A Syntactical Approach to Weak ω-Groupoids

Thorsten Altenkirch[1] and Ondřej Rypáček[2]

1 **Functional Programming Laboratory**
School of Computer Science
University of Nottingham, UK
thorsten.altenkirch@nottingham.ac.uk

2 **Department of Computer Science**
University of Sheffield, UK
ondrej.rypacek@gmail.com

Abstract

When moving to a Type Theory without proof-irrelevance the notion of a setoid has to be generalized to the notion of a weak ω-groupoid. As a first step in this direction we study the formalisation of weak ω-groupoids in Type Theory. This is motivated by Voevodsky's proposal of univalent type theory which is incompatible with proof-irrelevance and the results by Lumsdaine and Garner/van de Berg showing that the standard eliminator for equality gives rise to a weak ω-groupoid.

1998 ACM Subject Classification F.4.1. Lambda calculus and related systems, F.4.1. Mechanical theorem proving, G.0

Keywords and phrases Type Theory,Category Theory,Higher dimensional structures

1 Introduction

The main motivation for the present work is the development of Univalent Type Theory by Voevodsky [22]. In a nutshell, Univalent Type Theory is a variant of Martin-Löf's Type Theory where we give up the principle of uniqueness of identity proofs (UIP) to make it possible to treat equivalence of structures (e.g. isomorphism of sets) as equality. While Voevodsky's motivation comes from Homotopy Theory, Univalent Type Theory has an intrinsic type theoretic motivation in enabling us to treat abstract structures as first class citizens making it possible to combine high level reasoning and concrete applications without unnecessary clutter.

The central principle of Univalent Type Theory is the Univalence Axiom which states that equality of types is weakly equivalent to weak equivalence. Here weak equivalence is a notion motivated by homotopy theoretic models of type theory which can be alternatively understood as a refinement of the notion of isomorphism in the absence of UIP. The Univalence axiom can be viewed as a strong extensionality principle and indeed it implies functional extensionality. As with functional extensionality, univalence doesn't easily fit within the computational understanding of Type Theory, since it does not fit into the common pattern of introduction and elimination rules. The first author has suggested a solution of this problem for functional extensionality [1]: we can justify extensionality by a translation based on the setoid model. This approach was later refined [2] to *Observational Type Theory* which is the base for the development of Epigram 2 [8].

However, Observational Type Theory relies essentially on UIP and hence is incompatible with Univalent Type Theory. To address this we need to replace setoids with a structure able to model non-unique identity proofs. A first step in this direction is the groupoid model [10]

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

but this forces UIP on the next level, i.e. for equality between equality proofs. [1] To be able to model Type Theory without UIP at any level we need to move to ω-groupoids. Moreover, the equalities we need to assume are in general non-strict (i.e. they are not definitional equalities in the sense of Type Theory) and hence we need to look at weak ω-groupoids. Indeed, as [7] and [16] have shown: Type Theory with Identity types gives rise to a weak ω-groupoid.

Our goal is to eliminate univalence by formalizing a weak ω-groupoid model of Type Theory in Type Theory. As a first step we need to implement the notion of a weak ω-groupoid in Type Theory and this is what we do in the present paper. An obvious possibility would have been to implement a categorical notion of weak ω-groupoids (e.g. based on globular operads) in Type Theory. However, this forces us to implement many categorical notions first, generating an avoidable overhead. It also seems that a structure with a more type theoretic flavour is more manageable in type theory and more accessible from a naive point of view. Hence, we are looking for a more direct type theoretic formulation of weak ω-groupoids. In the present paper we attempt this by defining a weak ω-groupoid to be a globular set with additional structure where this structure is given by interpreting a syntactic theory in the globular set. This approach is new and remains to be proved in our future work that it is equivalent to an established definition. The material presented here has been formalised in Agda [18]. We present it in a natural deduction style for the reader unfamiliar with Agda syntax.

In the following text, we first discuss and define globular sets in type theory in Section 2. In Section 3 we start introducing the syntax of ω-groupoids by defining the general syntactical framework which includes variables contexts, categories, and objects. We also define interpretation of the framework into a globular set. Sections 4, 5 and 6 are concerned with definition of the syntax of objects. Section 4 describes the construction of all units and objects; Section 5 defines the construction of all coherence cells witnessing the left and right unit laws, associativity of composition and interchange. In Section 6, we complete the definition of coherence by introducing all coherence cells between coherence cells. In Section 7 we summarise the results, and provide a rough comparison to other (categorical) approaches to weak ω-categories.

2 Globular Sets

In Type Theory we use the notion of a setoid to describe a set with a specific equality. That is a A : Setoid is given by the following data:

$$\begin{array}{lcl} \mathsf{obj}_A & : & \mathsf{Set} \\ \mathrm{hom}_A & : & \mathsf{obj}_A \to \mathsf{obj}_A \to \mathbf{Prop} \end{array}$$

and proof objects $\mathsf{id}, -^{-1}, -;-$ witnessing that hom is an equivalence relation. Here we write **Prop** to denote the class of sets which have at most one inhabitant. This restriction is important when showing that the category of setoids has certain structure, in particular forms a model of Type Theory. That is setoids can model a type theory with a proof-irrelevant equality. To model proof-relevant equality we need to generalize the notion of a setoid so that the hom-sets are generalized setoids again. It is not enough to just postulate the laws of an equivalence relation at each level, we also need some laws how these proofs interact. On the first level we require the laws of a groupoid, e.g. we want that $\mathsf{id};\alpha$ is equal to α. Here *is*

[1] [15] have shown that the appropriate restriction of Univalence can be eliminated in this setting.

equal means that they are related by the equality relation of a setoid again. As demonstrated in [7, 16], the structure we are looking for is a weak ω-groupoid. It is the goal of this paper to develop a formalisation of this structure. As a first step let's ignore the proof objects (i.e. the data of an equivalence relation and the groupoid laws etc). We end up with a coinductive definition of a *globular set* $G : \mathsf{Glob}$ given by

$$\begin{array}{lcl} \mathsf{obj}_G & : & \mathsf{Set} \\ \mathrm{hom}_G & : & \mathsf{obj}_G \to \mathsf{obj}_G \to \infty\,\mathsf{Glob} \end{array}$$

Here we use ∞ to indicate that Glob is defined coinductively. More formally, Glob is the terminal coalgebra of $\Sigma\mathsf{obj} : \mathsf{Set}.\mathsf{obj} \to \mathsf{obj} \to -$. Given globular sets A, B a morphism $f : \mathsf{Glob}(A, B)$ between them is given by

$$\begin{array}{lcl} \mathrm{obj}_f^{\to} & : & \mathsf{obj}_A \to \mathsf{obj}_B \\ \mathrm{hom}_f^{\to} & : & \Pi a, b : \mathsf{obj}_A.\mathsf{Glob}(\mathrm{hom}_A\ a\, b, \mathrm{hom}_B(\mathrm{obj}_f^{\to}\ a, \mathrm{obj}_f^{\to}\ b)) \end{array}$$

