Higher-categorical Strucures and Type Theories: DRAFT!

Peter LeFanu Lumsdaine

July 2010

Carnegie Mellon University

TODO: Title Here!

Peter LeFanu Lumsdaine

July 2010

Committee

Steven M. Awodey (advisor)
someone
someone else
some more

Contents

CHAPTER 1

Introduction and background

1. Homotopy Type Theory: an introduction

1.1. Similar to the introduction to my previous paper: a quick, accessible intro to the higher-categorical view of identity types.

1.2. Refer to appendices for full background on globular higher cats & on DTT. However: include a *rough* introduction to the higher cats, & a *full* introduction to (& discussion of) identity types.

2. Survey of the field

2.1. Goals! What we're working towards in the short-term (eg sound and complete semantics, good analysis of categorical properties of Π, Σ-types, etc.

2.2. What's actually been done! Some models, a few structures, syntactic analysis in dimension 2, applications as independence results...

Lots of references should go in this, of course!

3. Outline of the present work

3.1. Overall structure (composition of 2-cells along bounding 0-cell), reason therefor (aim of analysing type theories in the well-understood quasi-categorical setting).

3.2. Overview of the "universal-algebraic aspects" setup: technical, dry, but necessary!

3.3. Results of the "syntactic structures" section.

3.4. Results of the "homotopical constructions" section.

4. Outlook: visions of a higher-categorical foundation

4.1. Write up some of what's currently just in folklore, the n-lab, the categories list, boozy nights out with the gang, etc. :

Voevodsky's model(s) + axiom; the type theorists' OTT etc.; notions of "the same"; "category theory without equality", etc.

5. Acknowledgements

(Should go before this chapter, or here at end of it?)

— Steve! Krzys/Chris. Other HTT'ers: Michael, Richard, Benno, Chris. Pittsburgh PL crowd: Bob H, Dan L, Noam Z. Also in Pittsburgh: Kohei, Henrik, James C, Rick S, Peter A, Dana. Chicago group: Mike, Emily, Claire, Daniel. Nottingham: Thorsten and his merry men. Elsewhere on-topic: Martin H, Andrej, Pierre-Louis C., Paul-Andr M?, Thomas F?. Off-topic: Yimu, orchestras, parents!

(To do: ask people's permission for this??)

CHAPTER 2

Algebraic structures from dependent type theory

Possibly fold this chapter into the next??

1. A category of Type Theories

Give brief, semi-formal outline of the type theory, referring to Appendix

In fact, we will not work directly with the category of type theories, but with an equivalent category of algebraic models. but The theory of such models is attractive, but suffers from rather an embarassment of riches: for instance [**Jac93**], [**Pit00**], [**Hof97**] and [**Dyb96**] each [TODO: chronologicise these] define slightly different notions of categorical models (several after the unpublished [**?**]), all equivalent (in some sense) to each other and to syntactically presented dependent type theories. (TODO: also look up "categories with families".)

Of course, these notions each have advantages and disadvantages: some are more elementary to present; some are more categorically elegant; some are more easily adaptable to extensions of the type theory... We thus take this opportunity to survey several of the various options, and the comparisons between them.

(TODO: reorganise/tidy all the below!

OK: new outline! (See notes of 23.vii for details.)

(1) sketch definition of the type theory, referring to appendix for full details
(2) categories with attributes
(3) (full split) comprehension categories
(4) equivalence of these
(5) based, stratified versions
(6) adjunction between syntactic theories and stratified CwA's.
(7) dependent contexts construction; slice construction.
(8) connections between plain, based, stratified, accessible CwA's.
(9) nice big figure!

Definition 1.1. A *split full comprehension category*[**Jac93**] (FSCC) is a category $\mathcal{C}$ together with a split fibration $p : \mathcal{E} \longrightarrow \mathcal{C}$ and a factorisation

$$p = \mathrm{cod} \cdot \mathcal{P} : \mathcal{E} \longrightarrow \mathcal{C}^{\rightarrow} \longrightarrow \mathcal{C}$$

such that (a) $\mathcal{P}$ maps cartesian arrows to pullback squares, and (b) $\mathcal{P}$ is full and faithful. By abuse of notation, we will often refer to $\mathcal{C}$ itself as the comprehension category; the pair $(p, \mathcal{P})$ is called a *comprehension structure* on $\mathcal{E}$.

