P is *normalised* if $P_0 \cong 1$. On the one hand, this assumption rarely seems to be required; contractibility already ensures that 0-operations are unique *up to homotopy*, which according to the usual philosophy of higher-categories is all one ought to need. On the other hand, imposing this condition is essentially harmless: every contractible operad has a canonical normalised contractible sub-operad, its *normalised core*, whose construction we recall now, as it will be of use to us in Ch. 4.

For any $0 \leq k \leq n \leq \omega$, there is a double adjunction

$$\begin{tikzcd} k\text{-}\mathbf{Opd} \arrow[r, shift left=2, "D_n"] \arrow[r, shift right=2, "I_n"'] & n\text{-}\mathbf{Opd} \arrow[l, "\mathrm{tr}^k" description] \end{tikzcd}$$

where $D_n P$ is the *discrete* n-operad on a k-operad P, with only unit cells (that is, units of the operad structure) in dimensions greater than k; $\mathrm{tr}^k Q$ is the *k-truncation* of an n-operad Q, with $\mathrm{tr}^k(Q)_i = Q_i$ for $i \leq k$; and $I_n P$ is the indiscrete n-operad on P, with a unique operation of shape π for each pasting diagram π together with a choice of $\leq n$-dimensional source and target operations from P.

In particular, $0\text{-}\mathbf{Opd} \simeq \mathbf{Mon}$; so for any n and $Q \in n\text{-}\mathbf{Opd}$ we have a unique map $1 \longrightarrow \mathrm{tr}^0 Q$ (where 1 is the trivial monoid, both initial and terminal), and hence can construct the *normalised core* Q^0 of Q,

$$\begin{tikzcd} Q^0 \arrow[r, tail] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & Q \arrow[d] \\ I_n 1 \arrow[r, tail] & I_n \mathrm{tr}^0 Q \end{tikzcd}$$

consisting of just those operations in Q whose 0-dimensional source and target are the operad unit.

It is straightforward to see that Q^0 is contractible if Q is, and so is in this case "weakly equivalent" to Q.

. . . à la Batanin. The preceding elegant presentation of globular operads in terms of cartesian monads is due to Leinster. However, for most of our purposes below it will be easier to construct the desired operads using the original machinery of [**Bat98**].[5] We thus take a brief detour to set up the machinery of *monoidal globular categories*, and to investigate pasting diagrams further, before returning to operads in this setting.

[5]Note howerver that we use Leinster's notion of contraction throughout, so the results are still equivalent to Leinster's definition as given above, rather than Batanin's original one.

Definition 2.3.10. (See [**Bat98**, 2.3] for the details here elided.) A *monoidal globular category* $\mathcal{E}$ is a globular category $\mathcal{E}_\bullet$

$$\begin{tikzcd} \mathcal{E}_0 & \mathcal{E}_1 \arrow[l, shift right, "S"'] \arrow[l, shift left, "T"] & \mathcal{E}_2 \arrow[l, shift right, "S"'] \arrow[l, shift left, "T"] & \mathcal{E}_3 \arrow[l, shift right, "S"'] \arrow[l, shift left, "T"] & \dots \arrow[l, shift right] \arrow[l, shift left] \end{tikzcd}$$

equipped with composition functors

$$\otimes_k \colon \mathcal{E}_n \times_k \mathcal{E}_n \longrightarrow \mathcal{E}_n$$

and unit maps

$$Z \colon \mathcal{E}_n \longrightarrow \mathcal{E}_{n+1}$$

satisfying the same source and target conditions as units and composition in a category, and also with natural transformations

$$\alpha \colon C \otimes_k (B \otimes_k A) \cong (C \otimes_k B) \otimes_k A$$

$$\rho \colon A \otimes_k (Z_n S_k A) \cong A \qquad \lambda \colon (Z_n T_k A) \otimes_k A \cong A$$

$$\chi \colon (D \otimes_j C) \otimes_k (B \otimes_j A) \cong (D \otimes_k B) \otimes_j (C \otimes_k A)$$

satisfying various coherence axioms.

A *monoidal globular functor* between monoidal globular categories $\mathcal{E}, \mathcal{F}$ is a globular functor $F_\bullet \colon \mathcal{E}_\bullet \longrightarrow \mathcal{F}_\bullet$, commuting with the composition and units up to coherent isomorphism. Together with *monoidal globular transformations*, these form a 2-category **MonGlobCat**.

So monoidal globular categories bear essentially the same relationship to strict ω-categories as monoidal categories do to monoids: sets are replaced by categories, and axioms hold just up to coherent isomorphism.

This is made precise by Mark Weber in [**Web05**]: **MonGlobCat** is (2-equivalent to) the 2-category of normalised pseudo-algebras for the "internal strict ω-category in **Cat**" 2-monad $\mathcal{T}$ on $[\mathbb{G}^{\mathrm{op}}, \mathbf{Cat}]$.

Example 2.3.11. For any category $\mathcal{C}$ with all pullbacks, the monoidal globular category $\mathbf{Spans}(\mathcal{C})$ of *higher spans* in $\mathcal{C}$ has

$$\mathbf{Spans}(\mathcal{C})_n = [(\mathbb{G}/n)^{\mathrm{op}}, \mathcal{C}]$$

$$
\begin{tikzcd}
 & A_n \arrow[dl, "s"'] \arrow[dr, "t"] & \\
A^s_{n-1} \arrow[d, "s"'] \arrow[drr, "t" near start] & & A^t_{n-1} \arrow[d, "t"] \arrow[dll, "s"' near start] \\
A^s_{n-2} & & A^t_{n-2} \\
\vdots & & \vdots \\
A^s_1 \arrow[d, "s"'] \arrow[drr, "t" near start] & & A^t_1 \arrow[d, "t"] \arrow[dll, "s"' near start] \\
A^s_0 & & A^t_0
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & A_1 \arrow[dl, equal] \arrow[dr, equal] & \\
A_1 \arrow[d, equal] \arrow[drr, equal] & & A_1 \arrow[d, equal] \arrow[dll, equal] \\
A_1 \arrow[d, "s"'] \arrow[drr, "t" near start] & & A_1 \arrow[d, "t"] \arrow[dll, "s"' near start] \\
A^s_0 & & A^t_0
\end{tikzcd}
$$

FIGURE 2.3.B. A n-span; the identity 3-span on a 1-span.

So an object of $\mathbf{Spans}(\mathcal{C})_n$, an *$n$-span* in $\mathcal{C}$, is a diagram $(A^s_i, A^t_{i\,(i<n)};\ A_n)$ as in Fig. 2.3.B, satisfying the equations $ss = st$, $ts = tt$ wherever appropriate; and a map of n-spans $(f^s_i, f^t_i; f_n)\colon (A^s_i, A^t_i; A_n) \longrightarrow (B^s_i, B^t_i; B_n)$ is a sequence of maps between them, commuting with all the source and target maps.

The source of $(A^s_i, A^t_{i\ (i<n)};\ A_n)$ is then $(A^s_i, A^t_{i\ (i<n-1)};\ A^s_{n-1})$, and similarly its target is $(A^s_i, A^t_{i\ (i<n-1)};\ A^t_{n-1})$.

Identity spans are defined as in Fig. 2.3.B; composition $\otimes_k$ of spans is defined by pullback, as in Fig. 2.3.C.

A pullback-preserving functor $\mathcal{C} \longrightarrow \mathcal{C}'$ induces a monoidal globular functor $\mathbf{Spans}(\mathcal{C}) \longrightarrow \mathbf{Spans}(\mathcal{C}')$.

We will often apply this in the case where $\mathcal{C} = \mathcal{D}^{\mathrm{op}}$, for some familiar $\mathcal{D}$; so then we will work with *co-spans* in $\mathcal{D}$, whose composites are computed by pushouts, and so on.

Although of course formally equivalent, there is an important difference in intuition between the two orientations. When working in $\mathbf{Spans}(\mathcal{C})$, the objects of a span are typically objects *containing* cells, such as the sets of n-cells of a higher category. When working in $\mathbf{Spans}(\mathcal{D}^{\mathrm{op}})$, we consider

$$
\begin{tikzcd}[column sep=small]
& A_4 \arrow[dl] \arrow[dr] & & & & B_4 \arrow[dl] \arrow[dr] & \\
A_3^s \arrow[d] \arrow[drrr] & & A_3^t \arrow[dll] \arrow[dr] & & B_3^s \arrow[dl] \arrow[drr] & & B_3^t \arrow[dlll] \arrow[d] \\
A_2^s \arrow[dr] \arrow[drrr] & & & A_2^t = B_2^s \arrow[dll] \arrow[d] & & & B_2^t \arrow[dlllll] \arrow[dlll] \\
& A_1^s = B_1^s \arrow[d] \arrow[drr] & & A_1^t = B_1^t \arrow[dll] \arrow[d] & & & \\
& A_0^s = B_0^s & & A_0^t = B_0^t & & &
\end{tikzcd}
\longmapsto
\begin{tikzcd}[column sep=small]
& A_4 \times_{A_2^t} B_4 \arrow[dl] \arrow[dr] & \\
A_3^s \times_{A_2^t} B_3^s \arrow[d] \arrow[drr] & & A_3^t \times_{A_2^t} B_3^s \arrow[dll] \arrow[d] \\
A_2^s \arrow[d] \arrow[drr] & & B_2^t \arrow[dll] \arrow[d] \\
A_1^s = B_1^s \arrow[d] \arrow[drr] & & A_1^t = B_1^t \arrow[dll] \arrow[d] \\
A_0^s = B_0^s & & A_0^t = B_0^t
\end{tikzcd}
$$

FIGURE 2.3.C. A 2-composition of 4-spans $B \otimes_2 A$

the objects of a span more often as objects *representing* cells, such as the topological globes D^n or the universal globes $\mathbf{y}(n)$ in $\widehat{\mathbb{G}}$, with "co-source" and "co-target" maps $s, t \colon A_i^{s,t} \longrightarrow A_{i+1}$ embedding the lower-dimensional globes as the sources or targets of higher ones.

The crucial function of monoidal globular categories is as the setting within which one can define globular operads and their algebras. We will not need the former in the present work, only the latter.

Definition 2.3.12. A *globular object* $A_\bullet$ of a monoidal globular category $\mathcal{E}$ is a globular functor $1 \longrightarrow \mathcal{E}_\bullet$, where 1 is the terminal globular category. Concretely, a globular object in $\mathcal{E}$ is a sequence of objects $A_i \in \mathcal{E}_i$, with $S(A_{i+1}) = T(A_{i+1}) = A_i$.

Globular objects in $\mathbf{Spans}(\mathcal{C})$ correspond precisely to globular objects in $\mathcal{C}$. (We often abuse notation here by writing A_n both for a span and for the object at its apex.)

Given a globular object $A_\bullet$ of a monoidal globular category $\mathcal{E}$, we may extend it to a monoidal globular functor $\mathbf{pd} \longrightarrow \mathcal{E}$ (where the sets of $\mathbf{pd}$

are regarded here as discrete categories), the *diagram objects* of $A_\bullet$. This follows immediately from the universal property of $T1$ together with Weber's description of monoidal globular categories as certain pseudo-algebras for $\mathcal{T}$:

$$\begin{tikzcd} \mathbf{pd} = \mathcal{T}1 \arrow[r, "\mathcal{T}A_\bullet"] & \mathcal{T}\mathcal{E}_\bullet \arrow[d, "\otimes"] \\ 1 \arrow[r, "A_\bullet"] & \mathcal{E}_\bullet \end{tikzcd} .$$

We denote objects in the image of this extended functor by A^π, for $\pi \in \mathbf{pd}$. Intuitively, if A_n is the object of n-cells of $A_\bullet$, then A^π is the object of diagrams of shape π in $A_\bullet$; or in the dual case, if C_n is a representing object for n-cells, then C^π represents diagrams of shape π.

In particular, $\mathbf{y}\colon \mathbb{G} \longrightarrow \widehat{\mathbb{G}}$ gives a globular object in $\mathbf{Spans}(\widehat{\mathbb{G}}^{\mathrm{op}})$, whose diagram objects are exactly the realisations $\widehat{\pi}$ constructed in 2.2.7 above.

2.3.13. Now if $\mathcal{C}$ is co-complete, and $C_\bullet : \mathbb{G} \longrightarrow \mathcal{C}$ is a co-globular object, we may form the left Kan extension ([**ML98**, X.3]) of $C_\bullet$ along the Yoneda embedding $\mathbf{y}\colon \mathbb{G} \rightarrowtail \widehat{\mathbb{G}}$, analogous to the "geometric realisation" of globular or simplicial objects in **Top**. $\mathrm{Lan}_{\mathbf{y}} C_\bullet$ realises any globular set as a colimit in $\mathcal{C}$, using the objects C_n as templates for the cells.

$$\begin{tikzcd} \widehat{\mathbb{G}} \arrow[r, "\mathrm{Lan}_{\mathbf{y}} C_\bullet"] & \mathcal{C} \\ \mathbb{G} \arrow[u, tail, "\mathbf{y}"] \arrow[ur, "C_\bullet"'] & \end{tikzcd}$$

Since $\mathrm{Lan}_{\mathbf{y}} C_\bullet$ preserves colimits, it lifts to give a monoidal globular functor $\mathbf{Spans}((\mathrm{Lan}_{\mathbf{y}} C_\bullet)^{\mathrm{op}})\colon \mathbf{Spans}(\widehat{\mathbb{G}}^{\mathrm{op}}) \longrightarrow \mathbf{Spans}(\mathcal{C}^{\mathrm{op}})$, and since it sends the basic globes $\mathbf{y}$ to $C_\bullet$, we have moreover a commutative (up to natural isomorphism) triangle:

$$\begin{tikzcd} & \mathbf{Spans}(\widehat{\mathbb{G}}^{\mathrm{op}}) \arrow[dr] & \\ \mathbf{pd} \arrow[ur, "\mathbf{y}"] \arrow[rr, "C_\bullet"'] & & \mathbf{Spans}(\mathcal{C}^{\mathrm{op}}) \end{tikzcd}$$

Thus we may compute the diagram objects of $C_\bullet$ as the image under $\mathrm{Lan}_{\mathbf{y}}C_\bullet$ of those of $\mathbf{y}$:

$$C^\pi = \mathrm{Lan}_{\mathbf{y}}C_\bullet(\widehat{\pi}) = \varinjlim_{c \in \int \widehat{\pi}} C_{\dim c}$$

This formula is easiest explained by example: if $\pi = \left(\begin{tikzcd} \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow"] & \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow"] & \bullet \end{tikzcd} \right)$, then

$$\begin{aligned} C_\pi &\cong \varinjlim \left(\begin{tikzcd} & C_1 \arrow[d, "s"'] & & C_1 \arrow[d, "s"'] & \\ C_0 \arrow[ur, "s"] \arrow[dr, "s"'] & C_2 & C_0 \arrow[ul, "t"'] \arrow[dl, "t"] \arrow[ur, "s"] \arrow[dr, "s"'] & C_2 & C_0 \arrow[ul, "t"'] \arrow[dl, "t"] \\ & C_1 \arrow[u, "t"'] & & C_1 \arrow[u, "t"'] & \end{tikzcd} \right) \\ &\cong C_2 +_{C_0} C_2, \end{aligned}$$

or in the dual case,

$$\begin{aligned} A_\pi &\cong \varprojlim \left(\begin{tikzcd} & A_1 \arrow[dl, "s"'] \arrow[dr, "t"] & & A_1 \arrow[dl, "s"'] \arrow[dr, "t"] & \\ A_0 & A_2 \arrow[u, "s"'] \arrow[d, "t"'] & A_0 & A_2 \arrow[u, "s"'] \arrow[d, "t"'] & A_0 \\ & A_1 \arrow[ul, "s"] \arrow[ur, "t"'] & & A_1 \arrow[ul, "s"] \arrow[ur, "t"'] & \end{tikzcd} \right) \\ &\cong A_2 \times_{A_0} A_2, \end{aligned}$$

giving the object of 0-composable pairs of 2-cells in A.

A slightly more careful calculation shows that diagram objects are computed by this formula even when $\mathcal{C}$ is not co-complete.

We can now use this to define internal algebras for globular operads, in monoidal globular categories:

Definition 2.3.14 ([**Bat98**, 7.2])**.** If $A_\bullet$ is a globular object in a monoidal globular category $\mathcal{E}$, there is a globular operad $\mathrm{End}_{\mathcal{E}}(A_\bullet)$, the *endomorphism operad* of $A_\bullet$, given by $\mathrm{End}_{\mathcal{E}}(A_\bullet)(\pi) = \mathcal{E}_n(A^\pi, A_n)$, for each $\pi \in \mathbf{pd}_n$.

A monoidal globular functor $F \colon \mathcal{E} \longrightarrow \mathcal{F}$ induces a map of endomorphism operads $\mathrm{End}_{\mathcal{E}}(A_\bullet) \longrightarrow \mathrm{End}_{\mathcal{F}}(FA_\bullet)$.

In the case $\mathcal{E} = \mathbf{Spans}(\mathbf{Sets})$, this agrees with our earlier description of the endomorphism operad of a globular set.

Definition 2.3.15. For a globular operad P, a P-algebra structure on a globular object $A_\bullet$ of $\mathcal{E}$ is an operad map $P \longrightarrow \mathrm{End}_{\mathcal{E}}(A_\bullet)$.

We then call $A_\bullet$ an *(internal) P-algebra* in $\mathcal{E}$; or when $\mathcal{E} = \mathbf{Spans}(\mathcal{C})$, a P-algebra in $\mathcal{C}$, or when $\mathcal{E} = \mathbf{Spans}(\mathcal{D}^{\mathrm{op}})$, a co-$P$-algebra in $\mathcal{D}$.

2.3.16. As usual with algebraic structures, homming into internal operad algebras yields external ones. If X is any object of a category $\mathcal{C}$, then there is an obvious globular functor

$$\mathrm{Hom}(X, -) \colon \mathbf{Spans}(\mathcal{C}) \longrightarrow \mathbf{Spans}(\mathbf{Sets}),$$

which is moreover monoidal globular; so if $A_\bullet$ is an internal P-algebra in $\mathcal{C}$, we have operad maps

$$P \longrightarrow \mathrm{End}_{\mathbf{Spans}(\mathcal{C})}(A_\bullet) \longrightarrow \mathrm{End}_{\mathbf{Spans}(\mathbf{Sets})}(\mathrm{Hom}(X, A_\bullet))$$

and hence a P-algebra structure on the globular set $\mathrm{Hom}(X, A_\bullet)$. Similarly, homming *out* of a co-algebra yields an algebra.

2.4. Contracting pasting diagrams

(Unlike the rest of this chapter, the material of this section is somewhat original; the basic concepts are not especially new, but the specific techniques used here do not seem to predate [**Lum10**] and [**GvdB10b**].)

2.4.1. Some early presentations of pasting diagrams introduced them as isomorphism classes of *contractible globular sets*. Here, we describe a methodical procedure for contracting them. Specifically, we give a partial operation $\pi \mapsto \pi^-$, which under repeated application eventually reduces any pasting diagram π to the trivial pasting diagram $\bullet$.

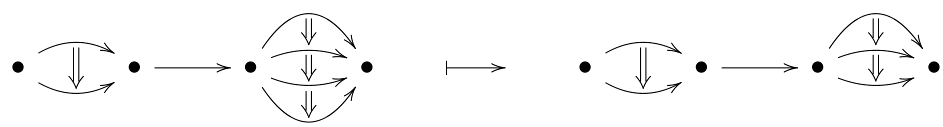
Using the free monoid representation, $()$ is the trivial pasting diagram $\bullet$; it is already contracted, so $()^-$ is not defined. When π is the path of length

$l > 1$

$$(\underbrace{(),\ldots,()}_{l})$$

we take π^- to be the path of length $l-1$. For any other pasting diagram $\pi = (\pi_1, \ldots, \pi_l)$, with $l \geq 1$ and π_i not all equal to $()$, we take $\pi^- := (\pi_1, \ldots, \pi_i^-, \ldots, \pi_k)$, where i is minimal such that π_i is not already contracted.