Note that this definition exploits the coinductive character of Glob. Identity and composition can be defined easily by iterating the set-theoretic definitions ad infinitum. As an example we can define the terminal object in $\mathbf{1}_{\mathsf{Glob}} : \mathsf{Glob}$ by the equations

$$\begin{array}{lcl} \mathsf{obj}_{\mathbf{1}_{\mathsf{Glob}}} & = & \mathbf{1}_{\mathsf{Set}} \\ \mathrm{hom}_{\mathbf{1}_{\mathsf{Glob}}}\ x\, y & = & \mathbf{1}_{\mathsf{Glob}} \end{array}$$

More interestingly, the globular set of identity proofs over a given set A, $\mathrm{Id}^\omega\, A : \mathsf{Glob}$ can be defined as follows:

$$\begin{array}{lcl} \mathsf{obj}_{\mathrm{Id}^\omega\, A} & = & A \\ \mathrm{hom}_{\mathrm{Id}^\omega\, A}\ a\, b & = & \mathrm{Id}^\omega\, (a = b) \end{array}$$

Our definition of globular sets is equivalent to the usual one as a presheaf category over the diagram:

$$0 \underset{t_0}{\overset{s_0}{\rightrightarrows}} 1 \underset{t_1}{\overset{s_1}{\rightrightarrows}} 2 \cdots n \underset{t_n}{\overset{s_n}{\rightrightarrows}} (n+1) \cdots$$

with the globular identities:

$$\begin{array}{lcl} s_{i+1} \circ s_i & = & t_{i+1} \circ s_i \\ t_{i+1} \circ t_i & = & s_{i+1} \circ t_i \end{array}$$

In the example of $\mathrm{Id}^\omega\, A$ the presheaf is given by a family $F^A : \mathbb{N} \to \mathsf{Set}$:

$$\begin{array}{lcl} F^A\, 0 & = & A \\ F^A\, 1 & = & \Sigma a, b : A, a = b \\ F^A\, 2 & = & \Sigma a, b : A, \Sigma \alpha, \beta : a = b, \alpha = \beta \\ \vdots & \vdots & \vdots \\ F^A\, (n+1) & = & \Sigma a, b : A, F^{a=b}\, n \end{array}$$

and source and target maps $s_i, t_i : F^A\,(n+1) \to F^A\, n$ satisfying the globular identities.

$$\begin{array}{lcl} s_0(a, b, -) & = & a \\ t_0(a, b, -) & = & b \end{array} \qquad \begin{array}{lcl} s_{n+1}\,(a, b, \alpha) & = & (a, b, s_n\, \alpha) \\ t_{n+1}\,(a, b, \alpha) & = & (a, b, t_n\, \alpha) \end{array}$$

3 Syntax

Our goal is to specify the conditions under which a globular set is a weak ω-groupoid. This means we need to require the existence of certain objects in various object sets within the structure. A natural way would be to generalize the definition of a setoid and add these components to the structure. However, it is not clear how to capture the coherence condition which basically says that any two morphisms which just represent identities in the strict case should be equal. Instead we will follow a different approach which can be compared to the definition of environment models for the λ-calculus: we shall define a syntax for weak ω-groupoids and then define a weak ω-groupoid as a globular set in which this syntax can be interpreted.

3.1 The syntactical framework

We start by presenting a syntactical framework which is a syntax for globular sets. This syntax could be used to identify any globular set with structure (e.g. weak or strict ω-categories), the specific aspects of a weak ω-groupoid will be introduced later by adding additional syntax for objects and other auxiliary syntactic components.

Our framework consists of the following main components which are defined by mutual induction[2]:

Contexts

$$\mathsf{Con} : \mathsf{Set}$$

Contexts serve to formalize the existence of hypothetical objects which are specified by the globular set in which they live. E.g. to formalize ordinary composition we have to assume that objects a, b, c and 1-cells $f : a \longrightarrow b$ and $g : b \longrightarrow c$ exist to be able to form $g \circ f : a \longrightarrow c$.

Categories

$$\frac{\Gamma \ : \ \mathsf{Con}}{\mathsf{VarCat}\ \Gamma \ : \ \mathsf{Set}} \qquad \frac{\Gamma : \mathsf{Con}}{\mathsf{Cat}\ \Gamma : \mathsf{Set}}$$

In order to define the valid compositions of cells one needs to know their boundaries, i.e. iterated domains and codomains in the globular case. Category expressions record this data. We define two kinds of categories: VarCats are categories which contain only variables, while Cats contain all cells freely generated from variables. The set of expressions for both kinds of categories depends on a context, e.g. we need at least to assume that there is one object in the top-level category to be able to form any other categories.

Variables & Objects

$$\frac{G : \mathsf{VarCat}\ \Gamma}{\mathsf{Var}\ G : \mathsf{Set}} \qquad \frac{C : \mathsf{Cat}\ \Gamma}{\mathsf{Obj}\ C : \mathsf{Set}}$$

VarCats contain only variables, which are projections out of the context Γ. On the other hand, given a category we define all expressions which identify objects lying within the category. As indicated above this is the main focus of the forthcoming sections.

---

[2] This is an instance of an inductive-inductive definition in Type Theory, see [3].

We now specify the constructors for the various sets (apart from objects). We use unnamed variables ala deBruijn, hence contexts are basically sequences of categories. However, note that this is a dependent context since the well-formedness of a category expression may depend on the previous context. At the same time we build globular sets from nameless variables in contexts.

$$\frac{}{\varepsilon : \mathsf{Con}} \qquad \frac{G : \mathsf{VarCat}\ \Gamma}{(\Gamma, G) : \mathsf{Con}} \qquad \frac{}{\bullet : \mathsf{VarCat}\ \Gamma} \qquad \frac{G : \mathsf{VarCat}\ \Gamma \quad a, b : \mathsf{Var}\ G}{G[a,b] : \mathsf{VarCat}\ \Gamma}$$

$$\frac{}{\mathsf{vz} : \mathsf{Var}\ (\mathsf{wk}\ G)} \qquad \frac{v : \mathsf{Var}\ G}{\mathsf{vs}\ v : \mathsf{Var}\ (\mathsf{wk}\ G\ G')}$$

where wk is weakening defined for categories by induction on the structure in the obvious way:

$$\frac{G,\ G' : \mathsf{VarCat}\ \Gamma}{\mathsf{wk}\ G\ G' : \mathsf{VarCat}\ (\Gamma, G')} \qquad \begin{aligned} \mathsf{wk}\ \bullet\ G' &= \bullet \\ \mathsf{wk}\ (G[a,b])\ G' &= (\mathsf{wk}\ G\ G')[\mathsf{vs}\ a, \mathsf{vs}\ b] \end{aligned}$$