A (strict) map of FSCC's is just a functor $F : \mathcal{C} \longrightarrow \mathcal{C}'$ and a map of fibrations $p \longrightarrow p'$ over F, commuting with the factorisations $\mathcal{P}$, $\mathcal{P}'$. A *map of comprehension structures* on $\mathcal{C}$ is just the case $F = 1_C$.

$$\begin{tikzcd}
\mathbf{FSCC} \arrow[r, phantom, "\simeq"] \arrow[d, bend right] \arrow[d, phantom, "\dashv"] & \mathbf{CwA} \arrow[d, bend right] \arrow[d, phantom, "\dashv"] & \\
\mathbf{FSCC}_\diamond \arrow[r, phantom, "\simeq"] \arrow[u, bend right] \arrow[d, bend left] \arrow[d, phantom, "\dashv"] & \mathbf{CwA}_\diamond \arrow[u, bend right] \arrow[d, bend left] \arrow[d, phantom, "\dashv"] & \\
\mathbf{FSCC}_{\mathbf{strat}} \arrow[r, phantom, "\simeq"] \arrow[u, hook, bend left] & \mathbf{CwA}_{\mathbf{strat}} \arrow[r, phantom, "\simeq"] \arrow[u, hook, bend left] \arrow[dr, bend left] \arrow[dr, phantom, "\vdash"] & \mathbf{SynThy} \arrow[d, bend left] \arrow[d, phantom, "\dashv"] \\
& & \mathbf{SynPres} \arrow[ul, bend left] \arrow[u, bend left]
\end{tikzcd}$$

If $\mathcal{C}$ has all pullbacks, then condition (a) just says that $\mathcal{P}$ is a map of fibrations.

In fact, for fixed $\mathcal{C}$, $\mathbf{FSCS}(\mathcal{C})$ is (equivalent to) a presheaf category—specifically, to the slice of $\hat{\mathcal{C}}$ over the presheaf $\mathrm{cod}^{\mathrm{spl}}$ in which an element of $P(A)$ is a map $f : B \longrightarrow A$ together with chosen pullbacks along all maps $j : A' \longrightarrow A$. (Explain how??) comprehension categories will frequently be presented in this form.

Example 1.2. For any type theory T, its category of context $\mathcal{C}(\mathsf{T})$ is a comprehension category, in which objects of $p(\Gamma)$ are types dependent over Γ, and $\mathcal{P}$ sends a type $A \in p(\Gamma)$ to the dependent projection $\Gamma, x : A \longrightarrow \Gamma$.

In fact, this construction is part of an equivalence between **Th** and a certain full co-reflexive subcategory of **FSCC**: see Appendix **??** for details. In light of this, we will typically refer to the objects of any FSCC as contexts, and the objects of the fibration as dependent types.

Categories with attributes. From [**Pit00**], [**Hof97**], [**Dyb96**], under various names. Give definitions; (in notation to match that used above); give equivalence (honest 1-equivalence) with comprehension categories.

Stratification. Define *stratified* cwa, & map of; note that it's (perhaps unexpectedly) a *full subcategory*(!) of cwa's, closed under connected limits and colimits. (NB: this deinition seems to be new (though comparable to Cartmell); have I missed something?)

Proposition: there is an *honest 1-equivalence* between stratified cwa's and dependent algebraic theories as laid out in appendix.

Note: since morphisms of DAT's are most easily defined by transfer from cwa's, the content of this proposition is really just that there are maps of *objects* from stratified cwa's to dat's and back, with an isomorphism $\mathcal{C} \cong FG(\mathbb{G})$.

Define *reachable* cwa's after Pitts.

Theorem: there is a *2-equivalence* between reachable cwa's and stratified cwa's.

Discuss significance of 1- versus 2-equivalence: latter gives equivalence of "type theories", which is fine on the categorical side, but not on the syntactic side: we care about the difference between *isomorphism* and *equivalence* there as syntactic presentations of theories are *0-categorical objects* (Voevodsky slogan!).

Also point out: type theory with *context and their maps and equalities as primitive judgements* (hence allowing equalities and maps between contexts of different lengths) should correspond to general cwa's.

Constructors. This all extends to theories with type constructors.

Define: a lax map of comprehension structures (do I mean colax??)

Point out: the 2-category $\mathbf{FSCS}(\mathcal{C})_{\mathrm{lax}}$ has finite products. (And these are probably better seen as 2-limits in $\mathbf{FSCS}(\mathcal{C})$.)

Definition 1.3. An FSCS *with units* on a category $\mathcal{C}$ is an FSCS $(p, \mathcal{P}_0)$ together with a strict map of FSCS's $1 : (1_B, id) \longrightarrow (p, \mathcal{P}_0)$.

An FSCS *with binary products* on $\mathcal{C}$ is an FSCS $(p, \mathcal{P}_0)$ with a strict map $\times$: $(ptimes_B p, \mathcal{P}_0 \times_B) \longrightarrow (p, \mathcal{P}_0)$.

Note: this implies a certain adjunction, so does corresponds to Jacobs' "fscc with units". [Show this??]

To introduce the structure corresponding to identity types, we will need a little more terminology.