Roughly, this operation removes the leftmost cell (in algebraic order, or rightmost in diagrammatic order) of dimension > 1 if there are any such, or of dimension 1 if there are none higher. It is clear by induction on e.g. dimension that under repeated application of $(-)^-$, any pasting diagram eventually reaches $()$.



π $\qquad\qquad\qquad\qquad$ π^-

2.4.2. We will need to know a little about the realisations of these; in fact, some apparently very crude statements will be enough.

For any non-trivial pasting diagram π, there is an inclusion

$$\widehat{\pi^-} \rightarrowtail \widehat{\pi}$$

whose image consists of all of π except for two cells, one the target of the other. When π is just a path, this condition determines what the inclusion must be; otherwise it is constructed by recursion on π^-, using the recursive definition of π^- together with the explicit description of $\widehat{\pi}$ in 2.2.6.

It follows from this description of the image that we have a pushout square, attaching a k-cell along a $(k-1)$-cell:

$$\begin{tikzcd}
\mathbf{y}(k-1) \arrow[r] \arrow[d, "s"'] & \widehat{\pi^-} \arrow[d] \\
\mathbf{y}(k) \arrow[r] \arrow[ur, phantom, "\ulcorner" very near end] & \widehat{\pi}
\end{tikzcd}$$

Hence if $A_\bullet$ (resp. $C_\bullet$) is a globular (co-globular) object in any (co-)complete category, we obtain by applying the appropriate Kan extension a pullback (pushout) square:

$$\begin{tikzcd} A^\pi \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & A_k \arrow[d, "s"] \\ A^{\pi^-} \arrow[r] & A_{k-1} \end{tikzcd} \qquad \begin{tikzcd} C_{k-1} \arrow[r] \arrow[d, "s"'] & C^{\pi^-} \arrow[d] \\ C_k \arrow[r] \arrow[ur, phantom, very near end, "\ulcorner"] & C^\pi \end{tikzcd}$$

(again, a slightly more careful calculation avoids the need for (co-)completeness, but for our purposes the (co-)complete case will suffice).

Finally, for any non-trivial π, we either have $s(\pi^-) = s\pi$, if the dimension of the cell removed is the dimension of π, or $s(\pi^-) = (s\pi)^-$ otherwise; in either case, the resulting square

$$\begin{tikzcd} \widehat{s(\pi^-)} \arrow[r, "s"] \arrow[d] & \widehat{\pi^-} \arrow[d] \\ \widehat{s(\pi)} \arrow[r, "s"] & \widehat{\pi} \end{tikzcd}$$

commutes, as does its realisation over any (co-)globular object; and similarly with t in place of s.

2.4.3. As presented here, this construction is rather ad hoc. A more satisfying analysis of this sort of step-by-step contraction can be given using the Batanin tree representation of pasting diagrams ([**Bat98**, 3],[**Lei04**, 8.1]): schemes for contracting a pasting diagram in this manner correspond precisely to ways of sequentially pruning leaves off its Batanin tree.

However, since we do not require the Batanin tree representation for any other purpose, for brevity we give just this formulaic procedure, which suffices for the applications in the present work.

CHAPTER 3

The fundamental weak ω-groupoid of a type

In this chapter, we will construct the fundamental weak ω-category of a type, as sketched in the introduction (0.2.1).

As explained there, the goal is to put a weak ω-category structure on the sets of closed terms of type A, Id_A, $\mathsf{Id}_{\mathsf{Id}_A}$, $\ldots$, for any type A in any theory $\mathbf{T}$. This is a categorically familiar scenario: these are sets of *global elements*, so to endow them with some algebraic structure, one need only put an internal such structure on the objects themselves. So, we proceed accordingly, putting an internal weak ω-category structure on A, Id_A, $\ldots$ themselves in (the classifying category of) $\mathbf{T}$.

To do this, of course, we need to find some contractible operad of composition laws which acts on these. In order to achieve this uniformly over all types in all theories, we consider the syntactic operad of *generically definable* composition operations, i.e. operations which can be derived over a type without any further rules or constructors assumed. Formally, we construct this as the endomorphism operad $\mathrm{End}_{\widehat{\mathbf{T}_{\mathsf{Id}}[X]}}(\underline{X}_\bullet)$, where $\mathbf{T}_{\mathsf{Id}}[X]$ is the theory given just by the Id-type rules together with a single closed type X.

We then show directly that this operad is indeed contractible, using a little specific analysis of the theory $\mathbf{T}_{\mathsf{Id}}[X]$, and discuss extensions of this result to larger operads.

3.1. Construction of P_{Id}

3.1.1. We saw above that for a type A in a theory $\mathbf{T} \in \mathbf{DTT}_{\mathsf{Id}}$, the contexts

$$x_0, y_0 : A,\ x_1, y_1 : \mathsf{Id}(x_0, y_0) \ldots,\ z : \mathsf{Id}(x_{n-1}, y_{n-1}),$$

and the dependent projections src, tgt between them form a globular context $\underline{A}_\bullet \colon \mathbb{G}^{\mathrm{op}} \longrightarrow \mathbf{T}$. (In denoting these contexts $\underline{A}_n$, we roughly follow [**War08**].)

We would like to describe the endomorphism operad of this object; unfortunately, $\mathbf{T}$ does not have all finite limits in general, so we cannot immediately apply the **Spans** construction and the machinery of 2.3.14.

However, taking presheaves embeds $\mathbf{T}$ into a category with the necessary limits, so we can certainly consider $\mathrm{End}_{\widehat{\mathbf{T}}}(\mathbf{y}\underline{A}_\bullet)$. Since the Yoneda embedding is full and faithful and preserves all existing limits, if we can show that all the limits $\underline{A}_\pi$ exist in $\mathbf{T}$, then we know that the explicit description of $\mathrm{End}_{\widehat{\mathbf{T}}}(\mathbf{y}\underline{A}_\bullet)$ may be computed directly in $\mathbf{T}$:

$$\mathrm{End}_{\widehat{\mathbf{T}}}(\mathbf{y}\underline{A}_\bullet)(\pi) \cong \mathbf{Spans}(\widehat{\mathbf{T}})_n((\mathbf{y}\underline{A}_\bullet)^\pi, \mathbf{y}\underline{A}_n) \cong \mathbf{Spans}(\mathbf{T})_n(A^\pi, \underline{A}_n)$$

We thus use Proposition 1.4.2 to construct contexts Γ_π exhibiting the limits $\underline{A}_\pi$.

Accordingly, suppose we are given $\pi \in \mathbf{pd}_n$, with associated globular set $\hat{\pi}$. There are various ways of putting a total order on the i-cells of $\hat{\pi}$ for each $i \leq n$; pick any such.

(Indeed, a canonical choice is given by the $\blacktriangleleft$ ordering of [**Str00**], easily seen in terms of the description of $\widehat{\pi}$ via the Batanin tree of π. This choice has some good compatibility between the orderings on different pasting diagrams, which will later spare us some use of EXCH rules, so for simplicity we will assume it is the ordering chosen; however, this is purely cosmetic, and any other choice of orderings could also be used.)

Then take Γ_π to be the context

$$\bigwedge_{c \in \hat{\pi}_0} x_c \colon A,\ \bigwedge_{c \in \hat{\pi}_1} x_c \colon \mathsf{Id}(x_{s(c)}, x_{t(c)}),\ \ldots\ \bigwedge_{c \in \hat{\pi}_n} x_c \colon \mathsf{Id}(x_{s(c)}, x_{t(c)}).$$

For instance, $\Gamma_{(\bullet \to \bullet \to \bullet)}$ is the context

$$x, y, z : A,\ p : \mathsf{Id}_A(x, y),\ q : \mathsf{Id}_A(y, z)$$

which we met back in 0.2.1.

Besides the contexts themselves, we of course also have source and target maps $\mathsf{src}, \mathsf{tgt} \colon \Gamma_\pi \longrightarrow \Gamma_{d\pi}$, and so on.

Lemma 3.1.2. *The context $\vec{x} : \Gamma_\pi$, together with the obvious dependent projections, computes the limit $\Gamma_\pi = \varprojlim_{c \in \int \pi} \underline{A}_{\dim c}$. Moreover, if $F \colon \mathsf{T} \longrightarrow \mathsf{S}$ is a translation of type theories, then $\mathcal{C}\ell(F) \colon \mathcal{C}\ell(\mathsf{T}) \longrightarrow \mathcal{C}\ell(\mathsf{S})$ preserves this limit.*

Proof. Immediate by Proposition 1.4.2 $\square$

Thus the description of $\mathrm{End}_{\widehat{\mathsf{T}}}(\mathbf{y}\underline{A}_\bullet)$ in terms of maps of spans in $\widehat{\mathsf{T}}$ may be applied directly in T, using the contexts Γ_π; with this justification, we write it simply as $\mathrm{End}_{\mathsf{T}}(\underline{A}_\bullet)$.

3.1.3. Let us unfold what this looks like in syntactic terms. For $\pi \in \mathbf{pd}_n$, an element of $\mathrm{End}_{\mathsf{T}}(\underline{A}_\bullet)(\pi)$ (a *composition law for* π) consists of a map of spans as in Fig. 3.1.A; that is, a context map $h \colon \Gamma_\pi \longrightarrow \underline{A}_n$ in T, and for $0 \leq k < n$, maps $f_k \colon \Gamma_{d^{n-k}(\pi)} \longrightarrow \underline{A}_k$ and $g_k \colon \Gamma_{d^{n-k}(\pi)} \longrightarrow \underline{A}_n$, commuting appropriately with the dependent projections.

$$\begin{tikzcd}
 & \Gamma_\pi \arrow[rrr, "\vec{h}"] \arrow[dl] \arrow[ddr] & & & \underline{A}_n \arrow[dl] \arrow[ddr] & \\
\Gamma_{s\pi} \arrow[rrr, "\vec{f}_{n-1}"] \arrow[dd] \arrow[dddrr] & & & \underline{A}_{n-1} \arrow[dd] \arrow[dddrr] & & \\
 & & \Gamma_{t\pi} \arrow[rrr, "\vec{g}_{n-1}"] \arrow[dll] \arrow[dd] & & & \underline{A}_{n-1} \arrow[dll] \arrow[dd] \\
\Gamma_{s^2\pi} \arrow[rrr, "\vec{f}_{n-2}"] & & & \underline{A}_{n-2} & & \\
 & & \Gamma_{t^2\pi} \arrow[rrr, "\vec{g}_{n-2}"] & & & \underline{A}_{n-2} \\
\vdots & & \vdots & \vdots & & \vdots \\
\Gamma_{s_1\pi} \arrow[rrr, "\vec{f}_1"] \arrow[dd] \arrow[dddrr] & & & \underline{A}_1 \arrow[dd] \arrow[dddrr] & & \\
 & & \Gamma_{t_1\pi} \arrow[rrr, "\vec{g}_1"] \arrow[dll] \arrow[dd] & & & \underline{A}_1 \arrow[dll] \arrow[dd] \\
\Gamma_{s_0\pi} \arrow[rrr, "\vec{f}_0"] & & & \underline{A}_0 & & \\
 & & \Gamma_{t_0\pi} \arrow[rrr, "\vec{g}_0"] & & & \underline{A}_0
\end{tikzcd}$$

FIGURE 3.1.A. An operation in $\mathrm{End}_{\mathsf{T}}(\underline{A}_\bullet)(\pi)$.

So, concretely, it is a sequence of terms $\vec{h} = ((f_i, g_i)_{0 \leq i < n}; h)$, such that

$$\begin{aligned}
\vec{x} : \Gamma_{d^n(\pi)} &\vdash f_0(\vec{x}) : A \\
\vec{x} : \Gamma_{d^n(\pi)} &\vdash g_0(\vec{x}) : A \\
&\vdots \\
\vec{x} : \Gamma_{d^{n-k}(\pi)} &\vdash f_k(\vec{x}) : \mathsf{Id}(f_{k-1}(\mathsf{src}\,\vec{x}), g_{k-1}(\mathsf{tgt}\,\vec{x})), \\
\vec{x} : \Gamma_{d^{n-k}(\pi)} &\vdash g_k(\vec{x}) : \mathsf{Id}(f_{k-1}(\mathsf{src}\,\vec{x})), g_{k-1}(\mathsf{tgt}\,\vec{x})), \\
&\vdots \\
\vec{x} : \Gamma_{\pi} &\vdash h(\vec{x}) : \mathsf{Id}(f_{n-1}(\mathsf{src}\,\vec{x}), g_{n-1}(\mathsf{tgt}\,\vec{x})).
\end{aligned}$$

The source of this is then the composition law $(f_0, g_0, \ldots, f_{n-1}, g_{n-1}; f_n) \in P(s(\pi))$, and its target is $(f_0, g_0, \ldots, f_{n-1}, g_{n-1}; g_n) \in P(t(\pi))$.

We make no attempt here to formulate a syntactic description of the operad composition; to do so is straightforward, but notationally rather heavy going. In specific cases it is "exactly what you would expect": essentially just substitution, with modest assistance from the other structural rules.

3.1.4. Following the approach outlined in the introduction, we wish to isolate the operad of composition operations which are derivable for *all* types. Consequently, we consider the theory $\mathbf{T}_{\mathsf{Id}}[X]$ of a generic type, given by the Id-type rules together with just a single extra axiom:

$$\frac{}{\diamond \vdash X\ \mathsf{type}}$$

The genericity of X becomes a universal mapping property of $\mathbf{T}_{\mathsf{Id}}[X]$: for any type theory $\mathbf{S}$ and closed type A of $\mathbf{S}$, there is a unique translation $F_A \colon \mathbf{T}_{\mathsf{Id}}[X] \longrightarrow \mathbf{S}$ sending X to A.

Definition 3.1.5. As as special case of the construction of 3.1.1, we take

$$P_{\mathsf{Id}} := \mathrm{End}_{\mathbf{T}_{\mathsf{Id}}[X]}(\underline{X}_{\bullet}),$$

the operad of all definable composition laws on the generic type.

3.1.6. For general T, A, we should not expect $\mathrm{End}_{\mathsf{T}}(\underline{A}_{\bullet})$ to be contractible: contractibility implies for instance that any two elements of $\mathrm{End}_{\mathsf{T}}(\underline{A}_{\bullet})(\bullet)$ are connected by an element of $\mathrm{End}_{\mathsf{T}}(\underline{A}_{\bullet})(\bullet \longrightarrow \bullet)$, or in other words that any two terms $x : A \vdash f(x), f'(x) : A$ are propositionally equal, which clearly may fail.

However, in the specific case of P_{Id}, we do wish to show contractibility, since this is the operad which naturally acts on every type; and it seems plausible, since obvious failures of contractibility require assuming extra term-constructors for A or its identity types.

What precisely does contractibility mean, here? For every pasting diagram π and every parallel pair of composition laws $\vec{f}, \vec{g} \in P_{\mathsf{Id}}(d(\pi))$, we need to find some filler $\vec{h} \in P_{\mathsf{Id}}(\pi)$, with $s(\vec{h}) = \vec{f}$, $t(\vec{h}) = \vec{g}$.

Given π, such a parallel pair amounts to terms $(f_i, g_i)_{0 \leq i < n}$ as in the definition of a composition law for π, and a filler is a term h completing the definition; that is, we seek to derive a judgment

$$\vec{x} : \Gamma_{\pi} \vdash h(\vec{x}) : \mathsf{Id}(f_{n-1}(\mathsf{src}\,\vec{x}), g_{n-1}(\mathsf{tgt}\,\vec{x})).$$

Playing with small examples (the reader is strongly encouraged to try this—to derive, for instance, the composition and associativity terms mentioned in 0.2.3) suggests that we should be able to do this by applying Id-ELIM (possibly repeatedly, working bottom-up as usual) to the variables of identity types in Γ_{π}. Id-ELIM says that to obtain h, it's enough to obtain it in the case where one of the variables is of the form $r(-)$, and its source and target variables are equal; and by repeated application, it's enough to obtain h in the case where multiple higher cells have had identities plugged in in this way.

Now, since the terms f_i, g_i have themselves been built up from just the Id-rules, as we plug $r(-)$ terms into them and identify the lower variables, they should sooner or later compute down by Id-COMP to be of the form $r^i(x)$ themselves. Eventually, after applying Id-ELIM as far as possible, plugging in reflexivity terms for the higher variables and contracting all variables of type X to a single $x : X$, the f_i, g_i should *all* reduce to reflexivity terms; and in particular f_{n-1} and g_{n-1} should both reduce to the form $r^{n-1}(x)$, so we can take the desired filler to be

$$x : X \vdash r^n(x) : \mathsf{Id}(r^{n-1}(x), r^{n-1}(x)).$$

Below, we formalise this argument. The crucial lemma is that the context $x : X$ is an initial object in $\mathcal{C}\ell(\mathbf{T}_{\mathsf{Id}}[X])$: that is, since any context Γ in $\mathbf{T}_{\mathsf{Id}}[X]$ is built up just from X and its higher identity types, there is always a unique way to substitute x and its reflexivity terms $r^i(x)$ for all variables of Γ, and when we substitute these in to any context morphism $f \colon \Gamma \longrightarrow \Gamma'$, the result must again reduce to terms of this form.

3.2. A proof-theoretic lemma: X is initial in $\mathbf{T}_{\mathsf{Id}}[X]$

Lemma 3.2.1. *The context $x : X$ is an initial object in $\mathcal{C}\ell(\mathbf{T}_{\mathsf{Id}}[X])$; that is, for any closed context Γ there is a unique context map $r^\Gamma \colon (x{:}X) \longrightarrow \Gamma$.*

Note that this lemma does not generally hold for more powerful type systems; e.g. in $\mathbf{T}_{\mathsf{Id},\Pi}[X]$, it is easily seen to be false, since for instance there is no term $x : X \vdash \tau : \Pi_{y:X} \mathsf{Id}(x, y)$.

We give here two proofs of this lemma; or rather, the same proof in two forms. The first, much shorter, is in categorical terms, using the co-slice construction and the universal property of $\mathbf{T}_{\mathsf{Id}}[X]$. The second is a direct syntactic proof, as given in [**Lum10**]. Essentially, this unwinds into a structural induction the construction of the CwA-with-Id-types structure on the co-slice, and hence shows concretely how the maps r^Γ are constructed.

Proof 1. By the universal property of $\mathbf{T}_{\mathsf{Id}}[X]$, the object $1_X \colon X \longrightarrow X$ of the coslice CwA $X\backslash\mathbf{T}_{\mathsf{Id}}[X]$ induces a translation $F_{1_X} \colon \mathbf{T}_{\mathsf{Id}}[X] \longrightarrow X\backslash\mathbf{T}_{\mathsf{Id}}[X]$ sending X to 1_X. The composition of this with the forgetful functor from the co-slice $U \colon X\backslash\mathbf{T}_{\mathsf{Id}}[X] \longrightarrow \mathbf{T}_{\mathsf{Id}}[X]$ is an endofunctor of $\mathbf{T}_{\mathsf{Id}}[X]$ which fixes X, and hence (by the universal property again) is the identity.