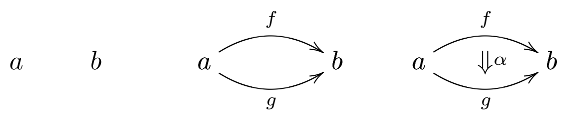
There are two ways to form category expressions: there is the top level category denoted by $\bullet$ and given any two objects a, b in a category C we can form the hom category $C[a,b]$.

$$\frac{}{\bullet : \mathsf{Cat}\ \Gamma} \qquad \frac{C : \mathsf{Cat}\ \Gamma \quad a,\ b : \mathsf{Obj}\ C}{C[\,a\,,\,b\,] : \mathsf{Cat}\ \Gamma}$$

Variables become objects by the following constructor of Obj, which mutually extends to VarCats:

$$\frac{v : \mathsf{Var}\ G}{\mathsf{var}\ v : \mathsf{Obj}\ (\mathsf{var}\ G)} \qquad \frac{G : \mathsf{VarCat}\ \Gamma}{\mathsf{var}\ G : \mathsf{Cat}\ \Gamma} \quad \text{where} \quad \begin{aligned} \mathsf{var}\ \bullet &= \bullet \\ \mathsf{var}\ G[a,b] &= (\mathsf{var}\ G)[\mathsf{var}\ a, \mathsf{var}\ b] \end{aligned}$$

We use the usual arrow notation for categories and objects. For instance, $\bullet[a,b]$, $\bullet[a,b][f,g]$ and $\alpha : \mathsf{Obj}\ (\bullet[a,b][f,g])$ are pictured respectively as follows:



We also write, as usual, $x : a_n \longrightarrow b_n : \cdots : a_0 \longrightarrow b_0$ for an $x : \mathsf{Obj}\ (\bullet[a_0,b_0]\cdots[a_n,b_n])$. Note that it is essential to first introduce VarCats and then Cats with an inclusion

$$\mathsf{var} : \Sigma(\mathsf{VarCat}\ \Gamma)\ \mathsf{Var} \longrightarrow \Sigma(\mathsf{Cat}\ \Gamma)\ \mathsf{Obj}$$

In this way we make sure that variables alone form a globular set, i.e. that the domain and codomain of a variable is a variable. In particular, that it is not possible to introduce a variable between syntactically constructed coherence cells. In this way we can talk about the ω-category freely generated by a globular set.

3.2 Interpretation

Given a globular set we define what we mean by an interpretation of the syntax. Once we have specified all the constructors for objects a weak ω-groupoid is given by such an interpretation. The interpretation of the structural components given in the present section is fixed. Again this is reminiscent of environment models.

An *interpretation* in a globular set G : Glob is given by the following data:

1. An assignment of sets to contexts:

$$\frac{\Gamma : \mathsf{Con}}{[\![\Gamma]\!] : \mathsf{Set}}$$

2. An assignment of globular sets to VarCat and Cat expressions:

$$\frac{G : \mathsf{VarCat}\ \Gamma \qquad \gamma : [\![\Gamma]\!]}{[\![G]\!]\ \gamma : \mathsf{Glob}} \qquad \frac{C : \mathsf{Cat}\ \Gamma \qquad \gamma : [\![\Gamma]\!]}{[\![C]\!]\ \gamma : \mathsf{Glob}}$$

3. An assignment of elements of object sets to object expressions and variables

$$\frac{G : \mathsf{VarCat}\ \Gamma \qquad x : \mathsf{Var}\ G \qquad \gamma : [\![\Gamma]\!]}{[\![x]\!]\ \gamma : \mathsf{obj}_{[\![G]\!]\ \gamma}} \qquad \frac{C : \mathsf{Cat}\ \Gamma \qquad A : \mathsf{Obj}\ C \qquad \gamma : [\![\Gamma]\!]}{[\![A]\!]\ \gamma : \mathsf{obj}_{[\![C]\!]\ \gamma}}$$

subject to the following conditions:

$$\begin{array}{lcl} [\![\varepsilon]\!] & = & 1 \\ [\![\Gamma, G]\!] & = & \Sigma\gamma : [\![\Gamma]\!], [\![G]\!]\ \gamma \\ [\![\bullet]\!]\ \gamma & = & G \\ [\![C[a,b]]\!]\ \gamma & = & \mathsf{hom}_{[\![C]\!]\gamma}\ ([\![a]\!]\ \gamma)\ ([\![b]\!]\ \gamma) \end{array} \qquad \begin{array}{lcl} [\![\mathsf{var}\ x]\!]\ \gamma & = & [\![x]\!]\ \gamma \\ [\![\mathsf{vz}]\!]\ (\gamma, a) & = & a \\ [\![\mathsf{vs}\,x]\!]\ (\gamma, a) & = & [\![x]\!]\ \gamma \end{array},$$

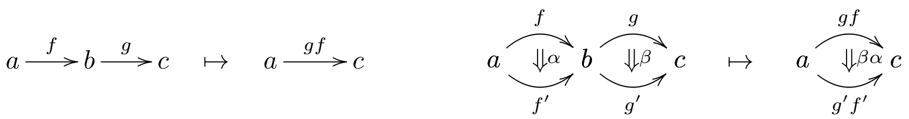
where the last case applies both to VarCats and Cats.

4 Structure

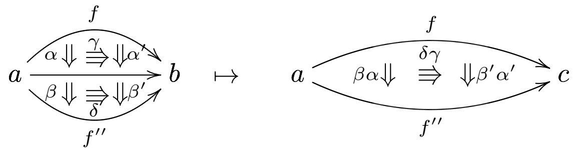
A category, strict or weak, is a globular set with additional structure. The difference between the strict and the weak case is whether we adorn the structure with (equational) constraints or whether one instead of axioms introduces more structure, which witnesses rather than postulates the constraints; so-called *coherence cells*. In this section we introduce the syntax for the structure of composition and units giving rise to syntax for what one could call a *pre-monoidal globular category*, where composites and units are expressible but unconstrained by coherence cells.