Thus for any context Γ of T_{Id}, $F_{1_X}(\Gamma)$ is some context map $r^\Gamma\colon X \longrightarrow \Gamma$, and for any context map $f\colon \Delta \longrightarrow \Gamma$, $F_{1_X}(f)$ must be just f itself viewed as a map of $X\backslash\mathsf{T}_{\mathsf{Id}}[X]$, so every triangle

$$\begin{tikzcd} & \Delta \arrow[d, "f"] \\ X \arrow[ur, "r^\Delta"] \arrow[dr, "r^\Gamma"'] & \\ & \Gamma \end{tikzcd}$$

must commute. In particular, with $\Delta = X$, this tells us (since $r^X = F_{1_X}(X) = 1_X$) that $f = r^\Gamma$ for any context morphism $f\colon X \longrightarrow \Gamma$, so X is initial as desired.

(This last step is just an instance of the general categorical fact that given an object X in a category $\mathcal{C}$ and natural maps $!_Y\colon X \longrightarrow Y$ to every other object, such that $!_X = 1_X$, it follows that X is initial.) $\square$

Proof 2. For the syntactic proof, we work by structural induction—as, essentially, we must, since this is a property of the theory $\mathsf{T}_{\mathsf{Id}}[X]$ which can fail in extensions.

So, given any derivation δ of a judgement J in $\mathsf{T}_{\mathsf{Id}}[X]$, we recursively derive various terms and/or judgments, depending on the form of J, assuming that we have already done so for all sub-derivations of δ. The form of the terms and judgements we derive will depend on the form of J as follows:

J	term	judgement	
$\vec{y} : \Gamma \vdash A(\vec{y})\ \mathsf{type}$	$r^{\Gamma \vdash A}(x)$	$x : X \vdash r^{\Gamma \vdash A}(x) : A(\vec{r}^{\Gamma}(x))$	
$\vec{y} : \Gamma \vdash A(\vec{y}) = A'(\vec{y})\ \mathsf{type}$	–	$x : X \vdash r^{\Gamma \vdash A}(x) = r^{\Gamma \vdash A'}(x) : A(\vec{r}^{\Gamma}(x))$	$(*)$
$\vec{y} : \Gamma \vdash \tau(\vec{y}) : A(\vec{y})$	–	$x : X \vdash \tau(r^{\Gamma}(x)) = r^{\Gamma \vdash A}(x) : A(r^{\Gamma}(x))$	$(**)$
$\vec{y} : \Gamma \vdash \tau(\vec{y}) = \tau'(\vec{y}) : A(\vec{y})$	–	–	

Here, for a context $\Gamma = y_0 : A_0, \ldots, y_n : A_n(\vec{y}_{<n})$, the context map $\vec{r}^{\Gamma}\colon (x\!:\!X) \longrightarrow \Gamma$ consists of the terms

$$r^{\vdash A_0}(x) : A_0, \quad r^{A_0 \vdash A_1}(x) : A_1(r^{\vdash A_0}(x)), \quad r^{A_0, A_1 \vdash A_2}(x) \ldots\ .$$

Moreover, applying (*) and (**) above to this definition shows that the maps $\vec{r}^{\Gamma}$ respect definitional equality in Γ, and are preserved by context maps in that for any $f\colon \Delta \longrightarrow \Gamma$, we have $f(\vec{r}^{\Delta}(x)) = \vec{r}^{\Gamma}(x)$.

Finally, once the induction is complete, applying this last fact together with the fact that $r^{(x X)}(x) := x$ will show that for any other context map $f\colon (x\colon X) \longrightarrow \Gamma$, we have $f(x) = f(r^{(x X)}) = \vec{r}^{\Gamma}(x)$, and so $\vec{r}^{\Gamma}$ is the unique such map, as originally desired.

(This is of course the same concluding step we used in the first proof.)

As usual, the induction proceeds by cases on the last rule used in the derivation of J. Care must be taken to ensure that the definition of $r^{\Gamma \vdash A}$ is appropriately stable under substitution, but this aside, all cases are essentially routine; details may be found in [**Lum10**]. □

3.3. Contractibility of P_{Id}

We are now ready to show that P_{Id} is contractible, arguing along the lines sketched in 3.1.6.

Theorem 3.3.1. *The operad P_{Id} is contractible.*

Once again, we offer both an elementary syntactic argument and a categorical gloss of the same proof.

Proof 1. As described above, this amounts to the statement: for every $n \in \mathbb{N}$ and pasting diagram $\pi \in \mathbf{pd}_n$, and every sequence $(f_i, g_i)_{i<n}$ of terms such that

$$\vec{x} : \Gamma_{d^n(\pi)} \vdash f_0(\vec{x}),\ g_0(\vec{x}) : X$$

$$\vec{x} : \Gamma_{d^{n-i}(\pi)} \vdash f_i(\vec{x}),\ g_i(\vec{x}) : \mathsf{Id}(f_{i-1}(\mathsf{src}\,\vec{x}), g_{i-1}(\mathsf{tgt}\,\vec{x}))$$

$(i < n)$ are derivable in $\mathbf{T}_{\mathsf{Id}}[X]$, we can find a "filler", i.e. a term h with

$$\vec{x} : \Gamma_{\pi} \vdash h(\vec{x}) : \mathsf{Id}(f_{n-1}(\mathsf{src}\,\vec{x}), g_{n-1}(\mathsf{tgt}\,\vec{x}))$$

We show this by induction on the number of cells in π, reducing cells by the "pruning" process of 2.4.1.

Suppose π has cells in some dimension greater 0. Then as in 2.4.1, consider the pasting diagram π^-, obtained (up to isomorphism) from that of π by removing some cell c and identifying $s(c)$ and $t(c)$.

Then by our explicit description of the cells of π^-, the context Γ_{π^-} is exactly (up to renaming of variables, and possibly re-ordering if we do not assume that we chose compatible orderings of the cells of pasting diagrams) the context obtained from Γ_π by removing the variables x^k_c and $x^{k-1}_{t(c)}$, and substituting $x^{k-1}_{s(c)}$ for any occurrences of the latter in subsequent types; and we have a natural context map $\vec{r}_\pi \colon \Gamma_{\pi^-} \longrightarrow \Gamma_\pi$ given by plugging in $x^{k-1}_{s(c)}$ for $x^{k-1}_{t(c)}$ and $r(x^{k-1}_{s(c)})$ for x^k_c; and these are exactly right for

$$\frac{\vec{x} : \Gamma_{\pi^-} \vdash h^-(\vec{x}) : \mathsf{Id}(f_{n-1}(\mathsf{src}\,\vec{r}(\vec{x})), g_{n-1}(\mathsf{tgt}\,\vec{r}(\vec{x})))}{\vec{x} : \Gamma_\pi \vdash \mathsf{J}_(h^-;\, x^{k-1}_{s(c)}, x^{k-1}_{t(c)}, x^k_c) : \mathsf{Id}(f_{n-1}(\mathsf{src}\,\vec{x}), g_{n-1}(\mathsf{tgt}\,\vec{x}))}$$

to be, up to reordering, the main hypothesis of an instance of Id-ELIM. So to derive the desired filler h, it is enough to derive h^- with

$$\vec{x} : \Gamma_{\pi^-} \vdash h^-(\vec{x}) : \mathsf{Id}(f_{n-1}(\mathsf{src}\,\vec{r}(\vec{x})), g_{n-1}(\mathsf{tgt}\,\vec{r}(\vec{x}))).$$

But now note that

$$d^{n-i}(\pi^-) = \begin{cases} d^{n-i}(\pi) & \text{for } n-i < k \\ (d^{n-i}(\pi))^- & \text{for } n-i \geq k \end{cases} ;$$

moreover, we can construct context maps

$$\vec{r}^s_i, \vec{r}^t_i \colon \Gamma_{d^{n-i}(\pi^-)} \longrightarrow \Gamma_{d^{n-i}(\pi)}$$

(analogous to $\vec{r}$ if $i \geq k$, and just the identity otherwise), and these commute with the maps src and tgt. So for each $i < n$, we have

$$\vec{x} : \Gamma_{d^{n-i}(\pi^-)} \vdash f_i(\vec{r}(\vec{x})) : \mathsf{Id}(f_{n-1}(\vec{r}(\mathsf{src}\,\vec{x})), g_{i-1}(\vec{r}(\mathsf{tgt}\,\vec{x}))),$$

$$\vec{x} : \Gamma_{d^{n-i}(\pi^-)} \vdash g_i(\vec{r}(\vec{x})) : \mathsf{Id}(f_{n-1}(\vec{r}(\mathsf{src}\,\vec{x})), g_{i-1}(\vec{r}(\mathsf{tgt}\,\vec{x}))),$$

i.e. the sequence of terms $(\vec{r}^*(f_i), \vec{r}^*(g_i))_{i<n}$ are a parallel pair for π^-. So by induction (since π^- has fewer cells than π), these terms have a filler; but this filler is exactly the desired term h^-.

Thus it is enough to show the existence of fillers in the case where π consists of just a single 0-cell, i.e. $\pi = \bullet$. But in this case, $\Gamma_\pi = \Gamma_{d^i(\pi)} = \Gamma_{d^i(\pi)} = (x:X)$ for each $i < n$, and so by the initiality of $(x:X)$ we must have $f_i(x) = g_i(x) = r^i(x)$ for each i; so now $h := r^n(x)$ gives the filler, and we are done. $\square$

Unwinding this induction, we can see that it exactly formalises the process described in 3.1.6, of repeatedly plugging in higher reflexivity terms for all variables, knowing that the given composites will themselves eventually compute down to higher reflexivity terms.

Proof 2. According to the categorical description of an endomorphism operad, we need to show that for any pasting diagram π, given all components $(\vec{f_i}, \vec{g_i})_{i<n}$ of a map of spans (as in Fig. 3.1.A of 3.1.3) apart from the apex, we can find some map $\vec{h}$ to complete it.

For this it is enough to show that we can find a top edge for the square

$$\begin{tikzcd} \Gamma_\pi \arrow[r, dotted] \arrow[d] & \underline{X}_n \arrow[d] \\ \Gamma_{\partial\pi} \arrow[r, "{[\vec{f_i}, \vec{g_i}]}"] & \Gamma_{\partial(n)} \end{tikzcd}$$

or, more generally, that we can complete any triangle of the form

$$\begin{tikzcd} & \underline{X}_n \arrow[d] \\ \Gamma_\pi \arrow[ur, dotted] \arrow[r] & \Gamma_{\partial(n)} \end{tikzcd} .$$

This we show by induction on π. In the case that π is the trivial diagram, then, we are done by the initiality of $\Gamma_\pi = (x:X)$.

Otherwise, consider $\Gamma_\pi \twoheadrightarrow \Gamma_{\pi^-}$. By the descriptions of 2.4.2, this is a dependent projection, projecting away a term of identity type and its target, so by the one-ended form of Id-ELIM[1], it has a retraction r with an

[1]A slightly more careful argument would of course show, as we saw in the syntactic version, that in fact the two-ended form also applies.

elim-structure. So now by induction, we can complete the triangle

$$\begin{tikzcd} & & \underline{X}_n \arrow[d, two heads] \\ \Gamma_{\pi^-} \arrow[r, tail, "r"] \arrow[urr, dotted] & \Gamma_\pi \arrow[r] & \Gamma_{\partial(n)} \end{tikzcd} ;$$

and then, seeing the result as a square

$$\begin{tikzcd} \Gamma_{\pi^-} \arrow[r] \arrow[d, tail, "r"'] & \underline{X}_n \arrow[d, two heads] \\ \Gamma_\pi \arrow[r] \arrow[ur, dotted] & \Gamma_{\partial(n)} \end{tikzcd} ,$$

the elim-structure on r gives us a filler, completing the original triangle as desired. □

Remarks 3.3.2. In each proof, Lemma 3.2.1 was applied only at the base case of the induction, and only to show that terms $x : X \vdash f_i(\vec{r}(x)), g_i(\vec{r}(x)) : \mathsf{Id}(r^n(x), r^n(x))$ must be equal to $r^{n+1}(x)$. Two alternative approaches could have been taken at this point. Firstly, we could have instead applied a normalisation/canonicity result for the theory to deduce that $f_i(x) = g_i(x) = r^i(x)$. This approach is taken in the proof of 4.3.1. Secondly, in theories for which normalisation/canonicity and the co-slice construction fail, or alternatively working directly over an arbitrary type (rather than the generic one) the full endomorphism operad may no longer be contractible; but we may still find a contractible sub-operad, by restricting to just those operations for which the required commutativity condition holds. This is the approach taken in [**GvdB10b**]; we use a variation of it for 4.3.5 below.

Also, besides just pulling back Γ_π to Γ_{π^-} in the categorical proof, we could have kept track of the boundaries as well

$$\begin{tikzcd} \Gamma_{\pi^-} \arrow[r, two heads] \arrow[d] & \Gamma_\pi \arrow[d] \\ \Gamma_{\partial\pi^-} \arrow[r] & \Gamma_{\partial\pi} \end{tikzcd}$$

or even of the whole spans, as we did in the syntactic version. While we have seen that these are not necessary here, this sort of thing can become useful in cases where we do not have enough left maps to continue until π is fully contracted; again, cf. the proof of 4.3.5 below.

An alternative point gloss on the categorical proof is given by the approach of [**GvdB10b**]: we are constructing by induction an elim-structure on the maps $(x\!:\!X) \longrightarrow \Gamma_\pi$.

3.4. Types as weak ω-categories

Putting the above results together, we obtain our main goal:

Theorem 3.4.1. *Let* $\mathsf{T} \in \mathbf{DTT}_{\mathsf{Id}}$ *be any type theory with identity types,* A *a closed type of* T*. Then the globular context* $\underline{A}_\bullet$ *carries the structure of an internal* P_{Id}*-algebra in* T*, and hence of an internal weak* ω*-category.*

Proof. By the universal property of $\mathsf{T}_{\mathsf{Id}}[X]$, there is a unique translation $F_A \colon \mathsf{T}_{\mathsf{Id}}[X] \longrightarrow \mathsf{T}$ taking X to A, and hence taking $\underline{X}_\bullet$ to $\underline{A}_\bullet$. Now by the functoriality of End (2.3.14), this induces an action of P_{Id} on $\underline{A}_\bullet$, and so, since by Theorem 3.3.1 P_{Id} admits a contraction, an action of L (the initial operad-with-contraction) on $\underline{A}_\bullet$, as desired. □

Corollary 3.4.2. *Let* T*,* A *be as above,* Δ *some (closed) context of* T*. Then the globular set of terms of types* A*,* Id_A*,* $\mathsf{Id}_{\mathsf{Id}_A}$*, …* *in context* Δ *carries the structure of a* P_{Id}*-algebra, and hence of a weak* ω*-category.*

When $\Delta = \diamond$*, this gives the* fundamental weak ω-category *of* A *as advertised in the introduction.*

Proof. This is just the globular hom-set $\mathsf{T}(\Delta, \underline{A}_\bullet)$, so by 2.3.16, it inherits a P_{Id}-action, and hence an L-action, from the actions on $\underline{A}_\bullet$. □

Using the "types to contexts" endofunctor $(-)^{\mathrm{Cxt}}/\!/\diamond$ of 1.3.3, we can extend this:

Corollary 3.4.3. *Let Θ, Δ be contexts of some theory $\mathsf{T} \in \mathbf{DTT}_{\mathsf{Id}}$. Then the globular set of terms of context maps from Δ into Θ and its higher identity contexts carries a natural weak ω-category structure.*

Proof. By applying the previous corollary to Θ, considered as a closed type of $\mathsf{T}^{\mathrm{Cxt}}/\!/\diamond$. $\square$

Similarly, we may extend these to dependent types A or contexts Θ over a base context Γ, by viewing them as closed types/contexts in the slice theory $\mathsf{T}/\!/\Gamma$.

Remark 3.4.4 (Functoriality)**.** The construction of the P_{Id}-algebra $\mathsf{T}(\Delta, \underline{A}_\bullet)$ is moreover covariantly functorial in T, and contravariantly in Δ. That is, translations $\mathsf{T} \longrightarrow \mathsf{T}'$ and context maps $\Delta' \longrightarrow \Delta$ each induce *strict* maps of P_{Id}-algebras, composing appropriately.[2] However, both of these require somewhat more technical machinery to prove than we have developed here. The functoriality in T is perhaps most clearly seen by considering the monoidal globular category $\mathbf{FibSpans}(\mathsf{T})$ introduced in [**GvdB10b**], of spans in T consisting entirely of dependent projections; then the algebras we have considered live within these monoidal categories, and a translation of theories induces a monoidal globular functor between them. Both these functoriality results also require a treatment of maps of operad algebras in the endomorphism-operad presentation, and hence of some slightly more involved globular structures than the operads we have considered so far.

More subtly, $\mathsf{T}(\Delta, \underline{A}_\bullet)$ should be functorial in A, but only to *weak* maps: a map of types $A \longrightarrow A'$ should induce *weak* maps of P_{Id}- or L-algebras—that is, weak ω-functors. This seems an altogether trickier question, due partly but not only to the lack, until fairly recently ([**Gar08**]), of a suitable definition of weak ω-functor. However, as this theory becomes better developed,

[2] For precise statements, one has to be slightly careful due to the dependence of typing of Δ over T. Concisely, the P_{Id} algebra $\mathsf{T}(\Delta, \underline{A}_\bullet)$ is functorial over the total category $\int_{\mathsf{T} \in \mathbf{DTT}_{\mathsf{Id}}} \mathrm{Cxt}(\mathsf{T})$ of the fibration of contexts.

the weak functoriality of $\mathsf{T}(\Delta, \underline{A}_{\bullet})$ in maps $f\colon A \longrightarrow A'$ should become a corollary of the results of the next chapter: within the classifying ω-category of the theory, this is just a whiskering operation between hom-ω-categories and as such, should presumably be a weak functor. Unfortunately, to the author's present knowledge, this whiskering statement (and similar results) do not yet appear in the literature; the theory of weak maps of operadic ω-categories is as yet at a very early stage of development.

CHAPTER 4

The classifying weak ω-category of a theory

In this chapter we will construct classifying weak ω-categories for theories with various extensionality rules, and discuss how it might be possible to extend this to require only the Id-types; we also give a variant which works for theories with just Id- and Π-types.

The construction of the classifying weak ω-category is closely analogous to that of the fundamental weak ω-groupoid of a space: it is obtained by homming out of a complex of representing objects ("globes"), and so it is enough to show that these representing objects form a co-(weak ω-category), just as the topological globes $D^0 \rightrightarrows D^1 \rightrightarrows D^2 \rightrightarrows \cdots$ do in **Top**.

To this end, in Section 4.1 we introduce our representing objects, the *type-theoretic globes* $\mathsf{G}_\bullet$. We then develop, in Section 4.2, classes of left and right maps on **DTT** (*term extensions* vs. *term-contractible maps*), and isolate a property $\overline{J}$ (in syntactic terms, a conservativity principle), ensuring that the maps $\mathsf{T}_\Phi[\widehat{\pi}] \longrightarrow \mathsf{T}_\Phi[\widehat{\pi^-}]$ are term-contractible. (This thus fulfils a function dual to that of the elim-structure on $\Gamma_{\pi^-} \rightarrowtail \Gamma_\pi$ in the preceding section.) Finally, in Section 4.3, we put the pieces together, with arguments extending those of 3.3.1 and its corollaries, to show that various operads are contractible, and deduce the desired weak ω-category structures.

4.1. Globular structures from DTT

The type-theoretic globes.

4.1.1. Once again, let Φ be some set of rules/constructors, including at least the Id-rules. The *type-theoretic globes* over Φ are then a sequence of theories G_n^Φ which play a similar rôle in $\mathbf{DTT}_\Phi$ to that which the discs D^n play in

Top: they are an internal weak-ω-cocategory, and as such will—almost—be representing objects for the classifying weak ω-category functor.

The idea is that G^Φ_0 should be the free theory on a singler closed type C; then G^Φ_1, the free theory on two types S, T and a map c_1 between them; G^Φ_2, free on two types S, T, two maps s_1, t_1 between them, and a term c_2 of propositional identity between these; and so on. Precisely:

Definition 4.1.2. G^Φ_n is the theory over Φ generated by axioms i-src, i-tgt (for $0 \leq i < n$), and n-cell, as follows:

$$\frac{}{\diamond \vdash \mathsf{S}\ \mathsf{type}}\ \text{0-src} \qquad \frac{}{\diamond \vdash \mathsf{T}\ \mathsf{type}}\ \text{0-tgt} \qquad \frac{}{\diamond \vdash \mathsf{C}\ \mathsf{type}}\ \text{0-cell}$$

$$\frac{}{x : \mathsf{S} \vdash \mathsf{s}_1(x) : \mathsf{T}}\ \text{1-src} \qquad \frac{}{x : \mathsf{S} \vdash \mathsf{t}_1(x) : \mathsf{T}}\ \text{1-tgt} \qquad \frac{}{x : \mathsf{S} \vdash \mathsf{c}_1(x) : \mathsf{T}}\ \text{1-cell}$$

$$\frac{}{\Gamma \vdash \mathsf{s}_i(x) : \mathsf{Id}(\mathsf{s}_{i-1}(x), \mathsf{t}_{i-1}(x))}\ i\text{-src} \qquad (i \geq 2)$$

and i-tgt , i-cell exactly as i-src except defining term-formers t_i, c_i in place of s_i.

$$\begin{tikzcd} \overset{\mathsf{C}}{\bullet} \end{tikzcd} \qquad \begin{tikzcd} \overset{\mathsf{S}}{\bullet} \arrow[r, "\mathsf{c}_1"] & \overset{\mathsf{T}}{\bullet} \end{tikzcd} \qquad \begin{tikzcd} \overset{\mathsf{S}}{\bullet} \arrow[r, bend left, "\mathsf{s}_1"] \arrow[r, bend right, "\mathsf{t}_1"'] \arrow[r, phantom, "\Downarrow \mathsf{c}_2"] & \overset{\mathsf{T}}{\bullet} \end{tikzcd}$$

G_0 G_1 G_2

(Often, when working with some fixed Φ, we write just G_n for G^Φ_n.)

4.1.3. There are evident co-source and co-target interpretations between these theories (sending for instance c_i to s_i or t_i), and moreover co-unit interpretations (sending c_{i+1} to $r(\mathsf{c}_i)$), forming a reflexive coglobular object $\mathsf{G}_\bullet$ in $\mathbf{DTT}_\Phi$:

$$\begin{tikzcd} \mathsf{G}_0 \arrow[r, shift left=2] \arrow[r, shift right=2] & \mathsf{G}_1 \arrow[l] \arrow[r, shift left=2] \arrow[r, shift right=2] & \mathsf{G}_2 \arrow[l] \arrow[r, shift left=2] \arrow[r, shift right=2] & \cdots \arrow[l] \end{tikzcd}$$

Leaving aside the reflexivity for now, we can thus see the globes as a functor

$$\mathsf{G}_\bullet \colon \mathbb{G} \longrightarrow \mathbf{DTT}_\Phi.$$

Since $\mathbf{DTT}_\Phi$ is co-complete, this induces by general nonsense ([**MLM94**, VII.2]) an adjoint pair of functors between $\widehat{\mathbb{G}}$ and $\mathbf{DTT}_\Phi$ (a "Kan situation"). Both these functors will be of central interest to us in the sequel:

$$\begin{tikzcd} \widehat{\mathbb{G}} \arrow[rr, bend right, "{\mathsf{T}_\Phi[-] := \mathrm{Lan}_{\mathbf{y}} \mathsf{G}_\bullet}"'] \arrow[rr, phantom, "\top"] & & \mathbf{DTT}_\Phi \arrow[ll, bend right, "{\mathcal{C}\ell_\omega^- := \mathbf{DTT}_\Phi(\mathsf{G}_\bullet, -)}"'] \\ \mathbb{G} \arrow[u, "\mathbf{y}"] \arrow[urr, bend right, "\mathsf{G}_\bullet"'] & & \end{tikzcd}$$

The right adjoint, $\mathcal{C}\ell_\omega^- \colon \mathbf{DTT}_\Phi \longrightarrow \widehat{\mathbb{G}}$, is defined by homming out of the globes, i.e. by setting $\mathcal{C}\ell_\omega^-(\mathsf{T})_n = \mathbf{DTT}_\Phi(\mathsf{G}_n, \mathsf{T})$. Thus, by the definitions of the globes, the 0-cells of $\mathcal{C}\ell_\omega^-(\mathsf{T})$ correspond exactly to closed types in T; the 1-cells $A \longrightarrow B$ to terms of A dependent on a single variable from B; the 2-cells to terms of type Id_B between 1-cells; and so on.

This is very nearly, but not quite, what we wanted for the underlying globular set of $\mathcal{C}\ell_\omega(\mathsf{T})$. The difference is that it has only the *types* of T as 0-cells, not all the contexts; however, we will proceed for now with $\mathcal{C}\ell_\omega^-$, and remedy this deficiency later.

Meanwhile, the left adjoint $\mathsf{T}_\Phi[-] \colon \widehat{\mathbb{G}} \longrightarrow \mathbf{DTT}_\Phi$ is constructed as the left Kan extension of $\mathsf{G}_\bullet$ along $\mathbf{y}$, and may be seen as freely adjoining a globular set to T_Φ, using the globes as templates. Explicitly, for a globular set $X_\bullet$, the theory $\mathsf{T}[X_\bullet]$ has an axiom for each cell of $X_\bullet$, realising the 0-cells as closed types, the 1-cells as terms between these types, and the higher-cells as terms of appropriate identity types.[1]

[1] A related construction is considered in [**AHW09**] and [**WH10**], corresponding to a slightly different co-globular theory: they omit our G_0, giving instead just a single closed base type, and realise 0-cells as closed terms of this type, 1-cells as terms of identity type between these, and so forth. Their $T_{\mathbf{ML}}$ is then the monad induced by the Kan adjunction.

In particular, $\mathsf{T}_\Phi[\mathbf{y}(n)] = \mathsf{G}_n$, and the theories $\mathsf{T}_\Phi[\widehat{\pi}]$ give the diagram objects of $\mathsf{G}_\bullet$. Also useful will be the boundary of the n-globe, $\partial\mathsf{G}_n := \mathsf{T}_\Phi[\partial(n)]$; up to isomorphism, this is the theory given by i-src and i-tgt, for $0 \leq i < n$, i.e. all the axioms of G_n except for n-cell itself.

4.1.4. Since $\mathbf{DTT}_\Phi$ is co-complete, we can consider (by 2.3.14) the co-endomorphism operad of the globes, $\mathrm{End}_{\mathbf{Spans}(\mathbf{DTT}_\Phi^{\mathrm{op}})}(\mathsf{G}_\bullet)$, or briefly just $\mathrm{End}(\mathsf{G}_\bullet)$. We know that its operations of some shape π are given by

$$\mathrm{End}(\mathsf{G}_\bullet)(\pi) \cong \mathbf{Spans}(\mathbf{DTT}_\Phi^{\mathrm{op}})_n(\mathsf{G}_n, \mathsf{T}_\Phi[\hat{\pi}])$$

and hence, unwinding this formula, consist of diagrams as in Fig. 4.1.A below.

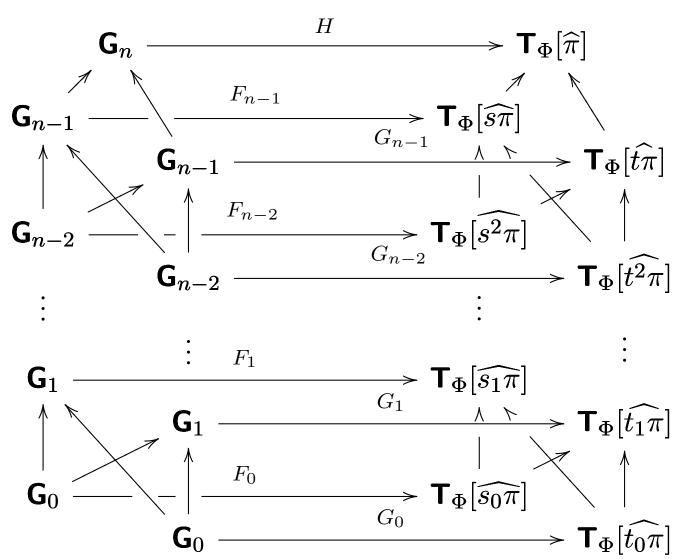


FIGURE 4.1.A. An operation in an endomorphism operad

4.1.5. For example, a composition law for $\begin{tikzcd} \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow"] & \bullet \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "\Downarrow"] & \bullet \end{tikzcd}$ is given by the map of spans

$$\begin{tikzcd}
& \mathsf{G}_2 \arrow[rrr] & & & \mathsf{T}_\Phi\left[\begin{tikzcd} P \arrow[r, bend left, "f"] \arrow[r, bend right, "f'"'] \arrow[r, phantom, "\Downarrow\alpha"] & Q \arrow[r, bend left, "g"] \arrow[r, bend right, "g'"'] \arrow[r, phantom, "\Downarrow\beta"] & R \end{tikzcd} \right] \\
\mathsf{G}_1 \arrow[ur] \arrow[rrr, "{\mathsf{S} \mapsto P,\ \mathsf{T} \mapsto R,\ \mathsf{c}_1(x) \mapsto g(f(x)) : R}"] & & & \mathsf{T}_\Phi\left[P \xrightarrow{f} Q \xrightarrow{g} R\right] \arrow[ur] & \\
& \mathsf{G}_1 \arrow[uu] \arrow[rrr, "{\mathsf{S} \mapsto P,\ \mathsf{T} \mapsto R,\ \mathsf{c}_1(x) \mapsto g'(f'(x)) : R}"] & & & \mathsf{T}_\Phi\left[P \xrightarrow{f'} Q \xrightarrow{g'} R\right] \arrow[uu] \\
\mathsf{G}_0 \arrow[uu] \arrow[ur] \arrow[rrr, "\mathsf{C} \mapsto P"] & & & \mathsf{T}_\Phi[P] \arrow[uu] \arrow[ur] & \\
& \mathsf{G}_0 \arrow[uuul] \arrow[uu] \arrow[rrr, "\mathsf{C} \mapsto R"] & & & \mathsf{T}_\Phi[R] \arrow[uu] \arrow[uuul]
\end{tikzcd}$$

while the apex map interprets S as P, T as R, $\mathsf{s}_1(x)$ as $f(g(x))$, $\mathsf{t}_1(x)$ as $f'(g'(x))$ (all as forced by the lower dimensions of the span), and $\mathsf{c}_2(x)$ as the term

$$\mathsf{J}_{y,y':Q,\, u:\mathsf{Id}_Q(y,y').\ \mathsf{Id}(g(y),g'(y'))}(\,y.\beta(y);\ f(x),\ f'(x),\ \alpha(x))$$

of type $\mathsf{Id}_R(\,g(f(x))\,,\,g'(f'(x))\,)$. (The reader tempted to try actually reading this term is encouraged to re-derive it herself instead: this will certainly be more enlightening, and probably also easier.)

4.1.6. Since the cells involved in $\mathsf{T}[\pi]$ are generic, this operation can be implemented as a composition law in any type theory with Id-types. Generally, by 2.3.16, $\mathrm{End}(\mathsf{G}_\bullet)$ acts naturally on $\mathcal{C}\ell_\omega^-$, allowing us to lift $\mathcal{C}\ell_\omega^-$ to a functor into $\mathrm{End}(\mathsf{G}_\bullet)$-**Alg** which by abuse of notation we again denote $\mathcal{C}\ell_\omega^-$:

$$\begin{tikzcd}
& \mathrm{End}(\mathsf{G}_\bullet)\text{-}\mathbf{Alg} \arrow[d, "U"] \\
\mathbf{DTT}_\Phi \arrow[ur, "\mathcal{C}\ell_\omega^-"] \arrow[r, "\mathcal{C}\ell_\omega^-"] & \widehat{\mathbb{G}}
\end{tikzcd}$$

Moreover, since U reflects and the original $\mathcal{C}\ell_\omega^-$ preserves all limits, so does the lifted $\mathcal{C}\ell_\omega^-$, and it is easily seen to be finitary; so by the adjoint functor theorem for locally presentable categories [**AR94**, 1.66], $\mathcal{C}\ell_\omega^-$ has a left adjoint, realising any $\mathrm{End}(\mathsf{G}_\bullet)$-algebra as a theory. (Its cells are realised as types and terms as under $\mathsf{T}_\Phi[-]$, and the $\mathrm{End}(\mathsf{G}_\bullet)$-action specifies various definitional-equality axioms between them.)

4.1.7. As mentioned before, this is not quite what we want; $\mathcal{C}\ell_\omega^-(\mathsf{T})$ has only types as objects, where we would like contexts. To remedy the situation, we can compose with the "types to contexts" endofunctor $(-)^{\mathsf{cxt}}/\!/\diamond$ on $\mathbf{DTT}_\Phi$, and define $\mathcal{C}\ell(\mathsf{T}) := \mathcal{C}\ell_\omega^-(\mathsf{T}^{\mathsf{cxt}}/\!/\diamond)$.

Now the objects of $\mathcal{C}\ell(\mathsf{T})$ are closed types of $\mathsf{T}^{\mathsf{cxt}}/\!/\diamond$, i.e. closed contexts of T, just as we wanted; and the fullness of the functor $\mathsf{T}^{\mathsf{cxt}}/\!/\diamond \longrightarrow \mathsf{T}$ ensures that higher cells also are as we intended.

This adjustment is not quite so ad hoc as it may appear: if $(-)^{\mathsf{cxt}}slice\diamond$ were indeed a monad (as it very nearly is), then $\mathcal{C}\ell_\omega$ would itself be a representable functor, on the Kleisli category of $(-)^{\mathsf{cxt}}slice\diamond$.

4.1.8. In Section 4.3 below, we will investigate the question of when $\mathrm{End}(\mathsf{G}_\bullet)$ is contractible, or at least of finding a contractible suboperad. This will require, however, the development of some more type-theoretic machinery. For now, we may content ourselves with showing that at least in dimensions ≤ 1, $\mathrm{End}(\mathsf{G}_\bullet)$ is very nice.

Specifically, recall from 2.3.9 the discrete/truncation/indiscrete adjunctions

$$1\text{-}\mathbf{Opd} \underset{I}{\overset{D}{\rightleftarrows}} \omega\text{-}\mathbf{Opd} \quad (\text{with } \mathrm{tr}^1 \colon \omega\text{-}\mathbf{Opd} \to 1\text{-}\mathbf{Opd})$$

.

Now take $P_{\mathbf{Cat}} \in 1\text{-}\mathbf{Opd}$ to be the operad for categories, i.e. the terminal 1-operad. Then there is a map $\psi \colon P_{\mathbf{Cat}} \longrightarrow \mathrm{tr}^1\mathrm{End}(\mathsf{G}_\bullet)$ (or equivalently $D(P_{\mathbf{Cat}}) \longrightarrow \mathrm{End}(\mathsf{G}_\bullet)$), so that truncating algebras and pulling back along this map induces a map $\mathrm{End}(\mathsf{G}_\bullet)\text{-}\mathbf{Alg} \longrightarrow \mathbf{Cat}$, which when applied to

$\mathcal{C}\ell_\omega(\mathbf{T})$ recovers the classifying category $\mathcal{C}\ell(\mathbf{T})$:

$$\begin{tikzcd}
 & \mathrm{End}(\mathbf{G}_\bullet)\text{-}\mathbf{Alg} \arrow[r,"\mathrm{tr}^1"] \arrow[dr] & \mathrm{tr}^1\mathrm{End}(\mathbf{G}_\bullet)\text{-}\mathbf{Alg} \arrow[r,"\psi^*"] \arrow[dr] & \mathbf{Cat} \arrow[d] \\
\mathbf{DTT}_\Phi \arrow[ur,"\mathcal{C}\ell_\omega"] \arrow[urrr,"\mathcal{C}\ell"] \arrow[rr,"\mathcal{C}\ell_\omega"] & & \widehat{\mathbb{G}} \arrow[r,"\mathrm{tr}^1"] & \widehat{\mathbb{G}}_1
\end{tikzcd}$$

This points us towards an easy abstract construction of the map ψ. We know that the 1-globular object $\mathbf{G}_0 \leftleftarrows \mathbf{G}_1$ represents the original classifying category functor $\mathcal{C}\ell : \mathbf{DTT}_\Phi \longrightarrow \mathbf{Cat}$; so by the Yoneda lemma, $\mathbf{G}_0 \leftleftarrows \mathbf{G}_1$ must carry some co-category structure; but such a structure corresponds exactly to an operad map of the form we want, since $\mathrm{tr}^1\mathrm{End}(\mathbf{G}_\bullet) \cong \mathrm{End}(\mathrm{tr}^1\mathbf{G}_\bullet)$.

4.1.9. However, constructing ψ concretely gives us an excuse to analyse low dimensions of $\mathrm{End}(\mathbf{G}_\bullet)$. A 0-dimensional operation in $\mathrm{End}(\mathbf{G}_\bullet)$ is perforce just a unary map $\mathbf{G}_0 \longrightarrow \mathbf{G}_0$ (there is only one 0-dimensional pasting diagram); so the single 0-dimensional operation in **Cat** (the identity on 0-cells) we send to the identity map $1_{\mathbf{G}_0}$. (There is no freedom here: ψ must preserve the operad structure, and $1_{\mathbf{G}_0}$ is the 0-dimensional operad unit of $\mathrm{End}(\mathbf{G}_\bullet)$.)

A 1-dimensional pasting diagram is just a path $path_l = (\cdot \longrightarrow \cdot \longrightarrow \ldots \longrightarrow \cdot)$ of some length $l \geq 0$. An operation of shape $path_l$ in $\mathrm{End}(\mathbf{G}_\bullet)$ with source and target $1_{\mathbf{G}_0}$ is a map of cospans

$$\begin{tikzcd}
 & \mathbf{G}_1 \arrow[rr,"H"] & & \mathbf{T}_\Phi[\widehat{path_l}] \\
\mathbf{G}_0 \arrow[ur,"s"] \arrow[rr,equal] & & \mathbf{G}_0 \arrow[ur,"s"] & \\
 & \mathbf{G}_0 \arrow[uu,"t"'] \arrow[rr,equal] & & \mathbf{G}_0 \arrow[uu,"t"']
\end{tikzcd}$$

But $\mathbf{T}_\Phi(path_l)$ admits a very simple axiomatisation

$$\frac{}{\diamond \ \vdash \ A_i \ \mathsf{type}} \ (0 \leq i \leq l) \qquad\qquad \frac{}{\vec{x} : A_{j-1} \ \vdash \ f_j(a) : A_j} \ (1 \leq j \leq l)$$

simply adjoining basic types and type formers

$$A_0 \xrightarrow{f_1} A_1 \xrightarrow{f_2} \quad \cdots \quad \xrightarrow{f_l} A_l \ .$$

The source and target maps of the right-hand cospan interpret the type of $\mathbf{G}_0$ as A_0 and A_l respectively; so suitable maps $H : \mathbf{G}_1 \longrightarrow \mathbf{T}_\Phi[\widehat{path_l}]$ correspond to interpretations of the type-constructor of $\mathbf{G}_1$ as some term

$$x : A_0 \ \vdash \ t(x) : A_l$$

For the unique l-ary operation of $P_{\mathbf{Cat}}$, we thus use the obvious composite term $f_l(f_{l-1}(\ldots(f_1(x))\ldots))$. It is routine to check that this indeed gives an operad map.