4.1 Composition

In the ordinary case, a category, $\mathcal{C}$, defines an indexed operation of composition on its arrows. Explicitly, for a, b, c objects of $\mathcal{C}$, f in $\mathcal{C}[a,b]$, g in $\mathcal{C}[b,c]$, there is a gf in $\mathcal{C}[a,c]$. In the higher-dimensional case, $\mathcal{C}(a,b)$ and $\mathcal{C}(b,c)$ are not mere sets but ω-categories and composition extends from sets the whole hom-categories. Informally: for a, b, c as before, f, g, n-cells of homcategories $\mathcal{C}(a,b)$ and $\mathcal{C}(b,c)$, respectively, one requires an n-cell $g \circ f$ of $\mathcal{C}(a,c)$. The fact that both f and g are of the same relative depth with respect to $\mathcal{C}$ is important, as well the fact the homcategories of f and g meet at a common object, b, of $\mathcal{C}$. Following are some examples of valid compositions for increasing n:



(1)



(2)

We formalise this as follows.

4.1.1 Telescopes

The type $\mathsf{Obj} : \mathsf{Cat} \longrightarrow \mathsf{Set}$ represents the set of syntactical objects lying directly in any category. In order to talk about arbitrary n-cells of a category, for instance to define their compositions, we must introduce *telescopes*. Informally, a telescope is a category in a category. Formally, telescopes, Tel, are defined below at the same time as their *concatenation* onto a category, $+\!\!+$, which takes a telescope to a category, and therefore allows us to put objects into a telescope:

$$\frac{C : \mathsf{Cat}\ \Gamma \qquad n : \mathbb{N}}{\mathsf{Tel}\ C\ n : \mathsf{Set}} \qquad \frac{t : \mathsf{Tel}\ C\ n}{C +\!\!+ t\ :\ \mathsf{Cat}\ \Gamma}$$

Telescopes are like categories except that the base case is an arbitrary category C rather than $\bullet$:

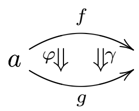
$$\frac{}{\bullet : \mathsf{Tel}\ C\ 0} \qquad \frac{t : \mathsf{Tel}\ C\ n \quad a, b : \mathsf{Obj}\,(C +\!\!+ t)}{t[a,b] : \mathsf{Tel}\ C\ (n+1)} \qquad \begin{array}{lcl} C\ +\!\!+\ \bullet & = & C \\ C\ +\!\!+\ t[a,b] & = & (C +\!\!+ t)[a,b] \end{array}$$

Here, we call n the *length of* t, and we say any $x : \mathsf{Obj}(C +\!\!+ t)$ to be *of depth* n.

We say that t *lies in* C. Note that only the left associative reading of $+\!\!+$ makes sense so expressions like $C +\!\!+ t +\!\!+ u$ are unambiguous.

We say that an object $x : \mathsf{Obj}\ (C +\!\!+ t)$ *lies in (the telescope)* t. When t lies in C, x is called an *object relative to* C. Alternatively, when the category t lies in is not important we use the following syntactical shorthand:

$$\frac{C : \mathsf{Cat}\ \Gamma \quad t : \mathsf{Tel}\ C\ n}{t \Downarrow : \mathsf{Cat}\ \Gamma \qquad t \Downarrow = C +\!\!+ t}$$

For example, given the category  , one has:

$$\bullet[\varphi,\gamma] \Downarrow =\ \bullet[a,b][f,g] +\!\!+ \bullet[\varphi,\gamma]\ =\ \bullet[a,b][f,g][\varphi,\gamma]\ .$$

4.1.2 Back to composition

We use telescopes to define syntax for all compositions of an ω-category. These are defined mutually recursively with their extensions to telescopes:

$$\frac{t : \mathsf{Tel}\ (C[a,b])\ n \qquad u : \mathsf{Tel}\ (C[b,c])\ n}{u \circ t : \mathsf{Tel}\ (C[a,c])} \qquad \frac{\alpha : \mathsf{Obj}(C[a,b] +\!\!+ t) \qquad \beta : \mathsf{Obj}(C[b,c] +\!\!+ u)}{\beta \circ \alpha : \mathsf{Obj}(C[a,c] +\!\!+ (u \circ t))}$$

Where $\circ$ is a new constructor of Obj and $\circ$ for telescopes is a function defined by cases

$$\begin{array}{lcl} \bullet \circ \bullet & = & \bullet \\ u[a',b'] \circ t[a,b] & = & (u \circ t)[a' \circ a, b' \circ b] \end{array}$$

Any α and β as above are said to be *composable*. Note that for a fixed category C, $\circ$ always defines the composition in C, called *horizontal* in the 2-categorical case, which can be applied to all composable $(n+1)$-cells of C, where n is the length of the telescopes t and u. To compose cells "vertically", one moves to a homcategory. In 2-category theory, horizontal composition is usually denoted $\circ$ or $*$ or is left out, whereas vertical composition by $\cdot$. In our case, we always use $\circ$ and the level we mean is contained in the (implicit) parameter C. For example, examples in (1) are both horizontal compositions where $C = \bullet$, while (2) is a vertical composition where $C = \bullet[a,b]$.

4.2 Units

We generate all higher units from a single constructor id defined as follows:

$$\frac{a : \mathsf{Obj}\ C}{\mathsf{id}\ a : \mathsf{Obj}\ C[a, a]}$$

By iteration we obtain the unit for horizontal composition of n-cells:

$$\frac{a : \mathsf{Obj}\ C \qquad n : \mathbb{N}}{\mathsf{idTel}\, a\, n : \mathsf{Tel}\ C\ n \qquad \mathsf{id}^n a : \mathsf{Obj}\ (\mathsf{idTel}\, a\, n \Downarrow)}$$

Again, an iterated unit is defined at the same time as its telescope.

$$\begin{aligned} \mathsf{idTel}\, a\, 0 &= \bullet & \mathsf{id}^0 a &= a \\ \mathsf{idTel}\, a\, (n+1) &= (\mathsf{idTel}\, a\, n)[\mathsf{id}^n a, \mathsf{id}^n a] & \mathsf{id}^{(n+1)} a &= \mathsf{id}\,(\mathsf{id}^n a) \end{aligned}$$

5 Laws

In a strict ω-category composition and identities – structure – are accompanied by axioms expressing their fundamental properties. Namely, composition should be *associative*, and it should satisfy the so-called *interchange law*; identities should be the *units* of composition. The fact that the axioms are equations and one can therefore replace equals for equals in expressions has the pleasant consequence that the complexity of axioms doesn't increase with dimension. Indeed, the whole theory for strict ω-categories can be generalised without much difficulty to categories enriched in an arbitrary monoidal category [11]. However, once the equational axioms are replaced by data – *coherence cells* – their complexity rises steeply with dimension.

The combinatorial complexity of coherence cells has been a major obstacle in the development of weak ω-categories. It has led to the development of many diverse approaches to weak ω-categories, e.g. [20, 6, 5, 21, 19, 14, 16]. Comprehensive surveys and comparisons can be found in [13, 9]. However the development of Type Theory has made it possible to express all coherence cells in a closed form. In this section we start to describe how in Type Theory all coherence cells can be generated by induction on their depth.

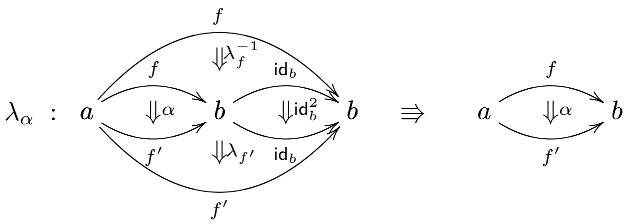
For example, the 1-categorical left-unity law:

$$\mathsf{id}_b \circ f \quad = \quad f\ , \tag{3}$$

for all $f : a \longrightarrow b$, is replaced in a weak ω-category by a pair of 2-cells $\lambda_f : \mathsf{id}_b \circ f \Longrightarrow f$ and $\lambda_f^{-1} : f \Longrightarrow \mathsf{id}_b \circ f$. A similar law should hold for $\circ$ and higher cells. I.e. it should also hold in the strict case that for any $\alpha : f \Longrightarrow f'$:

$$\mathsf{id}_b^2 \circ \alpha \quad = \quad \alpha\ , \tag{4}$$

where $\mathsf{id}_b^2 = \mathsf{id}_{\mathsf{id}_b}$. Note that (4) makes sense because (3) holds. In the weak case, it is not the case that the boundary of $\mathsf{id}_b^2 \circ \alpha$ is equal to the boundary of α and it is simply not possible to categorify (4) by introducing a pair of 3-cells between the left and right side of (4). However, we can use λ_f and λ_f^{-1} to coerce the boundary of the former, $\mathsf{id}_b \circ f$ and $\mathsf{id}_b \circ f'$, to the boundary of the latter, f and f', respectively. The following figure illustrates this idea:



(5)

The reader is invited to try to write down the fourth iteration, i.e. the domain and codomain of λ_γ for $\gamma : \alpha \Rrightarrow \alpha' : f \Longrightarrow' f : a \longrightarrow a'$. Note that each higher pair of λ's can be seen as expressing the naturality of the preceding lower lambda.

Similarly one must introduce ρ's to witness the right unit law, χ's to witness interchange, and α's to witness associativity. In the case of groupoids where every arrow has an inverse there are ι's and κ's to witness the left and right cancellation properties. The example of λ has been chosen because of its relative simplicity.

Moreover, all such coherence cells must satisfy a coherence property basically saying that any pair of n-cells from d to d' involving only coherence cells and units[3] must have a mediating n+1-cell connecting d and d'. Intuitively, as the coherence cells λ, ρ, α and χ we have just described witness axioms, the higher coherence cells witness their closure under composition and identity.

5.1 Formalising left units

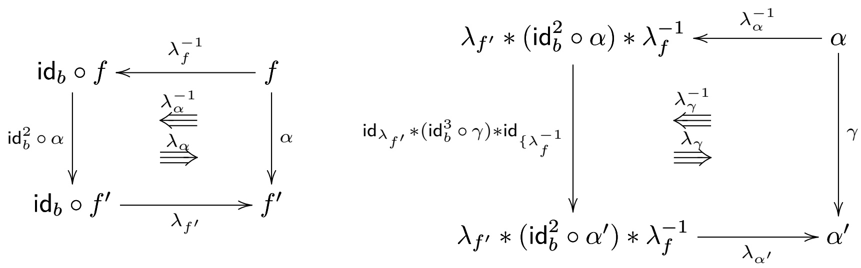
In (5) we made the boundaries of the left- and right-hand sides match by applying the function:

$$\Phi \quad \equiv \quad (l, l') \quad \mapsto \quad x \quad \mapsto \quad l' \cdot x \cdot l$$

to $(\lambda_f^{-1}, \lambda_{f'})$ and $\mathsf{id}_b^2 \circ \alpha$. The 3-cells λ_α and λ_α^{-1} are then introduced as

$$\begin{array}{lll} \lambda_\alpha & : & \mathsf{Obj}\ (\bullet[a,b][f,f'][\Phi\,(\lambda_f^{-1},\lambda_{f'})\,(\mathsf{id}_b^2 \circ \alpha), \alpha]) \\ \lambda_\alpha^{-1} & : & \mathsf{Obj}\ (\bullet[a,b][f,f'][\alpha, \Phi\,(\lambda_f^{-1},\lambda_{f'})\,(\mathsf{id}_b^2 \circ \alpha)]) \end{array} \quad .$$

These arrows should be natural in 3-cells $\gamma : \alpha \Rrightarrow \alpha'$. In a diagram:



Note that going top-left-bottom around the square one gets

$$\Phi\,(\lambda_\alpha^{-1}, \lambda_{\alpha'})\,(\Phi\,(\lambda_f^{-1}, \lambda_{f'})\,\gamma)) \quad .$$

This is the basic idea of the recursion generating all higher λ's. A similar pattern occurs in the definition of the other coherence cells.

[3] Identity cells can be seen as coherence cells witnessing reflexivity of equality.

5.2 Formalising all coherence cells

To summarise and generalise, we want to introduce for each α in a telescope t of length n and each β in a telescope u of length n a cell $\Phi\, m\, \alpha \longrightarrow \beta$ where m is the data necessary to define a function $\mathsf{Obj}\ (t \Downarrow) \to \mathsf{Obj}\ (u \Downarrow)$. We will call such an m a *telescope morphism from t to u*; formally $m : t \rightrightarrows u$. Then Φ has type $t \rightrightarrows u \to \mathsf{Obj}(t \Downarrow) \to \mathsf{Obj}(u \Downarrow)$. Formally, we define telescope morphisms as follows; in mutual recursion with their application to telescopes and objects in telescopes:

$$\frac{t, u : \mathsf{Tel}\, C\, n}{t \rightrightarrows u : \mathsf{Set}} \qquad \frac{}{\bullet : \bullet \rightrightarrows \bullet} \qquad \frac{m : t \rightrightarrows u \quad \alpha : \mathsf{Obj}\,(u \Downarrow [a', m@a]) \quad \mathsf{Obj}\,(u \Downarrow [m@b, b'])}{m[\alpha, \beta] : t[a, b] \rightrightarrows u[a', b']}$$

where

$$\frac{m : t \rightrightarrows u \quad t' : \mathsf{Tel}\,(t \Downarrow)\, n}{m \overrightarrow{@} t' : \mathsf{Tel}\,(u \Downarrow)\, n} \qquad \frac{m : t \rightrightarrows u \quad a : \mathsf{Obj}\,(t \Downarrow +\!\!+\, t')}{m@a : \mathsf{Obj}\,(u \Downarrow +\!\!+\, m \overrightarrow{@} t')}$$

$$\begin{aligned} \bullet \overrightarrow{@} t &= t \\ m'[\alpha, \beta] \overrightarrow{@} t &= (m' \overrightarrow{@} t)[m@\alpha, m@\beta]) \end{aligned}$$

Φ is then a special case of @ for $t' = \bullet$. To define @ we need the following auxiliary function, among others, which extends a telescope on the left.

$$\frac{t : \mathsf{Tel}\ (C[a, b])\ n}{[a, b]t : \mathsf{Tel}\ C\ (n + 1)} \quad \text{where} \quad \begin{aligned} [a, b]\bullet &= \bullet[a, b] \\ [a, b](t[c, d]) &= ([a, b]t)[c, d] \end{aligned}$$

Note that here c and d don't actually fit into the telescope $[a, b]t$ because the latter is definitionally different from t. However, it is straightforward to prove by induction that

$$t \Downarrow \;\equiv\; [a, b]t \Downarrow\ , \tag{6}$$

and use the proof to make c and d fit. However, in the interest of clarity we left the details out above. The full details can be found in [4].