4.1.10. When Φ gives a particularly well-behaved type system, we can say a little more.

In the case when Φ consists of just the Id-rules, then firstly $\mathbf{G}_0$ has no other closed types besides the basic C, so $1_{\mathbf{G}_0}$ is its only endomorphism, and the only element of $\mathrm{End}(\mathbf{G}_\bullet)$ in dimension 0; and secondly, $\mathbf{T}_\Phi[\widehat{path_l}]$ enjoys both normalisation and *canonicity*[2], so the "obvious term" we used was in fact the only possible such term: there is only one l-ary operation in $\mathrm{End}(\mathbf{G}_\bullet)$ with source and target $1_{\mathbf{G}_n 0}$.

So in this case, the map $P_{\mathbf{Cat}} \longrightarrow \mathrm{tr}^1\mathrm{End}(\mathbf{G}_\bullet)$ (always injective, since $P_{\mathbf{Cat}}$ is terminal) is moreover surjective, and gives an isomorphism $\mathrm{tr}^1\mathrm{End}(\mathbf{G}_\bullet) \cong P_{\mathbf{Cat}}$.

In richer type systems, $\mathbf{G}_0$ will typically have more closed types (e.g. $C \to C$), and hence $\mathrm{End}(\mathbf{G}_0)$ will have more 0-dimensional operations. But in important cases, such as when Φ consists of just the Id- and Π-rules, we retain normalisation and canonicity for $\mathbf{T}_\Phi[\widehat{path_l}]$, so by the argument above our map is at least an isomorphism from $P_{\mathbf{Cat}}$ to the *normalised core* of $\mathrm{End}(\mathbf{G}_\bullet)$.

A variant for Π-types.

4.1.11. We can also consider a variant set of globes ${}^\Pi\mathbf{G}^\Phi_\bullet$, for any set of constructors Φ including Π-types. The axioms for each ${}^\Pi\mathbf{G}_n$ are selected

[2]canonicity: the property that every closed normal form is an intro (aka canonical) form; there are no stuck (aka neutral) normal forms

from rules i-src-Π, i-tgt-Π, i-cell-Π, analogously to the axioms for $\mathbf{G}_n$. These axioms differ from before in dimensions ≥ 1, by using closed rather than open terms:

$$\frac{}{\diamond \vdash s_1 : \Pi_{x:S_0} T_0}\ \textsf{1-src-}\Pi \qquad \frac{}{\diamond \vdash s_i : \mathsf{Id}_{\Pi_{x:S_0} T_0} \mathsf{Id}(s_{i-1}, t_{i-1})}\ i\textsf{-src-}\Pi \qquad \text{etc.}$$

As before, we get a Kan adjunction

$$\widehat{\mathbb{G}} \underset{{}^{\Pi}\mathcal{C}\ell_\omega^-}{\overset{\mathbf{T}_\Phi[-]^\Pi}{\rightleftarrows}} \mathbf{DTT}_\Phi$$

and lift ${}^{\Pi}\mathcal{C}\ell_\omega^-$ to $\mathrm{End}({}^{\Pi}\mathbf{G}_\bullet)$-$\mathbf{Alg}$; also as before, we "correct" the functor ${}^{\Pi}\mathcal{C}\ell_\omega^-$ by precomposing with $(-)^{\mathsf{cxt}}/\!/\diamond$, to get an alternate candidate for the classifying weak ω-category:

$${}^{\Pi}\mathcal{C}\ell_\omega \colon \mathbf{DTT}_\Phi \longrightarrow \mathrm{End}({}^{\Pi}\mathbf{G}_\bullet)\text{-}\mathbf{Alg}$$

The objects of ${}^{\Pi}\mathcal{C}\ell_\omega(\mathbf{T})$ are the same as those of $\mathcal{C}\ell_\omega(\mathbf{T})$. The difference is in the higher cells: rather than open context maps, 1-cells are now context maps from the empty context $\diamond$ into "Π-contexts", and higher cells are context maps from $\diamond$ into the identity contexts over these.

Thus, while not exactly what we first thought of, this is a reasonable alternative candidate for the "classifying weak ω-category".

4.1.12. There is an evident map of globular objects $\mathbf{G}_\bullet \longrightarrow {}^{\Pi}\mathbf{G}_\bullet$, interpreting the open term-constructors of $\mathbf{G}_\bullet$ by applications of the closed terms of Π-type in ${}^{\Pi}\mathbf{G}_\bullet$. This induces a natural map $\mathcal{C}\ell_\omega^{\Pi}(\mathbf{T}) \longrightarrow \mathcal{C}\ell_\omega(\mathbf{T})$.

In theories with Π-EXT, there is also a map ${}^{\Pi}\mathbf{G}_\bullet \longrightarrow \mathbf{G}_\bullet$ coming the other way. In the presence of Π-EXT-APP-DEF, these maps exhibit ${}^{\Pi}\mathbf{G}_\bullet$ as a retraction of $\mathbf{G}_\bullet$, and hence ${}^{\Pi}\mathcal{C}\ell_\omega$ as a retract of $\mathcal{C}\ell_\omega$.

4.1.13. We would also like to repeat the construction of 4.1.8 and construct a map

$${}^{\Pi}\psi \colon P_{\mathbf{Cat}} \longrightarrow \mathrm{tr}^1\mathrm{End}({}^{\Pi}\mathbf{G}_\bullet),$$

and indeed we can do so, under the further assumption that Φ also contains the (definitional) η-rule for Π-types. In this case, we interpret the l-ary operation using the map ${}^{\Pi}\mathsf{G}_1 \longrightarrow \mathsf{T}_\Phi[\widehat{path_l}]$ which interprets the basic term $c_1 : S_0 \to T_0$ as the composite term $\lambda x : S_0.\ (c_l \cdot (c_{l-1} \cdot \ldots (c_1 \cdot x) \ldots))$. This certainly preserves the operad composition; the η-rule required to ensure that it preserves the operad unit, i.e. that in the case $l = 1$, the resulting operation (sending c_1 to $\lambda x.\, c_1 \cdot x$) is just the identity on ${}^{\Pi}\mathsf{G}_1$.

In the case where Φ is exactly $(\mathsf{Id}, \Pi, \Pi\text{-}\eta)$, then both normalisation and canonicity hold , but C_0 is not the only closed type of ${}^{\Pi}\mathsf{G}_0$. So ${}^{\Pi}\psi\colon P_{\mathbf{Cat}} \longrightarrow \mathrm{tr}^1\mathrm{End}({}^{\Pi}\mathsf{G}_\bullet)$ is not an isomorphism in this case, as it is not surjective on 0-cells; but it is at least full on 1-cells.

4.2. Homotopical structures on DTT

Left and right maps in DTT.

4.2.1. In this section, we will set up various classes of left and right maps on $\mathbf{CwA}^{\Phi}_{\diamond}$, with a view to applying the methods of 3.3.1 to find a contractible operad acting on the globes.

As remarked in the introduction to this chapter, we will succeed only for theories with Π-types and some extensionality rules. However, only one step of the proof (showing that the reflexivity maps $\mathsf{T}_\Phi[\widehat{\pi}] \longrightarrow \mathsf{T}_\Phi[\widehat{\pi^-}]$ are right maps) uses these extra rules; and it seems hopeful that a similar approach could also apply for theories with only Id-types.

We therefore isolate and investigate a certain property of type systems, $\overline{J}$, which suffices for the proof of this step, and which seems to be of independent interest. We discuss equivalent statements of $\overline{J}$ from several rather different points of view: as a conservativity statement for certain theory extensions; as a second-order form of the Id-elim rule; and as a form of observational equality for Π-types. $\overline{J}$ turns out to be derivable in theories with the Π-EXT-APP rules; it seems plausible, but elusive, that it is admissible for theories with just Id-types; and it may fail over intermediate theories.

4.2.2 (Type and term extensions)**.** For the remainder of this section, fix some collection Φ of the constructors and rules of 1.1.4, and work in $\mathbf{DTT}_\Phi$. (The main cases of interest in the sequel are where Φ is either (Id), (Id, Π, η), or $(\mathsf{Id}, \Pi, \Pi\text{-}\textsc{ext}, \Pi\text{-}\textsc{ext-app})$.)

For $n \geq 0$, we define theories $\mathsf{T}_\Phi[\Gamma_{(n)}]$, $\mathsf{T}_\Phi[\Gamma_{(n)} \vdash A]$, and $\mathsf{T}_\Phi[\Gamma_{(n)} \vdash a : A]$ to be the free theories on, respectively, a context of length n; a dependent type, in context of length n; and a term in such a type. Axiomatically, each may be specified by some subset of the rules below: $\mathsf{T}_\Phi[\Gamma_{(n)}]$ by the rules i-cxt, for $0 \leq i < n$; $\mathsf{T}_\Phi[\Gamma_{(n)} \vdash A]$, by these rules together with n-$\textsc{type}$; and $\mathsf{T}_\Phi[\Gamma_{(n)} \vdash a : A]$ by all of the above, together with n-$\textsc{term}$:

$$\frac{\Gamma \vdash a_0 : A_0 \ \ldots\ \Gamma \vdash a_{i-1} : A_{i-1}}{\Gamma \vdash A_i(a_0, \ldots, a_{i-1})\ \mathsf{type}}\ i\text{-}\mathsf{cxt} \qquad \frac{\Gamma \vdash a_0 : A_0 \ \ldots\ \Gamma \vdash a_{i-1} : A_{i-1}}{\Gamma \vdash A(a_0, \ldots, a_{i-1})\ \mathsf{type}}\ i\text{-}\textsc{type}$$

$$\frac{\Gamma \vdash a_0 : A_0 \ \ldots\ \Gamma \vdash a_{i-1} : A_{i-1}}{\Gamma \vdash a(a_0, \ldots, a_{i-1}) : A_i(a_0, \ldots, a_{i-1})}\ i\text{-}\textsc{term}$$

(Of course, we have $\mathsf{T}[\Gamma_{(n-1)} \vdash A] \cong \mathsf{T}[\Gamma_{(n)}]$; we retain the distinction just for notational clarity.)

4.2.3. The importance of these theories lies in their universal mapping properties. For any theory T, maps $\mathsf{T}_\Phi[\Gamma_{(l)}] \longrightarrow \mathsf{T}$ correspond precisely to contexts of length n in T; maps $\mathsf{T}_\Phi[\Gamma_{(l)} \vdash A\ \mathsf{type}] \longrightarrow \mathsf{T}$, to types over such a context; and maps $\mathsf{T}_\Phi[\Gamma_{(l)} \vdash a : A] \longrightarrow \mathsf{T}$ to terms of such a type.

An analogy can profitably be drawn here between type theories and higher categories. Globular higher categories are made up of cells, which are *represented* by the free n-categories on individual cells. Similarly, type theories are made up of judgements—contexts, types, and terms—which are represented by the theories above.

But now, many important aspects of higher category theory—in particular, their homotopical structure—can be described in terms of the inclusions of boundaries into those basic cells. Much of this carries over substantially to type theories once we observe that *judgements have boundaries too* — indeed, this idea is already implicit in referring to e.g. $\Gamma \vdash a : A$ as a *term judgement*: we are thinking of a as the essential substance of the judgement, and the function of Γ and A as just to situate a within its surroundings, as seen in the form of the algebraic rules of 1.1.8.

The "inclusions of boundaries into cells" are defined as follows:

Definition 4.2.4. The *universal type (resp. term) extensions* are the inclusion maps

$$i_n^{\mathrm{ty}} \colon \mathbf{T}_\Phi[\Gamma_{(n)}] \rightarrowtail \mathbf{T}_\Phi[\Gamma_{(n)} \vdash A],$$
$$i_n^{\mathrm{tm}} \colon \mathbf{T}_\Phi[\Gamma_{(n)} \vdash A] \rightarrowtail \mathbf{T}_\Phi[\Gamma_{(n)} \vdash a : A].$$

A *basic term/type extension* is a pushout of one of the universal extensions. A *term/type/term-and-type extension* is any composite (possibly transfinite) of basic extensions.

We indicate such extensions in diagrams by tailed arrows: $\mathbf{T} \rightarrowtail \mathbf{S}$.

So in syntactic terms, a basic term extension is just any extension of a theory $\mathbf{T}$ by a new term-constructor

$$x_1 : A_1,\ \ldots,\ x_{n-1} : A_{n-1}(\vec{x}^{<n-1})\ \vdash\ a(\vec{x}) : A_n(\vec{x}),$$

where the A_i (for $i = 1, \ldots n$) are existing types of the theory. Similarly, a basic type extension is an extension by a single algebraic type-forming axiom. A general term/ype/term-and-type extension is any extension of theories formed by iteratively adding (arbitrary sets of) axioms of these forms.

In the langage of Section 2.1, the classes of term, type, and term-and-type extensions are just classes of $\mathcal{J}$-cell complexes for evident suitable choices of $\mathcal{J}$. As such, they are immediately closed under composition, identities, and pushouts.

Definition 4.2.5. A *term-contraction* (resp. *type-contraction*, *contraction*) on a map $F \colon \mathbf{T} \longrightarrow \mathbf{S}$ is a $\mathcal{J}^{\boxslash}$-structure, where $\mathcal{J}$ is the set of universal term (resp. type, term and type) extensions. Explicitly, it is a function assigning a diagonal filler to every square

$$\begin{tikzcd}
\mathsf{T}_\Phi[\Gamma_{(n)} \vdash A \ \mathsf{type}] \arrow[r] \arrow[d, tail, "i_n^{\mathrm{tm}}"'] & \mathsf{T} \arrow[d, "F"] \\
\mathsf{T}_\Phi[\Gamma_{(n)} \vdash a : A] \arrow[r] \arrow[ur, dotted] & \mathsf{S}
\end{tikzcd}$$

with left-hand-side a universal type (term, term or type) extension.

A map admitting such structure is called *term-contractible* (*type-contractible*, *contractible*); assuming choice, this is equivalent to being weakly orthogonal to all universal term (type, term and type) extensions.

We will write $\mathcal{R}_{\mathrm{tm}}$, $\mathcal{R}_{\mathrm{ty}}$, R_{tmty} for the classes of contractible maps, and indicate them in diagrams by double-headed arrows: $\mathsf{T} \longrightarrow\!\!\triangleright \mathsf{S}$.

4.2.6. By 2.1.3, a type-contractible (term-contractible, contractible) map in fact has canonical liftings against all type (term, term-and-type) extensions; and the classes $\mathcal{R}_{\mathrm{tm}}$, $\mathcal{R}_{\mathrm{ty}}$, $\mathcal{R}_{\mathrm{ty}}$ are closed under identities, transfinite composition, pullbacks, and retracts.

Also, the universal extensions were axiomatised purely algebraically over the structural core, with no specific constructors required. In other words, for any Φ, the universal extensions in $\mathbf{DTT}_\Phi$ are the image under the left adjoint $F \colon \mathbf{DTT} \longrightarrow \mathbf{DTT}_\Phi$ of the universal extensions in $\mathbf{DTT}$.

It follows immediately that contractions and contractibility are preserved and reflected by the forgetful functor $U \colon \mathbf{DTT}_\Phi \longrightarrow \mathbf{DTT}$, right adjoint to F.

4.2.7. Contractibility is familiar in syntactic terms as a form of conservativity. Term-contractibility of a translation $F \colon \mathsf{T} \longrightarrow \mathsf{S}$, for instance, states that whenever we have a type $\Gamma \vdash_{\mathsf{T}} A \ \mathsf{type}$ of T whose interpretation in S is inhabited by some term $F(\Gamma) \vdash_{\mathsf{S}} a : F(A)$, it is already inhabited in T by some term $\Gamma \vdash_{\mathsf{T}} \overline{a} : A$, which moreover is a *lifting* of a, in that we can prove $F(\Gamma) \vdash_{\mathsf{S}} F(\overline{a}) = a : F(A)$ in S. Type-contractibility asserts the same sort of lifting property for types derivable in S over a context from T.

This syntactic formulation of type-contractibility has been considered previously by Hofmann as a conservativity principle: see the discussion of logical frameworks in [**Hof97**, §4], and Example 4.2.10 below.

4.2.8. Note that while neither form of contractibility directly provides any kind of lifting for definitional equality judgements, in the presence of identity types one can obtain weak forms of such liftings just from term-contractibility. If for instance $\Gamma \vdash_{\mathsf{T}} a, a' : A$ and $F(\Gamma) \vdash_{\mathsf{S}} F(a) = F(a') : F(A)$, then term-contractibility lets us lift $r(F(a))$ to some term $\Gamma \vdash_{\mathsf{T}} \overline{r(F(a))} : \mathsf{Id}_A(a, a')$.

Essentially, definitional equality for terms implies propositional equality, and for types, isomorphism-up-to-propositional-equality ("homotopy-equivalence"); and since these are matters of term-judgements, they can be lifted along a term-contractible map.

Often, term-contractibility implies type-contractibility. In particular, in many important theories, the type-forming axioms do not mention any of the specific term-constructors. From this it follows that if $F \colon \mathsf{T} \longrightarrow \mathsf{S}$ is a morphism between two such theories, where S has the same type-forming rules as T, then if F is term-contractible, it is also type-contractible. However, we will not need this fact in the sequel.

Example 4.2.9. For any context morphism $f : \Delta \longrightarrow \Theta$, the induced map of slices $f^* \colon \mathsf{T}/\!/\Theta \longrightarrow \mathsf{T}/\!/\Delta$ is orthogonal to i_0^{tm} just if f admits an elim-structure as defined in 1.2.8 above—that is, if f is a left map in the sense of Gambino–Garner [**GG08**]—or syntactically, if f admits an "elimination rule" (and associated computation rule):

$$\frac{\vec{y} : \Theta \vdash C(\vec{y})\ \mathsf{type} \qquad \vec{z} : \Delta \vdash d(\vec{z}) : C(f(\vec{z}))}{\vec{x} : \Theta \ \vdash \ \mathsf{elim}_f(\vec{y}.C, \vec{z}.d; \vec{x}) : C(\vec{x})}$$

$$\begin{tikzcd}
\mathsf{T}_\Phi[\diamond \vdash A\ \mathsf{type}] \arrow[r, "C"] \arrow[d, tail, "i_0^{\mathrm{tm}}"'] & \mathsf{T}/\!/\Theta \arrow[d, "f^*"] \\
\mathsf{T}_\Phi[\diamond \vdash a : A] \arrow[r, "d"] \arrow[ur, dotted, "e_{C,d}"] & \mathsf{T}/\!/\Delta
\end{tikzcd}$$

It is fully term-contractible exactly if every pullback of f along a dependent projection is a Gambino–Garner left map, or equivalently if it supports the "Frobenius" form of this elimination rule, with extra dependent premises in the context, i.e.

$$\vec{y} : \Theta, w : \Xi(\vec{y}) \vdash C(\vec{y}, \vec{w}) \ \mathsf{type} \ldots$$

This follows from f alone being a left map as long as $\mathbf{T}$ has Π-types (by standard arguments), or identity types (by [**GG08**, 5.2.1]).

In particular, for every reflexivity map $\vec{r}\colon \Delta.B \longrightarrow \Delta.B.B.\mathsf{Id}_B$, the map r^* between slices is term-contractible; similarly for the one-ended variant of Id-elim.