We are now in the position do define @ as follows: The base case is trivial:

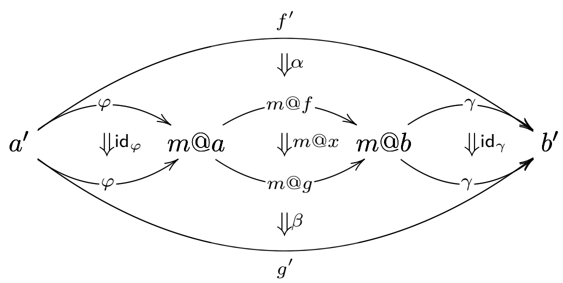
$$\bullet@x = x$$

The hom-case follows the pattern outlined in Section 5.1.

$$\frac{m'[\alpha, \beta] : t[a, b] \rightrightarrows u[a', b'] \qquad t' : \mathsf{Tel}\ (t[a, b] \Downarrow)\ n \qquad x : \mathsf{Obj}\ (t' \Downarrow)}{m'[\alpha, \beta]@x \;=\; \mathsf{id}^n \beta \circ (m'@x) \circ \mathsf{id}^n \alpha}$$

In summary, $m@x$ is defined by induction on m where in each step the length of m decreases by one and the depth of x increases by one. To make the levels match the category of x has to be whiskered by the morphisms α, β for $m = m'[\alpha, \beta]$. When $m = \bullet$, the recursion stops. The meticulous reader will have noticed that the expression $m'@x$ above is not well typed as x lives in $t[a, b] +\!\!+\, t'$ and we need an object in $t +\!\!+\, [a, b]t'$. But this is easily fixed by substituting using (6). Other similar inaccuracies are dealt with similarly.

Here is an illustration for $m = \bullet[\varphi, \gamma][\alpha, \beta]$, $t' = \bullet$, $t = \bullet[a, b][f, g]$, $u = \bullet[a', b'][f', g']$:



Having defined telescope morphisms, it is relatively easy to define λ's of all depths relative to an arbitrary category. All that is needed is a telescope morphism, $\overrightarrow{\lambda}$, together with a new constructor, λ, of Obj:

$$\frac{t : \mathsf{Tel}\ C[a,b]\ n}{\overrightarrow{\lambda}\, t : (\mathsf{idTel}\,(\,\mathsf{id}\, b)\, n) \circ t \rightrightarrows t} \qquad \frac{t : \mathsf{Tel}\ C[a,b]\, n \quad f : \mathsf{Obj}(t \Downarrow)}{\lambda\, t\, f : \mathsf{Obj}\,((t \Downarrow)[\overrightarrow{\lambda}\, t @(\mathsf{id}^n b \circ f), f])}$$

where

$$\begin{array}{lcl} \overrightarrow{\lambda}\, \bullet & = & \bullet \\ \overrightarrow{\lambda}\,(t'[a,b]) & = & (\overrightarrow{\lambda}\, t')[(\lambda\, t'\, a)^{-1}, \lambda\, t'\, b] \end{array}$$

Here we could define a pair of constructors λ and λ^{-1} for the two opposite directions of λ. Instead, as we are interested in groupoids, we define a generic constructor $^{-1}$ on all cells of a homcategory:

$$\frac{f : \mathsf{Obj}\ (C[a,b])}{f^{-1} : \mathsf{Obj}\,(C[b,a])}$$

The introduction of formal inverses forces the introduction of coherence cells witnessing their being left and right inverses. See the next section.

5.3 Right units, associativity and interchange

Similarly to λ's we define the remaining coherence cells, i.e. ρ's to witness right units, α's to witness associativity of composition, ι's and κ's to witness inverses and χ's to witness interchange. These are defined analogically to λ's.

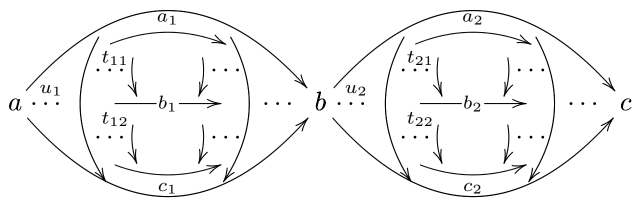
To this end, note that everything in the definition of λ is forced by the type of $\overrightarrow{\lambda}$. In general it is enough to give for ρ and α the type of the telescope morphism. Just as in the case of λ, it is in each case just a "telescopisation" of the ordinary case.

$$\frac{t : \mathsf{Tel}\, C[a,b]\, n}{\overrightarrow{\rho}\, t : t \circ (\mathsf{idTel}\ \ (\mathsf{id}\, a)\, n) \rightrightarrows t} \qquad \frac{t : \mathsf{Tel}\ C[a,b]\ m \quad u : \mathsf{Tel}\ C[b,c]\ n \quad v : \mathsf{Tel}\ C[c,d]\ o}{\overrightarrow{\alpha}\, t\, u\, v : (v \circ u) \circ t \rightrightarrows v \circ (u \circ t)}$$

Because of the way we introduce identities, the laws of inverses are also simple:

$$\frac{f : \mathsf{Obj}\ (C[a,b])}{\iota\ f : \mathsf{Obj}\ (C[a,a][f^{-1} \circ f, \mathsf{id}\ a]) \qquad \kappa\ f : \mathsf{Obj}(C[b,b][f \circ f^{-1}, \mathsf{id}\ b])}$$

The coherence cells witnessing interchange, $\overrightarrow{\chi}$ are in the ω case is more complicated. In the simple 2-categorical case, the interchange law states that $(\gamma' \cdot \gamma) * (\varphi' \cdot \varphi) \ = \ (\gamma' * \varphi') \cdot (\gamma * \varphi)$. In the ω-case the law remains syntactically the same but we consider each of φ, φ', γ and γ' in their telescopes with the generalised notion of composability. The following picture illustrates the idea:



Where $\cdots$ indicate telescopes of arbitrary depth where u_1 and u_2 have to be of the same length; and $t_{ij},\ i,j \in \{1,2\}$ have to be of the same length. In this situation, it is possible, up to definitional equality of telescopes, to form both the composition $(t_{22} \circ t_{21}) \circ (t_{12} \circ t_{11})$ and also $(t_{22} \circ t_{12}) \circ (t_{21} \circ t_{11})$. A telescope morphism from the former to the latter telescope induces a coherence cell for interchange. This is formalised as follows:

$$\frac{u_1 : \mathsf{Tel}\ \mathcal{C}[a,b]\ n \quad u_2 : \mathsf{Tel}\ \mathcal{C}[b,c]\ n \quad t_{11} : \mathsf{Tel}\ (C[a,b] +\!\!+\, u_1)\ m \quad t_{12} : \mathsf{Tel}\ (C[a,b] +\!\!+\, u_1)\ m \quad t_{21} : \mathsf{Tel}\ (C[b,c] +\!\!+\, u_2)\ m \quad t_{22} : \mathsf{Tel}\ (C[b,c] +\!\!+\, u_2)\ m}{\chi\, t_{11}\, t_{12}\, t_{21}\, t_{22}\ :\ (t_{22} \circ t_{21}) \circ (t_{12} \circ t_{11}) \rightrightarrows (t_{22} \circ t_{12}) \circ (t_{21} \circ t_{11})}$$

6 Coherence

6.1 The need of more coherence

In the previous sections we showed how to define all coherence cells witnessing the axioms of a strict ω-groupoid. These can be composed to witnesses identity of cells which don't exactly match the sides of either of the axioms.

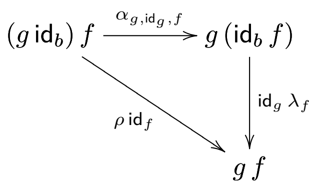
For instance, to witness the equality

$$(g\ \mathsf{id}_b)\ f \quad = \quad g\ f$$

we can compose λ after α to obtain witnesses such as:

$$\mathsf{id}_g \lambda_f \cdot \alpha_{g,\mathsf{id}_g,f}\ :\ (g\ \mathsf{id}_b)\ f \longrightarrow g\ f \qquad \text{and} \qquad \rho\,\mathsf{id}_f\ :\ (g\ \mathsf{id}_b)\ f \longrightarrow g\ f \quad .$$

In the strict case equality of 1-cells is a proposition and therefore all proofs of equality of 1-cells are equal. In the weak 2-categorical case, equality of 1-cells is not propositional but equality of 2-cells is. In other words, equality of 1-cells is witnessed by 2-cells which must satisfy new axioms. For instance:

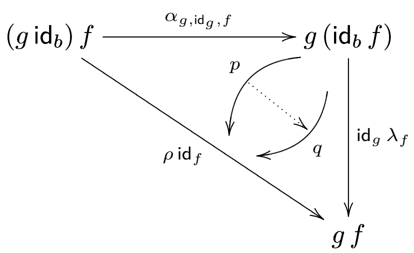


(7)

In our case, when all levels are weakened the equality is replaced be a new 3-cell

$$\mathsf{id}_g \lambda_f \cdot \alpha_{g,\mathsf{id}_g,f} \longrightarrow \rho\,\mathsf{id}_f$$

Moreover, for any pair of such 3-cells, p, q : $\mathsf{id}_g \lambda_f \cdot \alpha_{g,\mathsf{id}_g,f} \longrightarrow \rho\,\mathsf{id}_f$, there must be 4-cell $p \longrightarrow q$, etc., all the way up to ω. This is a weakening of propositionality of equality of n-cells in the strict setting. The following diagram illustrates this up to level 4:



(8)

In summary and full generality:

For any pair of coherence cells with the same domain and codomain, there must be a mediating coherence cell.

6.2 Formalising coherence cells between coherence cells

We formalise the above principle as follows. We define a predicate thin on objects such that $\mathsf{thin}\,(\mathsf{id}\,f) = \top$ where $\top$ is the one element Set. Moreover each coherence cell is thin

$$\mathsf{thin}\,(\lambda\,\_\,\_) = \top \quad \mathsf{thin}(\rho\,\_\,\_) \ = \ \top \quad \mathsf{thin}\,(\alpha,\_\,\_\,\_) = \top \quad \mathsf{thin}\ (\chi\,\_\,\_\,\_\,\_) = \top$$
$$\mathsf{thin}\,(\iota\ f) = \top \quad \mathsf{thin}\,(\kappa\ f) = \top$$

Further we close thin under weakening, composition and inverses. Finally, we introduce a constructor of Obj and a clause of thin:

$$\frac{f\ \ g : \mathsf{Obj}\ C[a,b] \qquad p : \mathsf{thin}\ f \qquad q : \mathsf{thin}\ g}{\mathsf{coh}\ p\ q : \mathsf{Obj}\ C[a,b][f,g]} \qquad \mathsf{thin}\,(\mathsf{coh}\,p\,q) = \top$$

The last remaining case are variables, which are not thin:

$$\mathsf{thin}\,(\mathsf{var}\,v) = \bot$$

6.3 The problem with coherence

The question of coherence in weak ω-categories is subtle. On the one hand, one needs enough coherence to make weakly equivalent all cells that should be identities in the strict case. On the other hand, if we shouldn't add too many equations not to loose any of our intended models. We believe our definition is correct as we are saying in the definition of coh only that all *formal* diagrams of coherence cells commute (see [17], §VII.2, for a related discussion). The fact that not all diagrams commute rests on the fact that one is not allowed to postulate equations between nonvariables, which is achieved by the separation of VarCat and Cat and the fact that contexts are built from VarCats. Otherwise it would be for instance possible[4] to assume a 0-cell a, and a pair of 2-cells: $x, y : \mathsf{id}\ a \to \mathsf{id}\ a : a \to a$. Then it is be possible to construct, by the Eckerman-Hilton argument, a cell $\eta : x \circ y \to y \circ x : \mathsf{id}\ a \to \mathsf{id}\ a : a \to a$ which is thin. As $\mathsf{id}\ (x \circ y)$ is also thin, $\eta \circ \eta$ and $\mathsf{id}(x \circ y)$ would be equivalent by coh. However, not every monoidal braided category is symmetrical. However it is not possible to construct this example in our syntactical framework as $(\mathsf{id}\ a)$ is not a variable.