Analogously to the above, "large elimination" rules give type-contractibility; but this will not concern us here.

Example 4.2.10. As remarked above, if $\mathbf{T}$ is any theory over some standard type system and $\mathbf{T}^{\mathrm{LF}}$ is its presentation in a logical framework, then according to [**Hof97**, §4], the interpretation of $\mathbf{T}$ in $\mathbf{T}^{\mathrm{LF}}$ is type-contractible.

More precisely, many standard type systems Φ (including all the combinations of rules and constructors we have considered) admit, in a uniform manner, a presentation by purely algebraic rules over the *logical framework* system LF (consisting, roughly, of strong Π-types and a single universe), and moreover (soundness of the presentation) each theory $\mathbf{T}$ has a natural interpretation into its LF presentation $\mathbf{T}_{\mathrm{LF}}$. In categorical terms, we have a functor $\mathbf{DTT}_\Phi \longrightarrow \mathbf{DTT}_{\mathrm{LF}}$, and natural transformation

$$\begin{tikzcd} \mathbf{DTT}_\Phi \arrow[rr, "(-)_{\mathrm{LF}}"] \arrow[dr, "U"'] & \arrow[d, phantom, "\nearrow"] & \mathbf{DTT}_{\mathrm{LF}} \arrow[dl, "U"] \\ & \mathbf{DTT} & \end{tikzcd}$$

Then the results of [**Hof97**, §4] state that the components $\mathbf{T} \longrightarrow \mathbf{T}_{\mathrm{LF}}$ of this natural transformation are type-contractible. (Recall from 4.2.6 that contractibility of a map of theories is independent of what system it is considered over.)

In fact, although not explicitly stated there, the proof given also shows (since the presheaf model is full) that these translations are *faithful*: we can lift not only terms, but *definitional* equalities between them.

4.2.11. Since **DTT** is the category of models of an algebraic theory, and so is locally presentable, we can use the machinery of [**Gar09d**] (the "algebraic small-object argument") to construct an algebraic weak factorisation system[3] on **DTT**, using the universal extensions (term, type, or both) as the generating left maps. The algebraic right maps in the resulting system are then just maps equipped with (term-, type-) contractions; the algebraic left maps are maps presented as (term, type, term-and-type) extensions.

We can think of the maps i_n^{ty}, i_n^{tm} here as *generating cofibrations* in a putative model structure on **DTT**, and the contractible maps as the *trivial fibrations*; this idea will be powerful in our discussion of $\overline{J}$ below.

Extensions by propositional copies: the conservativity principle $\overline{J}$.

4.2.12. One of the fundamental lemmas for many logical systems is that *extension by definitions* should be well behaved: that if we extend a theory T by adding a new term a', and an axiom that a' is equal to some pre-existing term a of T, then the resulting theory is in some sense equivalent to T.

For dependent type theories, this is clear when the new constructor is posited to be *definitionally* equal to an existing one: the resulting theory $\mathsf{T}[a' := a]$ is isomorphic to T itself.

However, one may also wish to understand a weaker situation, where the new term is only posited to be *propositionally* equal to the existing one;

[3] aka natural weak factorisation system

that is, where we extend T by axioms

$$\frac{}{\vec{x} : \Gamma \vdash a'(\vec{x}) : A(\vec{x})} \qquad \frac{}{\Gamma \vdash l(\vec{x}) : \mathsf{Id}_A(a'(\vec{x}), a(\vec{x}))}$$

Briefly, denote the resulting theory by $\mathsf{T}[a' :\approx a]$, or when more detail is needed, by $\mathsf{T}[\vec{x} : \Gamma \ \vdash \ a'(\vec{x}) :\approx_{l(\vec{x})} a(\vec{x}) : A(\vec{x})]$ or similar.

Categorically, extensions of this form are precisely pushouts of the universal ones

$$\mathsf{T}_\Phi[\Gamma_{(n)} \vdash a : A] \rightarrowtail \mathsf{T}_\Phi[\Gamma_{(n)} \vdash a : A][a' :\approx a].$$

We will call (possibly transfinite) compositions of such pushouts *extensions by propositional copies*, and write them as $\mathsf{T}[a'_i(\vec{x}) :\approx a_i(\vec{x})]$. We are once again dealing with a class of relative cell complexes, so it is, as ever, closed under pushouts and (transfinite) composition.

Note also that each of the source or target inclusions $\mathsf{G}_n \rightarrowtail \mathsf{G}_{n+1}$, for $n \geq 1$, is an extension by propositional copies.

What can we now say about the inclusion $\mathsf{T} \rightarrowtail \mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})]$? It is certainly not an isomorphism in general, nor indeed contractible, since a' will generally not be in its image. On the other hand, it is by definition a term-extension. It is also certainly a monomorphism, since it has a retraction $\mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})] \twoheadrightarrow \mathsf{T}$, given by interpreting a' as a and l as $r(a)$.

Our principle $\overline{J}$ describes one sense in which $\mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})]$ may reasonably be equivalent to T:

Definition 4.2.13. Say that $\overline{J}$ *holds for* Φ if for every extension by propositional copies, the retraction $\mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})] \twoheadrightarrow \mathsf{T}$ is term-contractible.

(Actually, we will not need the full strength of the principle as stated here: for our purposes, it would be enough to show that this holds when T can be axiomatised over Φ without any definitional equality axioms, i.e. when $\mathsf{T}_\Phi \rightarrowtail \mathsf{T}$ is a term-extension, or in homotopy-theoretic language, when T is a *cofibrant* theory; we will also only require the case where the term $a(x)$ adjoined is dependent over just a single type.)

Why is this a plausible principle? If we restrict to the case of adjoining copies of *closed* terms, i.e. in the case where $\Gamma = \diamond$, then it is just (the one-ended form of) the Id-ELIM rule. Syntactically, this is the fact that working in an extension by closed terms is equivalent to working over an extended context, with the new variables. Categorically, the extension and its retraction in this case are isomorphic to the maps of slices

$$\begin{tikzcd}
\mathsf{T}[a' :\approx a] \arrow[r, leftrightarrow, "\cong"] \arrow[d, shift left, two heads] & \mathsf{T}/\!\!/(x : A, u : \mathsf{Id}(x,a)) \arrow[d, shift left, two heads] \\
\mathsf{T} \arrow[u, shift left, tail] \arrow[r, leftrightarrow, "\cong"] & \mathsf{T}/\!\!/\diamond \arrow[u, shift left, tail]
\end{tikzcd}$$

induced by the retraction of contexts

$$\diamond \underset{!}{\overset{a, r(a)}{\rightleftarrows}} (x : A, u : \mathsf{Id}(x,a)) \ .$$

Then the map $(a, r(a))\colon \diamond \rightarrowtail x : A, i : \mathsf{Id}(x,a)$ is the introduction map for an instance of the one-ended form of Id-ELIM, so by Example 4.2.9, the retraction of slices $(a, r(a))^*$ is term-contractible.

Thus $\overline{J}$ asserts that something which holds *derivably* for closed terms also holds *admissibly* for open terms: in particular, that open terms of identity types satisfy the same rules as closed ones do.

Unlike most type-theoretic principles, is important to note, however, that $\overline{J}$ is *not* a property of a theory in isolation, but of a *category* of theories, of what I have here called a type system.

Another heuristic motivation for $\overline{J}$ is homotopy-theoretic. Above, we suggested that term-extensions and term-contractible maps could be the cofibrations and trivial fibrations of a model structure on $\mathbf{DTT}_\Phi$; now, the basic extensions by propositional copies may be taken as candidates for generating trivial cofibrations, so any composite of pushouts of these must again be a trivial cofibration. it is certainly always a term-extension, so to show it trivial, it would suffice to show that its retration is a trivial fibration, which is exactly what $\overline{J}$ asserts.

By the preceding remark, one way to prove $\overline{J}$ is to reduce the general case to the closed case, via Π-types. This is possible, as long as we assume enough ext rules to make sure that identity types on Π-types are well-behaved:

Proposition 4.2.14. *$\overline{J}$ holds for $(\mathsf{Id}, \Pi, \Pi\text{-}\textsc{ext}, \Pi\text{-}\textsc{ext-app-def})$ and any set of constructors extending this.*

Proof. The diagram

$$\begin{tikzcd} \mathbf{T}[a'(\vec{x}) :\approx a(\vec{x}) : A(\vec{x})] \arrow[r, shift left] \arrow[d] & \mathbf{T}[f :\approx (\lambda\vec{x}.\ a(\vec{x})) : \Pi_{\vec{x}} A(\vec{x})] \arrow[l, shift left] \arrow[d] \\ \mathbf{T} \arrow[r, equal] & \mathbf{T} \end{tikzcd}$$

exhibits its left-hand side (the map we wish to show contractible) as a retract of its right-hand side. (The fact that the squares commute and are a retraction require the computation rules for ext.) But the right-hand side is just the closed case of $\overline{J}$, which we've seen is contractible. □

However, Π-EXT-APP-DEF is a very *strict* rule: it holds in for instance the groupoid model, but one would not expect it to hold in most weak higher-categorical models. Happily, though, with a little more work the hypotheses here can be weakened.

Another way to see $\overline{J}$ is as a close cousin of Garner's rule $\Pi\text{-}\mathsf{Id}\text{-}\textsc{elim}$. Recall that this latter asserts that a version of Id-elimination holds over products of identity types $\Pi_x\ \mathsf{Id}(f \cdot x, g \cdot x)$: that these types are inductively generated by the terms $\lambda x.r(f \cdot x)$. But terms of such types are very close to the open terms of $l(\vec{x})$ of identity type that appear in $\overline{J}$—and the connection may be made clear by working in a second-order formulation, using a logical framework as metalanguage. In these terms, a second-order version of $\overline{J}$ can be seen as being a strong functional extensionality principle, similar to

Π-Id-ELIM, but stated without Π-types. It is then not hard to show that this form of $\overline{J}$ is derivable from Π-Id-ELIM, and so:

Proposition 4.2.15. *$\overline{J}$ holds for $(\mathsf{Id}, \Pi, \Pi\text{-EXT}, \Pi\text{-EXT-APP})$, and more generally for any set of constructors which implies Π-Id-ELIM and satisfies Hofmann's conservativity theorem 4.2.10 for logical frameworks.*

Proof. Recall the statement of the rule Π-Id-ELIM:

$$\frac{\begin{array}{c}\Gamma,\ u, v : \Pi_{x:A} B(x),\ w : \Pi_{x:A}\ \mathsf{Id}_{B(x)}(u \cdot x, v \cdot x)\ \vdash\ C(u, v, w)\ \mathsf{type} \\ \Gamma,\ f : (x{:}A)B(x)\ \vdash\ d(f) : C(\lambda f, \lambda f, \lambda(r \circ f)) \\ \Gamma\ \vdash\ k, k' : \Pi_{x:A} B(x) \qquad \Gamma\ \vdash\ l : \Pi_{x:A}\ \mathsf{Id}_{B(x)}(k{\cdot}x, k' \cdot x)\end{array}}{\Gamma\ \vdash\ \mathsf{L}(C, d, k, k', l) : C(k, k', l)}\ \Pi\text{-}\mathsf{Id}\text{-ELIM}$$

and its associated computation rule Π-Id-COMP, concluding:

$$\mathsf{L}(C, d, \lambda h, \lambda h, \lambda(r \circ h)) = d(h) : C(\lambda h, \lambda h, \lambda(r \circ h)).$$

(Recall that $(x{:}A)B(x)$ and $[x{:}A]b(x)$ denote type- and term-abstraction in the metalanguage, while $u \cdot x$ denotes application.)

There are three essentially independent differences between Π-Id-ELIM and the most natural second-order form of $\overline{J}$: firstly, the essential difference, between open terms and terms of Π-type; secondly, the difference between one-ended and two-ended forms; and thirdly, the difference between a term dependent on one and on several variables. We tackle these one at a time, via several intermediate rules.

First, we consider a two-ended form of $\overline{J}$, with the term adjoined dependent on just one variable:

$$\frac{\begin{array}{c}\Gamma,\ \underline{k}, \underline{k}' : (x{:}A)B(x),\ \underline{l} : (x{:}A)\mathsf{Id}_{B(x)}(\underline{k}x, \underline{k}'x)\ \vdash\ \underline{C}(\underline{k}, \underline{k}', \underline{l})\ \mathsf{type} \\ \Gamma,\ \underline{f} : (x{:}A)B(x)\ \vdash\ \underline{d}(\underline{f}) : \underline{C}(\underline{f}, \underline{f}, r \circ \underline{f}) \\ \Gamma\ \vdash\ \underline{k}, \underline{k}' : (x{:}A)B(x) \qquad \Gamma\ \vdash\ \underline{l} : (x{:}A)\mathsf{Id}_{B(x)}(\underline{k}x, \underline{k}'x)\end{array}}{\Gamma \vdash \overline{\mathsf{J}}^2(\underline{C}, \underline{d}, \underline{k}, \underline{k}', \underline{l}) : \underline{C}(\underline{k}, \underline{k}', \underline{l})}\ \overline{J}\text{-2O-2E}$$

with corresponding computation rule concluding that

$$\overline{\mathsf{J}}^2(\underline{C}, \underline{d}, \underline{k}, \underline{k}, r \circ \underline{k}) = \underline{d}(\underline{k}) : C(\underline{k}, \underline{k}, r \cdot \underline{k}).$$

We can now define the eliminator $\overline{\mathsf{J}}$ in terms of L by:

$$\overline{\mathsf{J}}(\underline{C}, \underline{d}, \underline{k}, \underline{k}', \underline{l}) \;:=\; \mathsf{L}(\,[k, k', l]\,\underline{C}([x]k{\cdot}x,\ [x]k'{\cdot}x,\ [x]l{\cdot}x),\ \underline{d},\ \lambda\underline{k},\ \lambda\underline{k}',\ \lambda\underline{l}).$$

It is routine to verify that under the hypotheses of the $\overline{J}$ rule, this typechecks, and satisfies the required computational behaviour.

Next we pass to a one-ended form:

$$\frac{\begin{array}{c} \Gamma\ \vdash \underline{k} : (x\!:\!A)B(x) \\ \Gamma,\ \underline{k}' : (x\!:\!A)B(x),\ \underline{l} : (x\!:\!A)\mathsf{Id}_{B(x)}(\underline{k}x, \underline{k}'x)\ \vdash\ \underline{C}(\underline{k}', \underline{l})\ \mathsf{type} \\ \Gamma\ \vdash\ \underline{d} : \underline{C}(\underline{k}, r \circ \underline{k}) \\ \Gamma\ \vdash\ \underline{k}' : (x\!:\!A)B(x) \qquad \Gamma\ \vdash\ \underline{l} : (x\!:\!A)\mathsf{Id}_{B(x)}(\underline{k}x, \underline{k}'x) \end{array}}{\Gamma \vdash \overline{\mathsf{J}}^1(\underline{k}, \underline{C}, \underline{d}, \underline{k}', \underline{l}) : \underline{C}(\underline{k}', \underline{l})}\ \overline{J}\text{-2o-1e}$$

with computation rule concluding $\overline{\mathsf{J}}^1(\underline{k}, \underline{C}, \underline{d}, \underline{k}, r \circ \underline{k}) = \underline{d}(\underline{k}) : C(\underline{k}, r \cdot \underline{k})$.

This can be derived from the previous form via a couple of auxiliary terms: under the hypotheses of $\overline{J}$-2o-1e, we define

$\overline{\mathsf{J}}^1(\underline{k}, \underline{C}, \underline{d}, \underline{k}', \underline{l}) :=$

$\overline{\mathsf{J}}^2($

$\quad [k_1, k_2 : (x\!:\!A)B(x),\ m : (x\!:\!A)\mathsf{Id}(k_1x, k_2x)]$

$\quad (\ l_1 : (x\!:\!A)\mathsf{Id}(kx, k_1x), l_2 : (x\!:\!A)\mathsf{Id}(kx, k_2x),$

$\quad n : (x\!:\!A)\mathsf{Id}((mx)^*(l_1x), l_2x),\ w : C(k_2, (x\!:\!A)(mx)^*(l_2x))\ )$

$\quad C(k_1, l_1),$

$\quad [f, l_1, l_2, n]\,\overline{\mathsf{J}}^1\text{-}\mathsf{aux}(f, l_1, l_2, n),$

$\quad k,\ k',\ l$

$)\ (r \circ k,\ l,\ \_,\ d)$

where

$\overline{\mathsf{J}}^1\text{-}\mathsf{aux}(f, l_1, l_2, n) :=$

$\overline{\mathsf{J}}^2($

$[u_1, u_2 : (x\!:\!A)\mathsf{Id}(kx, fx),\ n : (x\!:\!A)\mathsf{Id}(u_1 x, u_2 x)]\ (w : C(f, u_2))\ C(f, l_1),$

$[h : (x\!:\!A)\mathsf{Id}(kx, fx)]\ ([w : C(f, h)]\ w),$

$l_1,\ l_2,\ n$

$)$

in which _ and expressions of the form $y^* z$ denote various terms which are easy to derive once their type is inferred.

(This is just analogous to the usual derivation of the one-ended form of Id-ELIM from the two-ended.)

Thirdly, to give a second-order version of the fully general form of $\overline{J}$, we must replace the type A in the rules above by a context Δ:

$$\frac{\begin{array}{c}\Gamma\ \vdash \underline{k} : (\vec{x}\!:\!\Delta)B(\vec{x}) \\ \Gamma,\ \underline{k}' : (\vec{x}\!:\!\Delta)B(\vec{x}),\ \underline{l} : (\vec{x}\!:\!A)\mathsf{Id}_{B(\vec{x})}(\underline{k}\vec{x}, \underline{k}'\vec{x})\ \vdash\ \underline{C}(\underline{k}', \underline{l})\ \mathsf{type} \\ \Gamma\ \vdash\ \underline{d} : \underline{C}(\underline{k}, r \circ \underline{k}) \\ \Gamma\ \vdash\ \underline{k}' : (\vec{x}\!:\!\Delta)B(\vec{x}) \qquad \Gamma\ \vdash\ \underline{l} : (\vec{x}\!:\!\Delta)\mathsf{Id}_{B(\vec{x})}(\underline{k}\vec{x}, \underline{k}'\vec{x})\end{array}}{\Gamma \vdash \overline{\mathsf{J}}(\underline{k}, \underline{C}, \underline{d}, \underline{k}', \underline{l}) : \underline{C}(\underline{k}', \underline{l})}$$

This is again straightforward but still lengthier than the preceding derivation; for this reason, and also since as remarked above we will only apply $\overline{J}$ when Δ consists of a single type, we omit the details here.