7 Conclusions and Further Work

7.1 Summary

We have presented a novel approach to defining weak ω-groupoids which is based on ideas from Type Theory. The central idea is to define the syntax of weak ω-groupoids and then define a weak ω-groupoid as a globular set with an interpretation of the syntax, which is where Type Theory has its greatest strength. Our approach to formalization of coherence is natural, in a way naive, since it is a natural generalisation of the corresponding first order laws.

We have formalized all of the material presented here in Agda [18], except for the definition of the telescope morphism for χ. We believe this is a technicality, albeit a difficult one. The Agda source file is available from [4].

[4] We are indebted to an anonymous referee for pointing out this example to us.

7.2 Related work

There exists an abundance of categorical definitions of weak ω-categories and groupoids. Although a direct comparison is a slippery road, we would like to give a rough comparison of the key similarities and differences between our and other definitions. Most importantly, our definition is fundamentally different in that it is formulated in Type Theory rather than Category Theory. This means that we couldn't just formalise any of the approaches [19, 6, 14] because the notion a strict ω-category is central in them in that it drives the definition of coherence cells. However, it is unclear to how define *strict* ω-categories in Type Theory without quotient types. This forces some of the choices we have made. Nevertheless, on an intuitive level there are similarities of our approach to some categorical approaches, in particular to Batanin's definition [6] which we briefly discuss below:

- Batanin's *spans* are essentially our telescopes. And as noted by Batanin, Cartier called Batanin's spans telescopes in 1994.
- Batanin's definition, same as ours, is globular and works with a system of units and binary compositions.
- We conjecture that for a $C : \mathsf{Cat}\ \Gamma$, the mapping $n \mapsto \Sigma(t : \mathsf{Tel}\ C\ n)(\mathsf{Obj}(t \Downarrow))$ forms a strict monoidal globular category freely generated by the globular set determined by Γ.
- It remains to show that our syntax defines a contractible ω-operad and our notion of interpretation defines its algebra.

7.3 Further work

The current formalisation is still quite complicated and we hope to find ways to simplify it. One interesting idea may be to use the syntactical approach to define opetopes based on dependent polynomial functors (i.e. indexed containers) [12], which has a very type-theoretic flavour.

It remains to prove that the definition proposed in this paper is a sensible one. This seems to be most easily done by showing that any interpretation of the syntax defines an algebra for Batanin's universal contractible operad.

We would like to use our framework to provide a formalisation of a variation of the results in [16, 7] by showing that Id^{ω} is a weak ω-groupoid. Such a formalisation would be different form their results because we are working inside Type Theory, rather than on a meta-level.

The main challenge ahead is to formalize the notion of a ω-groupoid model of Type Theory. Once this has been done we will be able to eliminate the univalence axiom and provide a computational interpretation of this principle.

Acknowledgments

We would like to thank Peter Lumsdaine and Darin Morrison who contributed to initial attempts to formalize weak ω-groupoids in Agda. We would like to acknowledge interesting and useful discussions on topics related to this paper with Steve Awodey, Thierry Coquand, Robert Harper, Kris Kapulkin, Nicolai Krauss, Dan Licata, Thomas Streicher and Vladimir Voevodsky. We would also like to thank the annonymous referees for providing valuable comments and indeed pointing out a shortcoming with our previous definition of coherence.

References

1 Thorsten Altenkirch. Extensional equality in intensional type theory. In *14th Symposium on Logic in Computer Science*, pages 412 – 420, 1999.

2 Thorsten Altenkirch, Conor McBride, and Wouter Swierstra. Observational equality, now! In *PLPV '07: Proceedings of the 2007 workshop on Programming languages meets program verification*, pages 57–68, New York, NY, USA, 2007. ACM.

3 Thorsten Altenkirch, Peter Morris, Fredrik Nordvall Forsberg, and Anton Setzer. A categorical semantics for inductive-inductive definitions. In Andrea Corradini, Bartek Klin, and Corina Cîrstea, editors, *Algebra and Coalgebra in Computer Science*, volume 6859 of *Lecture Notes in Computer Science*, pages 70–84. Springer Berlin / Heidelberg, 2011.

4 Thorsten Altenkirch and Ondřej Rypáček. A syntactical approach to weak ω-groupoids: Agda implementation. `https://github.com/txa/OmegaCats/blob/master/Syntactical/WeakOmegaCat/Core.agda`.

5 J. Baez and J. Dolan. Higher-dimensional algebra iii. n-categories and the algebra of opetopes. *Advances in Mathematics*, 135(2):145–206, 1998.

6 Michael Batanin. Monoidal globular categories as natural environment for the theory of weak n-categories. *Advances in Mathematics*, 136:39–103, 1998.

7 Benno Van Den Berg and Richard Garner. Types are weak ω-groupoids, 2008.

8 James Chapman, Pierre-Évariste Dagand, Conor McBride, and Peter Morris. The gentle art of levitation. *SIGPLAN Not.*, 45:3–14, September 2010.

9 Eugenia Cheng and Aaron Lauda. Higher-dimensional categories: an illustrated guide book.

10 Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory (Venice, 1995)*, volume 36 of *Oxford Logic Guides*, pages 83–111. Oxford Univ. Press, New York, 1998.

11 G. M. Kelly. *Basic Concepts of Enriched Category Theory*, volume 64 of *Lecture Notes in Mathematics*. Cambridge University Press, 1982.

12 Joachim Kock, André Joyal, Michael Batanin, and Jean-François Mascari. Polynomial functors and opetopes. *Advances in Mathematics*, 224(6):2690 – 2737, 2010.

13 T. Leinster. A survey of definitions of n-category. *Th. Appl. Cat. 10*, 10:1–70, 2002.

14 Tom Leinster. *Operads in higher-dimensional category theory*. PhD thesis, University of Cambridge, Cambridge, 2000.

15 Daniel R. Licata and Robert Harper. 2-dimensional directed type theory. *Electr. Notes Theor. Comput. Sci.*, 276:263–289, 2011.

16 Peter Lefanu Lumsdaine. Weak ω-categories from intensional type theory. *Logical Methods in Computer Science*, 6:1–19, 2010.

17 Saunders MacLane. *Categories for the Working Mathematician*. Springer.

18 Ulf Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Chalmers University of Technology, 2007.

19 Jacques Penon. Approche polygraphique des ∞-categories non strictes. *Cahiers de Topologie et G´eom´etrie Diff´erentielle*, 40(1):31–80, 1999.

20 Ross Street. The algebra of oriented simplexes. *Journal of Pure and Applied Algebra*, 49(3):283 – 335, 1987.

21 Todd Trimble. What are 'fundamental n-groupoids'? Cambridge, August 1999. seminar at DPMMS.

22 Vladimir Voevodsky. Univalent foundations of mathematics. In Lev Beklemishev and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 6642 of *Lecture Notes in Computer Science*, pages 4–4. Springer Berlin / Heidelberg, 2011.