From this last second-order form we are ready to deduce the original form of $\overline{J}$, but it is still not quite immediate. Since we have used second-order reasoning, this form gives us contractibility not between a theory $\mathbf{T}$ and an extension $\mathbf{T}[a'(\vec{x}) :\approx a(\vec{x})]$, but between $\mathbf{T}_{\mathrm{LF}}$ and its extension by terms $a' : [\vec{x}]A,\ l : [\vec{x}]\mathsf{Id}(a'(\vec{x}), a(\vec{x}))$ (briefly, $\mathbf{T}_{\mathrm{LF}}[a', l]$). But given any type to lift from T to $\mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})]$, consider the diagram

$$\begin{tikzcd}
\bullet \arrow[r] \arrow[d, tail, "i_n^{\mathrm{tm}}"'] & \mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})] \arrow[r, "\triangleright"] \arrow[d] & \mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})]_{\mathrm{LF}} \arrow[r, "\cong"] & \mathsf{T}_{\mathrm{LF}}[a', l] \arrow[d, "\triangledown"] \\
\bullet \arrow[r] & \mathsf{T} \arrow[rr, "\triangleright"] & & \mathsf{T}_{\mathrm{LF}}
\end{tikzcd}$$

The contractibility of the maps $\mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})] \longrightarrow \mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})]_{\mathrm{LF}}$ and $\mathsf{T}_{\mathrm{LF}}[a', l] \longrightarrow \mathsf{T}_{\mathrm{LF}}$ lets us lift to a term in $\mathsf{T}[a'(\vec{x}) :\approx a(\vec{x})]$ which commutes correctly down to T_{LF}; but now by faithfulness of $\mathsf{T} \longrightarrow \mathsf{T}_{\mathrm{LF}}$, it must in fact commute correctly already into T. $\square$

(It is tempting to wonder if the can be reversed; however, this is at least not obviously possible. The obvious candidate for defining L in terms of $\overline{\mathsf{J}}$,

$$\mathsf{L}(C, d, k, k', l) \;:=\; \overline{\mathsf{J}}(\,[\underline{k}, \underline{k}', \underline{l}]\, C(\lambda \underline{k}, \lambda \underline{k}', \lambda \underline{l}),\; d,\; [x]\, k{\cdot}x,\; [x]\, k'{\cdot}x,\; [x]\, l{\cdot}x),$$

does indeed typecheck successfully; but the desired computation rule only holds up to propositional equality, not definitional.)

Thus in theories with reasonably strong functional extensionality principles, $\overline{J}$ holds, and holds robustly: it is derivable, so will continue to hold under strengthenings of the system. However, it can fail as we weaken the system:

Proposition 4.2.16. *For any set of constructors Φ including Π-types, $\overline{J}$ implies a weak form of functional extensionality: if $x : A \vdash k(x), k'(x) : B$ and $x : A \vdash l(x) : \mathsf{Id}(k(x), k'(x))$, then there is some term $\hat{l}$ for which $\vdash \hat{l} : \mathsf{Id}(\lambda x.\, k(x),\, \lambda x.\, k'(x))$.*

Proof. $\overline{J}$ tells us that the map $\mathsf{T}_\Phi[k(x), k'(x), l(x)] \twoheadrightarrow \mathsf{T}_\Phi[k(x)]$ is term-contractible; applying this to the type $\mathsf{Id}(\lambda x.\, k(x),\, \lambda x.\, k'(x))$ upstairs and the term $r(\lambda x.\, k(x))$ downstairs yields a term as desired. $\square$

Corollary 4.2.17. *$\overline{J}$ fails for (Id, Π, η) and (Id, Π).*

Proof. The well-known failures of EXT in these systems are also failures of the conclusion of Proposition 4.2.16. □

However, these failures involve essential use of Π-types.

Conjecture 4.2.18. $\overline{J}$ *holds for* (Id).

Proposition 4.2.17 shows that if the conjecture is true, then $\overline{J}$ is not stable under extensions of the constructor sets, so can't hold for (Id) as robustly as it does for $(\mathsf{Id}, \Pi, \Pi\text{-EXT-APP})$: it may be *admissible* for the type theory with just Id-types, but it cannot be *derivable*.

4.2.19 (Variant forms of $\overline{J}$)**.** In 1.1.6, we considered alternatives to the standard Id-ELIM rules. Several of these correspond to analogous variants of $\overline{J}$.

Most straightforwardly, $\overline{J}$ as given here is most akin to the one-ended form of Id-ELIM; analogously, one can also consider a two-ended form of $\overline{J}$, in which $a(\vec{x})$ as well as $a'(\vec{x})$ is freely adjoined. Indeed, the applications of $\overline{J}$ we use later are all of this form; but using the one-ended form somewhat simplifies the presentation of the results of this section.

One could also consider a principle one might call $\overline{K}$, along the lines of Streicher's K, giving contractibility between theories $\mathbf{T}[a(\vec{x})]$ and $\mathbf{T}[a(\vec{x});\, l(\vec{x})\colon \mathsf{Id}(a(\vec{x}), a(\vec{x}))]$. Like K, this of course collapses (at least up to propositional equality) all the higher-dimensional structure.

Even stronger (or at least possibly so), one could consider a similar analogue of the reflection rule, asserting that for any (pre-existing) terms $a(\vec{x})$, $a'(\vec{x})$, $l(\vec{x})$ of a theory $\mathbf{T}$, the map $\mathbf{T} \longrightarrow \mathbf{T}[a'(\vec{x}) = a(\vec{x}), l(\vec{x}) = r(a(\vec{x}))]$ is term-contractible; this simply adds, by fiat, all the propositional consequences of REFL.

4.2.20. This is perhaps the place to mention a homotopy-theoretic take on the difference between J and K (or between $\overline{J}$ and $\overline{K}$), and how K trivialises the higher-categorical structure. In terms of the homotopy theory given by the universal extensions, $\overline{J}$ asserts roughly that the inclusion of one endpoint into a line segment is a weak equivalence (and moreover is preserved by certain pushouts along cofibrations). This is homotopically essential, and powerful. $\overline{K}$ forces the inclusion of the basepoint into a *loop* to be a weak equivalence. It is then clear that the structures represented by such loops must be homotopically trivial, and more usefully, the topological intuition which makes this clear can be pulled back through the dictionary to recover the usual type-theoretic arguments.

4.3. Contractible operads; weak ω-categories from DTT

In this section, we investigate various conditions under which we can map some contractible operad into $\mathrm{End}(\mathbf{G}_\bullet)$, and hence give a weak ω-category structure. In summary, we obtain a weak ω-structure:

(1) on $\mathcal{C}\ell_\omega$, conjecturally (depending on $\overline{J}$), for all theories with Id-types;
(2) on $\mathcal{C}\ell_\omega$, unconditionally, for theories with Id- and Π-types and the extensionality rules Π-EXT and Π-EXT-APP; and
(3) on ${}^\Pi\mathcal{C}\ell_\omega$, for theories with Id- and Π-types and the Π-η rule.

$\mathrm{End}(\mathbf{G}_\bullet)$ in theories with Id-types.

Theorem 4.3.1. *If $\overline{J}$ holds for Id, then $\mathrm{End}(\mathbf{G}_\bullet^{\mathsf{Id}})$ is contractible.*

Proof. As seen in the second proof of Theorem 3.3.1, contractibility for this operad demands that given any pasting diagram $\pi \in T1(n)$, and $(F_0, G_0, \dots G_{n-1})$ as in Fig. 4.3.B, we must construct H to complete the map of spans; more concisely, we must complete the triangle

$$\begin{tikzcd} \partial\mathbf{G}_n \arrow[r, "{[F_i,G_i]}"] \arrow[d, tail] & \mathbf{T}_{\mathsf{Id}}[\partial\hat{\pi}] \arrow[r, tail] & \mathbf{T}_{\mathsf{Id}}[\hat{\pi}] \\ \mathbf{G}_n \arrow[urr, dotted] & & \end{tikzcd} .$$

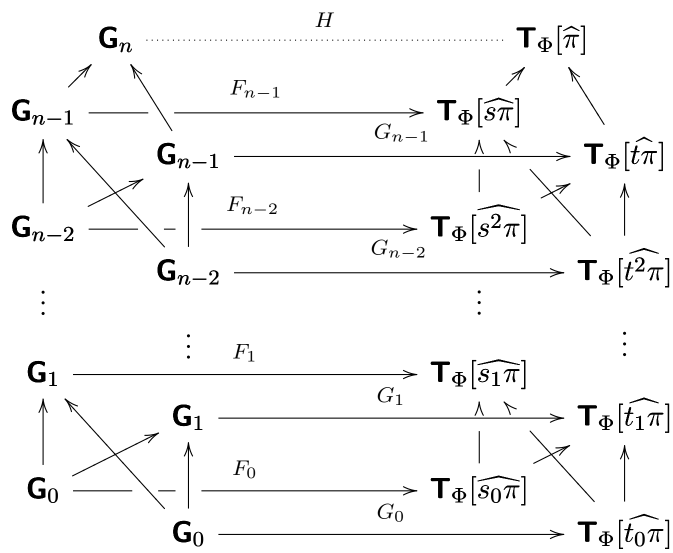


FIGURE 4.3.B. Contractibility in an endomorphism operad

The cases $n = 0, 1$ are dealt with by 4.1.8 et seq. above: in dimension 0, contractibility is satisfied just by the existence of the 0-dimensional identity operation; while in dimension 1, it demands the existence of composition operations of each arity, which are supplied by the map $\psi\colon P_{\mathbf{Cat}} \longrightarrow \mathrm{tr}^1 \mathrm{End}(\mathsf{G}_\bullet)$.

On the other hand, when $n > 0$, it is immediate from the axiomatisations given that the map $\partial\mathsf{G}_n \rightarrowtail \mathsf{G}_n$ is a term-extension. Also, according to the pruning procedure described in 2.4.1 above, we can obtain $\mathsf{T}_{\mathsf{Id}}[\hat{\pi}]$ as an extension of $\mathsf{T}_{\mathsf{Id}}[\widehat{s_1\pi}]$ by propositional copies: it is a composition of maps $\mathsf{T}_{\mathsf{Id}}[\rho^-] \rightarrowtail \mathsf{T}_{\mathsf{Id}}[\rho]$, each of which is an extension by propositional copies by the pushout squares of 2.4.2. So *provided* $\overline{J}$ *holds for* $\mathbf{DTT}_{\mathsf{Id}}$, the retraction

$$\mathsf{T}_{\mathsf{Id}}[\hat{\pi}] \twoheadrightarrow \mathsf{T}_{\mathsf{Id}}[\widehat{s_1\pi}]$$

(interpreting all identity cells as reflexivity terms) is term-contractible.

Thus to complete the triangle above, it is sufficient to complete the square

$$\begin{tikzcd} \partial \mathsf{G}_n \arrow[r, "{[F_i,G_i]}"] \arrow[d, tail] & \mathsf{T}_{\mathsf{Id}}[\hat{\pi}] \arrow[d] \\ \mathsf{G}_n \arrow[r, dotted] & \mathsf{T}_{\mathsf{Id}}[\widehat{s_1\pi}] \end{tikzcd} ,$$

i.e. to complete a triangle of the form

$$\begin{tikzcd} \partial \mathsf{G}_n \arrow[d, tail] \arrow[dr] & \\ \mathsf{G}_n \arrow[r, dotted] & \mathsf{T}_{\mathsf{Id}}[\widehat{s_1\pi}] \end{tikzcd} .$$

But now $s_1\pi$ is just some $path_l$, so as in 4.1.9, 4.1.10 we have an explicit axiomatisation of $\mathsf{T}_{\mathsf{Id}}[\widehat{s_1\pi}]$, and we know that this theory enjoys canonicity. So in trying to extend $[F_i, G_i]$ along $\partial \mathsf{G}_n \rightarrowtail \mathsf{G}_n$, we have interpreted i-src and i-tgt in $\mathsf{T}_{\mathsf{Id}}[\widehat{s_1\pi}]$, for $i < n$, and wish to interpret n-cell; i.e. we wish to prove a propositional equality between the interpretations of $s_{n-1}(x)$ and $t_{n-1}(x)$. But by canonicity, and the simplicity of our set of constructors, any two terms of the same type in $\mathsf{T}_{\mathsf{Id}}[\widehat{s_1\pi}]$ in context $x : A_0$ are *definitionally* equal; so interpreting c_n as a reflexivity term, we are done. (Specifically, s_1, t_1 must both be interpreted as the obvious composite of basic constructors described in 4.1.9, and for $i > 1$, s_i and t_i must be interpreted as the reflexivity term over s_{i-1}, t_{i-1}.) $\square$

End($\mathsf{G}_\bullet$) **in theories with** Π**-ext-app.**

4.3.2. By turning our attention to theories with not only Id-types but also Π-types, Π-EXT, and Π-EXT-APP, we ensure that $\overline{J}$ holds unconditionally. However, this comes at the possible cost of normalisation and canonicity. Thus, in trying to repeat the argument above to show that End($\mathsf{G}_\bullet$) is contractible, we fall at the last hurdle: we do not know that $[F_i, G_i]$ gives the "correct" map $\partial \mathsf{G}_n \longrightarrow \mathsf{T}_{\Pi\text{-EXT-APP}}[\widehat{path_l}]$.

To remedy this, we simply restrict to the sub-operad of operations for which this holds. This is just an elaboration of the tactic used in [**GvdB10b**], of restricting to the operad of point-preserving operations, as discussed in 3.3.2.

Definition 4.3.3. For $\mathcal{E}$ any category with pullbacks, define a monoidal category $\mathbf{Ref}_1\mathbf{Glob}(\mathcal{E})$ as follows:

Objects in dimension n are globular objects $A_\bullet$ of $\mathcal{E}$, together with *reflexivity data from dimension 1*: for $1 \leq i \leq n$, a map $r_i \colon A_1 \longrightarrow A_i$, such that $s_i r_{i+1} = t_i r_{i+1} = r_i$, and $r_1 = 1_{A_1}$.

$$\begin{tikzcd}
A_n \arrow[d, shift right, "s"'] \arrow[d, shift left, "t"] \\
A_{n-1} \\
\vdots \\
A_2 \arrow[d, shift right, "s"'] \arrow[d, shift left, "t"] \\
A_1 \arrow[u, bend left, "r"] \arrow[uuu, bend left, "r"] \arrow[uuuu, bend left=50, "r"] \arrow[d, shift right, "s"'] \arrow[d, shift left, "t"] \\
A_0
\end{tikzcd}$$

A map between two of these is a map (f_i, g_i, h) between their "underlying" spans, with $f_0 = g_0$, $f_1 = g_1$, and commuting with the reflexivity data in that $f_i r_i = g_i r_i = r_i f_1$, and $h r_n = r_n f_1$.

The monoidal globular structure of $\mathbf{Ref}_1\mathbf{Glob}(\mathcal{E})$ is lifted from that of $\mathbf{Spans}(\mathcal{E})$. Any tensor product $A_\bullet \otimes_k B_\bullet$ in $\mathbf{Spans}(\mathcal{E})$ of globular objects is again globular, and reflexivity data on the multiplicands lifts naturally to reflexivity data on the product; and similarly, the units over globular objects are globular and carry natural reflexivity data.

We thus have a monoidal globular category and faithful forgetful functor

$$\mathbf{Ref}_1\mathbf{Glob}(\mathcal{E}) \longrightarrow \mathbf{Spans}(\mathcal{E}).$$

(Note that, as the definition of the maps hints, the globularity condition on objects in dimensions > 1 is not actually required here, and it would arguably be more natural to omit it. However, all spans occurring in the construction of endomorphism operads remain fully globular, so it makes no difference for present purposes, and it simplifies the specification of the reflexivity data.)

Instantiating this construction with $\mathcal{E} = \mathbf{DTT}^{\mathrm{op}}$, the globes $\mathsf{G}_\bullet$ lift (using their reflexivity maps) to a globular object in $\mathbf{Ref}_1\mathbf{Glob}(\mathbf{DTT}^{\mathrm{op}})$. We thus obtain a new endomorphism operad for them; since the forgetful functor is faithful, this is just a sub-operad of the old

$$\mathrm{End}_{\mathbf{Ref}_1\mathbf{Glob}(\mathbf{DTT}^{\mathrm{op}})}(\mathsf{G}_\bullet) \rightarrowtail \mathrm{End}_{\mathbf{Spans}(\mathbf{DTT}^{\mathrm{op}})}(\mathsf{G}_\bullet)$$

consisting of operations as in Figure 4.1.A, satsifying the additional condition that whenever all the higher term-formers in $\mathsf{T}[\pi]$ are interpreted as reflexivity terms, then the interpretations of the term-formers of $\mathsf{G}_\bullet n$ also compute down to reflexivity terms.

4.3.4. From here we need to restrict still a little further before we have a contractible operad: we need to look at just those operations which do the correct thing in dimensions ≤ 1. Specifically, the map $\psi \colon P_{\mathbf{Cat}} \longrightarrow \mathrm{tr}^1\mathrm{End}(\mathsf{G}_\bullet)$ is easily seen to factor through $\mathrm{End}_{\mathbf{Ref}_1\mathbf{Glob}(\mathbf{DTT}^{\mathrm{op}})}(\mathsf{G}_\bullet)$; so let $Q_{\Pi\text{-}\textsc{ext}\text{-}\textsc{app}}$ be the pullback

$$\begin{tikzcd}
Q_{\Pi\text{-}\textsc{ext}\text{-}\textsc{app}} \arrow[r, tail] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & \mathrm{End}_{\mathbf{Ref}_1\mathbf{Glob}(\mathbf{DTT}^{\mathrm{op}})}(\mathsf{G}_\bullet) \arrow[d, "\eta"'] \\
P_{\mathbf{str}\text{-}\omega\text{-}\mathbf{Cat}} = IP_{\mathbf{Cat}} \arrow[r, tail, "I\psi"] & I\mathrm{tr}^1\mathrm{End}_{\mathbf{Ref}_1\mathbf{Glob}(\mathbf{DTT}^{\mathrm{op}})}(\mathsf{G}_\bullet)
\end{tikzcd}$$

(where $I \colon 1\text{-}\mathbf{Opd} \longrightarrow \omega\text{-}\mathbf{Opd}$ is the "indiscrete ω-operad" functor, right adjoint to tr^1 as described in 2.3.9); then $Q_{\Pi\text{-}\textsc{ext}\text{-}\textsc{app}}$ consists precisely of those operations of $\mathrm{End}_{\mathbf{Ref}_1\mathbf{Glob}(\mathbf{DTT}^{\mathrm{op}})}(\mathsf{G}_\bullet)$ whose ≤ 1-dimensional parts lie in the image of ψ. We are now set up for:

Theorem 4.3.5. *The suboperad $Q_{\Pi\text{-}\textsc{ext}\text{-}\textsc{app}} \rightarrowtail \mathrm{End}(\mathsf{G}_\bullet)$ is contractible.*

Proof. Contractibility in dimensions ≤ 1 holds by fiat: in these dimensions, $Q_{\Pi\text{-}\textsc{ext}\text{-}\textsc{app}}$ is isomorphic to the terminal operad.

For higher dimensions, note that operations in this new operad are just as in the old, except that additionally the maps involved must commute with the reflexivity data. So contractibility now demands that for $\pi \in \mathbf{pd}_n$, and suitable $(F_i, G_i)_{i<n}$, we must produce a map $H : \mathsf{G}_n \longrightarrow \mathsf{T}_{\Pi\text{-EXT-APP}}[\widehat{\pi}]$ making both squares in the following diagram commute:

$$\begin{tikzcd}
\partial\mathsf{G}_n \arrow[r, "{[F_i,G_i]}"] \arrow[d, tail] & \mathsf{T}_{\Pi\text{-EXT-APP}}[\partial\hat{\pi}] \arrow[d, tail] \\
\mathsf{G}_n \arrow[r, dotted] \arrow[d, -open triangle] & \mathsf{T}_{\Pi\text{-EXT-APP}}[\hat{\pi}] \arrow[d, -open triangle] \\
\mathsf{G}_1 \arrow[r, "F_1=G_1"'] & \mathsf{T}_{\Pi\text{-EXT-APP}}[\widehat{s_1\pi}]
\end{tikzcd}$$

But the overall rectangle commutes, so (rearranging it as a square) the desired filler follows by $\overline{J}$. $\square$

$\mathrm{End}({}^{\Pi}\mathsf{G}_\bullet)$ in theories with Id, Π, Π-η. Turning our attention to ${}^{\Pi}\mathsf{G}_\bullet$, and considering theories with the rules $(\mathsf{Id}, \Pi, \Pi\text{-}\eta)$, we are now well set up to construct a contractible sub-operad Q_Π of $\mathrm{End}({}^{\Pi}\mathsf{G}_\bullet)$. Specifically, take Q_Π to be the normalised core of $\mathrm{End}({}^{\Pi}\mathsf{G}_\bullet)$—that is, all those operations whose 0-dimensional source and target are the operad unit $1_{{}^{\Pi}\mathsf{G}_0}$. Since we have canonicity, we do not need to restrict further as we did in the construction of $Q_{\Pi\text{-EXT-APP}}$.

Theorem 4.3.6. *The operad Q_Π is contractible.*

Proof. The proof of Theorem 4.3.1 goes through almost verbatim. The only difference is that the type-contractibility of the maps into which we factor $\mathsf{T}_\Phi[\widehat{\pi}]^\Pi \twoheadrightarrow \mathsf{T}_\Phi[\widehat{s_1\pi}]^\Pi$ does not depend on $\overline{J}$, instead following just from Example 4.2.9. $\square$

Classifying weak ω-categories.

4.3.7. Theorems 4.3.1, 4.3.5, and 4.3.6 give three situations in which there is a map from some contractible operad Q into $\mathrm{End}(\mathsf{G}_\bullet)$ (or $\mathrm{End}({}^{\Pi}\mathsf{G}_\bullet)$). In each case, this induces a map $L \longrightarrow Q \longrightarrow \mathrm{End}({}^{(\Pi)}\mathsf{G}_\bullet)$, and hence a functor ("restriction of scalars") $\mathrm{End}({}^{(\Pi)}\mathbf{G}_\bullet)\text{-}\mathbf{Alg} \longrightarrow L\text{-}\mathbf{Alg} = \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat}$. We thus have:

Theorem 4.3.8.

(1) *There is a functor* $\mathcal{C}\ell_\omega \colon \mathbf{DTT}_{\Pi\text{-EXT-APP}} \longrightarrow \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat}$*, giving the "classifying weak* ω*-category" of any theory with at least* Id*-types,* Π*-types, and the* Π-EXT*,* Π-EXT-APP *rules.*

(2) *If* $\overline{J}$ *holds for* Id*, then we moreover have* $\mathcal{C}\ell_\omega \colon \mathbf{DTT}_{\mathsf{Id}} \longrightarrow \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat}$*, giving the classifying weak* ω*-category for any theory with at least* Id*-types.*

(3) *There is a functor* ${}^{\Pi}\mathcal{C}\ell_\omega \colon \mathbf{DTT}_{\mathsf{Id},\Pi} \longrightarrow \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat}$ *giving a variant of the classifying weak* ω*-category, for any theory with at least* Π*-types,* Id*-types, and the rule* Π-η.

This statement, while pleasing, has a few loose ends which deserve to be tied up.

4.3.9. Firstly, what can be said about non-contextual CwA's? This is an important question for applications, since CwA's arising semantically are rarely contextual, and we would like to get higher categories not just from syntactically presented theories, but also from models.

Given a general CwA $\mathcal{C}$, we can take $\mathcal{C}\ell_\omega(\mathcal{C}/\!/\diamond)$, the classifying weak ω-category of its contextual core. (Since $/\!/\diamond$ is a coreflection, we could equivalently use $\mathcal{C}\ell_\omega^-$.) Thinking of general CwA's as semantic and contextual ones as syntactically presented, this is just taking the classifying weak ω-category of the internal language of $\mathcal{C}$.

Fortunately, we have not lost much. Most semantic CwA's, though not contextual, are *accessible*, and hence equivalent to their contextual core, so in these cases $\mathcal{C}\ell_\omega(\mathcal{C}/\!/\diamond)$ should remain at least weakly equivalent to any other reasonable definition of the classifying weak ω-category of $\mathcal{C}$.

On the other hand, the loss of inaccessible objects seems inevitable: there is no way to define "identity objects" over an arbitrary object, if one cannot express it as a context of types, and without this, it is not clear how one would expect the object to participate as a 0-cell in the classifying weak ω-category.

4.3.10. Secondly, there is an obvious abuse of notation: if $\overline{J}$ holds for Id, then we are overloading $\mathcal{C}\ell_\omega$ not only with different codomains, a mild and common sin, but also with different domains, a potentially worse one. Given a theory T with at least the Π-Id-ELIM rule, we could compute $\mathcal{C}\ell_\omega(\mathsf{T})$ as such a theory, or we could treat it as a theory over Id-, and compute $\mathcal{C}\ell_\omega(\mathsf{T})$ from there. Will these agree? In other words, does $\mathcal{C}\ell_\omega$ commute with the forgetful functor $U\colon \mathbf{DTT}_{\Pi\text{-EXT-APP}} \longrightarrow \mathbf{DTT}_{\mathsf{Id}}$, as in the following diagram?

$$\begin{tikzcd}
\mathbf{DTT}_{\Pi\text{-EXT-APP}} \arrow[r] \arrow[d, bend left, "U"] \arrow[d, phantom, "\dashv"] & \mathrm{End}(\mathsf{G}_\bullet^{\Pi\text{-EXT-APP}})\text{-}\mathbf{Alg} \arrow[r] \arrow[d] \arrow[dr, phantom, "(?)"] & Q_{\Pi\text{-EXT-APP}}\text{-}\mathbf{Alg} \arrow[dr] & \\
\mathbf{DTT}_{\mathsf{Id}} \arrow[u, bend left, "F"] \arrow[r] \arrow[dr] & \mathrm{End}(\mathsf{G}_\bullet^{\mathsf{Id}})\text{-}\mathbf{Alg} \arrow[rr] \arrow[d] & & \mathbf{wk}\text{-}\omega\text{-}\mathbf{Cat} \arrow[dll] \\
& \widehat{\mathbb{G}} & &
\end{tikzcd}$$

Most parts of this diagram are easily seen to commute up to natural isomorphism. The essential point is that the globes $\mathsf{G}_\bullet^{\Pi\text{-EXT-APP}}$ are just the image of $\mathsf{G}_\bullet^{\mathsf{Id}}$ under the left adjoint $F\colon \mathbf{DTT}_{\mathsf{Id}} \longrightarrow \mathbf{DTT}_{\Pi\text{-EXT-APP}}$, since their axiomatisations involve only Id-types; and hence as functors into globular sets, or even into $\mathrm{End}(\mathsf{G}_\bullet^{\mathsf{Id}})$-algebras, we have $\mathbf{DTT}_{\Pi\text{-EXT-APP}}(\mathsf{G}_\bullet^{\Pi\text{-EXT-APP}}, \mathsf{T}) \cong \mathbf{DTT}_{\mathsf{Id}}(\mathsf{G}_\bullet^{\mathsf{Id}}, U(\mathsf{T}))$.

This leaves, as the dubious part, the square marked by (?). This comes down to the question of whether the corresponding square of operad maps commutes

$$\begin{tikzcd}
L \arrow[r] \arrow[d] \arrow[dr, phantom, "(?)"] & \mathrm{End}(\mathsf{G}_\bullet^{\mathsf{Id}}) \arrow[d] \\
Q_{\Pi\text{-EXT-APP}} \arrow[r] & \mathrm{End}(\mathsf{G}_\bullet^{\Pi\text{-EXT-APP}})
\end{tikzcd}$$

and now the subtlety emerges: the maps out of L are defined in terms of the specific contractions used on $Q_{\Pi\text{-EXT-APP}}$, and these in turn depend on how precisely we have implemented $\overline{J}$.

What we can at least see is that (by the normalisation results used in 4.3.1) $\mathrm{End}(\mathsf{G}_\bullet^{\mathsf{Id}}) \longrightarrow \mathrm{End}(\mathsf{G}_\bullet^{\Pi\text{-EXT-APP}})$ factors through $Q_{\Pi\text{-EXT-APP}}$, so it comes down to commutativity of the triangle

$$\begin{tikzcd} & \mathrm{End}(\mathsf{G}_\bullet^{\mathsf{Id}}) \arrow[d] \\ L \arrow[ur] \arrow[dr] \arrow[r, phantom, "(?)"] & \\ & Q_{\Pi\text{-EXT-APP}} \end{tikzcd}$$

This will certainly hold if $\mathrm{End}(\mathsf{G}_\bullet^{\mathsf{Id}}) \longrightarrow Q_{\Pi\text{-EXT-APP}}$ is a map of operads-with-contraction, i.e. iff it preserves the contraction; and it is fairly routine (though rather notationally fiddly) to show that if the forgetful functor $U\colon \mathbf{DTT}_{\Pi\text{-EXT-APP}} \longrightarrow \mathbf{DTT}_{\mathsf{Id}}$ "preserves the implementation of $\overline{J}$" in an appropriate sense, then this will be the case.

However, it seems unlikely to the present author that the implementation of $\overline{J}$ given above for theories with Π-EXT-APP could be preserved by U. This defect can (as often in problems with contractions) be finessed by some ad hoc modification of the contractions on the operads involved; but essentially, this is a problem that should be resolved not by strict-higher-categorical fiddling, but by the theory of weak higher-categorical equivalence, under which the squares marked (?) would commute *up to weak equivalence.* In lieu of the development of this theory, then, we leave a resolution of this problem aside for now.

Such tools should also allow one to address the question of when ${}^{\Pi}\mathcal{C}\ell_\omega$ may be equivalent to $\mathcal{C}\ell_\omega$.

4.3.11. Finally, there is an abuse of terminology: what, if anything, does the classifying weak ω-category classify?

As given at present, in the form of the functors ${}^{(\Pi)}\mathcal{C}\ell_\omega\colon \mathbf{DTT}_\Phi \longrightarrow \textbf{wk-}\omega\textbf{-Cat}$ above, it cannot classify anything: that is, it cannot have a right adjoint, since it does not preserve the initial object, nor a left, since it does not preserve the terminal object. (The initial theory in $\mathbf{DTT}_\Phi$ is T_Φ, whose $\mathcal{C}\ell_\omega$ is certainly non-empty, containing the empty context $\diamond$ as a 0-cell, and so is not initial in **wk-ω-Cat**. Similarly, the terminal theory in each $\mathbf{DTT}_\Phi$ has one context of each length $l \in \mathbb{N}$, and so its $\mathcal{C}\ell_\omega$ will have $\mathbb{N}$-many 0-cells.)

However, this is not surprising: in the passage from type theories to weak ω-categories, we have forgotten much structure (and co-structure). One of the next important steps in the current program should be the axiomatisation of higher-categorical structure corresponding to the structure on the type theories, in such a way that the classifying ω-categories of theories carry such structure, and (hopefully) do indeed appropriately classify some kind of models in these structured higher categories.

For the 2-truncated case, this is well-worked-out in [**Gar09c**]. However, in higher dimensions, the understanding of appropriate "weak logical structure" is at a very early stage of development; so this too is a story for another day.

Bibliography

[AHW09] Steve Awodey, Pieter Hofstra, and Michael A. Warren, *Martin-Löf complexes*, submitted, 2009, `arXiv:0906.4521`.

[AM06] Thorsten Altenkirch and Conor McBride, *Towards observational type theory*, Manuscript, available online, February 2006.

[AMS07] Thorsten Altenkirch, Conor McBride, and Wouter Swierstra, *Observational equality, now!*, PLPV '07: Proceedings of the 2007 workshop on Programming languages meets program verification (New York, NY, USA), ACM, 2007, pp. 57–68, `doi:http://doi.acm.org/10.1145/1292597.1292608`.

[AR94] Jiří Adámek and Jiří Rosický, *Locally presentable and accessible categories*, London Mathematical Society Lecture Note Series, vol. 189, Cambridge University Press, Cambridge, 1994, `doi:10.1017/CBO9780511600579`. MR 1294136 (95j:18001)

[AW09] Steve Awodey and Michael A. Warren, *Homotopy theoretic models of identity types*, Math. Proc. Cambridge Philos. Soc. **146** (2009), no. 1, 45–55, `arXiv:0709.0248`, `doi:10.1017/S0305004108001783`. MR 2461866

[Bat98] M. A. Batanin, *Monoidal globular categories as a natural environment for the theory of weak n-categories*, Adv. Math. **136** (1998), no. 1, 39–103.

[Car78] John Cartmell, *Generalised algebraic theories and contextual categories*, Ph.D. thesis, Oxford, 1978.

[Car86] ———, *Generalised algebraic theories and contextual categories*, Ann. Pure Appl. Logic **32** (1986), no. 3, 209–243.

[CL04] Eugenia Cheng and Aaron Lauda, *Higher dimensional categories: an illustrated guide book*, 2004, `http://www.cheng.staff.shef.ac.uk/guidebook/index.html`.

[Dyb96] Peter Dybjer, *Internal type theory*, Types for proofs and programs (Torino, 1995), Lecture Notes in Comput. Sci., vol. 1158, Springer, Berlin, 1996, pp. 120–134. MR 1474535 (98e:03040)

[Gar08] Richard Garner, *Homomorphisms of higher categories*, submitted, 2008, `arXiv:0810.4450`.

[Gar09a] ______, *A homotopy-theoretic universal property of Leinster's operad for weak ω-categories*, Math. Proc. Cambridge Philos. Soc. **147** (2009), no. 3, 615–628, doi:10.1017/S030500410900259X. MR 2557146 (2010j:18030)

[Gar09b] ______, *On the strength of dependent products in the type theory of Martin-Löf*, Ann. Pure Appl. Logic **160** (2009), no. 1, 1–12, doi:10.1016/j.apal.2008.12.003. MR 2525970 (2010m:03016)

[Gar09c] ______, *Two-dimensional models of type theory*, Math. Structures Comput. Sci. **19** (2009), no. 4, 687–736, arXiv:0808.2122, doi:10.1017/S0960129509007646. MR 2525957

[Gar09d] ______, *Understanding the small object argument*, Appl. Categ. Structures **17** (2009), no. 3, 247–285, doi:10.1007/s10485-008-9137-4. MR 2506256 (2010a:18005)

[GG08] Nicola Gambino and Richard Garner, *The identity type weak factorisation system*, Theoret. Comput. Sci. **409** (2008), no. 1, 94–109, arXiv:0803.4349, doi:10.1016/j.tcs.2008.08.030. MR 2469279

[GvdB10a] Richard Garner and Benno van den Berg, *Topological and simplicial models of identity types*, ACM Trans. Comput. Log. (2010), to appear, arXiv:1007.4638v1.

[GvdB10b] ______, *Types are weak ω-groupoids*, Proc. Lond. Math. Soc. (2010), Advance Access published, arXiv:0812.0298, doi:10.1112/plms/pdq026, http://plms.oxfordjournals.org/content/early/2010/10/12/plms.pdq026.abstract.

[Hof97] Martin Hofmann, *Syntax and semantics of dependent types*, Semantics and logics of computation (Cambridge, 1995), Publ. Newton Inst., vol. 14, Cambridge Univ. Press, Cambridge, 1997, pp. 79–130, doi:10.1017/CBO9780511526619.004. MR 1629519

[Hov99] Mark Hovey, *Model categories*, Mathematical Surveys and Monographs, vol. 63, American Mathematical Society, Providence, RI, 1999. MR 1650134 (99h:55031)

[HS98] Martin Hofmann and Thomas Streicher, *The groupoid interpretation of type theory*, Twenty-five years of constructive type theory (Venice, 1995), Oxford Logic Guides, vol. 36, Oxford Univ. Press, New York, 1998, pp. 83–111.

[Jac93] Bart Jacobs, *Comprehension categories and the semantics of type dependency*, Theoret. Comput. Sci. **107** (1993), no. 2, 169–207, doi:10.1016/0304-3975(93)90169-T. MR 1201808 (94i:18009)

[Joh02] Peter T. Johnstone, *Sketches of an elephant: a topos theory compendium. Vol. 2*, Oxford Logic Guides, vol. 44, The Clarendon Press Oxford University Press, Oxford, 2002. MR 2063092 (2005g:18007)

[Lei02] Tom Leinster, *A survey of definitions of n-category*, Theory Appl. Categ. **10** (2002), 1–70 (electronic), `arXiv:math/0107188`.

[Lei04] ______, *Higher operads, higher categories*, London Mathematical Society Lecture Note Series, vol. 298, Cambridge University Press, Cambridge, 2004, `arXiv:math/0305049`.

[Lum09] Peter LeFanu Lumsdaine, *Weak ω-categories from intensional type theory (conference version)*, Typed lambda calculi and applications (Berlin), Lecture Notes in Computer Science, vol. 5608, Springer, 2009, pp. 172–187.

[Lum10] ______, *Weak ω-categories from intensional type theory*, Logical Methods in Computer Science **6** (2010), no. 3:24, 1–19 (electronic), `arXiv:0812.0409v4`, `doi:10.2168/LMCS-6(3:24)2010`, `www.lmcs-online.org/ojs/viewarticle.php?id=628`.

[ML75] Per Martin-Löf, *An intuitionistic theory of types: predicative part*, Logic Colloquium '73 (Bristol, 1973), North-Holland, Amsterdam, 1975, pp. 73–118. Studies in Logic and the Foundations of Mathematics, Vol. 80. MR 0387009 (52 #7856)

[ML98] Saunders Mac Lane, *Categories for the working mathematician*, second ed., Graduate Texts in Mathematics, vol. 5, Springer-Verlag, New York, 1998. MR 1712872 (2001j:18001)

[MLM94] Saunders Mac Lane and Ieke Moerdijk, *Sheaves in geometry and logic*, Universitext, Springer-Verlag, New York, 1994, A first introduction to topos theory, Corrected reprint of the 1992 edition. MR 1300636 (96c:03119)

[Mog91] Eugenio Moggi, *A category-theoretic account of program modules*, Math. Structures Comput. Sci. **1** (1991), no. 1, 103–139. MR 1108806 (92h:68057)

[NPS90] Bengt Nordström, Kent Petersson, and Jan M. Smith, *Programming in Martin-Löf's type theory*, International Series of Monographs on Computer Science, vol. 7, The Clarendon Press Oxford University Press, New York, 1990, An introduction. MR 1243882 (94g:68020)

[Pit00] Andrew M. Pitts, *Categorical logic*, Handbook of logic in computer science, Vol. 5, Handb. Log. Comput. Sci., vol. 5, Oxford Univ. Press, New York, 2000, pp. 39–128. MR 1859470 (2002j:03078)

[Str91] Thomas Streicher, *Semantics of type theory*, Progress in Theoretical Computer Science, Birkhäuser Boston Inc., Boston, MA, 1991, Correctness, completeness and independence results, With a foreword by Martin Wirsing. MR 1134134 (93d:68042)

[Str93] Thomas Streicher, *Investigations Into Intensional Type Theory*, Habilitationsschrift, Ludwig-Maximilians-Universität München, November 1993.

[Str00] Ross Street, *The petit topos of globular sets*, Journal of Pure and Applied Algebra **154** (2000), 299–315.

[Voe] Vladimir Voevodsky, *The equivalence axiom and univalent models of type theory*, Lecture delivered at Carnegie Mellon University, February 2010, and unpublished notes, `http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/CMU_talk.pdf`.

[War08] Michael A. Warren, *Homotopy theoretic aspects of constructive type theory*, Ph.D. thesis, Carnegie Mellon University, 2008.

[Web05] Mark Weber, *Operads within monoidal pseudo algebras*, Appl. Categ. Structures **13** (2005), no. 5-6, 389–420, `arXiv:math/0410230`, `doi:10.1007/s10485-005-9000-9`.

[WH10] Michael A. Warren and Pieter Hofstra, *Combinatorial realizability models of type theory*, in preparation, 2010.