Definition 1.4 (Dependent contexts)**.** Given any comprehension category $(\mathcal{C}, p, \mathcal{P})$, we may construct another comprehension structure $(p^{\mathsf{cxt}}, \mathcal{P}^{\mathsf{cxt}})$ on $\mathcal{C}$:

An object of $p^{\mathsf{cxt}}[\Gamma]$ is a list $A_1, \ldots, A_n$, where $A_i \in p(\Gamma.A_1 \ldots .A_{i-1})$, for each $i \leq n$; context extension is defined by $\Gamma.(A_1 \ldots, A_n) = \Gamma.A_1 \ldots .A_n$, and pullback $f^* : p^{\mathsf{cxt}}(\Gamma) \longrightarrow p^{\mathsf{cxt}}(\Delta)$ is similarly defined in terms of pullback in p.

This is the object part of an evident functor $(-)^{\mathsf{cxt}}$ on $\mathbf{FSCS}(\mathcal{C})$.

The $(-)^{\mathsf{cxt}}$-construction has a natural type-theoretic interpretation: if $(\mathcal{C}, p, \mathcal{P})$ was obtained from a type theory, then for any Γ, $p^{\mathsf{cxt}}(\Gamma)$ is (isomorphic to) the category of dependent contexts over Γ and dependent context morphisms between them.

Morevoer, $(-)^{\mathsf{cxt}}$ has a natural monad structure, and indeed is the "free monoid" monad for a certain monoidal structure on $\mathbf{FSCS}(\mathcal{C})$; and all this is natural in $\mathcal{C}$, giving a total monad $(-)^{\mathsf{cxt}}$ on **FSCC** over **Cat**. However, these aspects will not concern us further.

(If I included a discussion of "stratified comp cats" earlier, mention how this construction naturally takes us outside of them unless we soup it up; and how the souped-up version gives a "strong Σ-types" monad; but why it *doesn't* at the moment.)

Definition 1.5. A *dependent projection* in $(\mathcal{C}, p, \mathcal{P})$ is a map isomorphic to one of the form $\Gamma.\Delta \longrightarrow \Gamma$, for some $\Delta \in p^{\mathsf{cxt}}(\Gamma)$. Graphically, dependent projections will be distinguished as $\Gamma' \twoheadrightarrow \Gamma$.

Note that any composite of dependent projections is again one. [TODO: is this clear? or draw the diagram for it? see notes, 16.vii] Moreover, pulbacks of dependent projections along arbitrary maps exist, and are again dependent projections.

[TODO: in some ways, better to take dependent projection to be extra structure, a *chosen*isomorphism and a *chosen* $\Delta \in p^{\mathsf{cxt}}(\Gamma)$. The distinction doesn't seem to seriously affect anything I do; make it, perhaps?]

The nice slice.

1.6. $(\mathcal{C}, p^{\mathsf{cxt}}, \mathcal{P}^{\mathsf{cxt}})$ can never be stratified: for any choice of $\diamond$, the empty dependent context $() \in p^{\mathsf{cxt}}(\diamond)$ satisfies
diamond.$() = \diamond$, so lengths cannot be consistently assigned.

However, for any $(\mathcal{C}, p, \mathcal{P})$ and $\Gamma \in \mathcal{C}$, there is an evident stratified attributes structure on $p^{\mathsf{cxt}}(\Gamma)$; the resulting cwa may be called the *nice slice* $(\mathcal{C}, p, \mathcal{P})/\Gamma$, and is the algebraic analogue of working in context Γ.

2. Type constructors

Thus far, we have considered only *algebraic* theories, with just a fixed, given set of (dependent) types. While such theories are interesting in their own right, one is more often interested in theories with type constructors: (dependent) sums and products, Id-types (most crucially for the present work), and beyond.

With this in mind, we look at what structure on the categorical side corresponds to the usual rules for such constructors. For better comparision, we will again give two equivalent presentations: one very literal to the syntax of the theory ("elim-structure"), one more abstract

Definition 2.1. An *elim-structure* on a map $f\colon \Xi \longrightarrow \Theta$ is a function E, assigning to each $C \in p(\Theta)$ and each map $d\colon \Xi \longrightarrow \Theta.C$ over Θ a section $E(C,d)\colon \Theta \longrightarrow \Theta.C$ of the dependent projection π_C satisfying $E(C,d) \cdot f = d$.

Syntactically, this corresponds to the usual style elimination rule

$$\frac{\vec{y} : \Theta \vdash C(\vec{y})\ \mathsf{type} \qquad \vec{x} : \Xi \vdash d(\vec{x}) : C(f(\vec{x}))}{\vec{y} : \Theta \vdash E(C,d;\vec{y}) : C(\vec{y})}$$

with computation rule concluding $E(C,d;f(\vec{x})) = d(\vec{x})$. (Compare Id-elim.)

Categorically, E gives fillers for certain triangles:

$$\begin{tikzcd} \Xi \arrow[r, "d"] \arrow[d, "f"'] & \Theta.C \arrow[dl, bend left] \\ \Theta \arrow[ur, dotted, "E(C,d)"] & \end{tikzcd}$$

This in turn implies a more familiar square-filling

$$\begin{tikzcd} \Xi \arrow[r] \arrow[d, "f"'] & \Delta \arrow[d, two heads] \\ \Theta \arrow[r] \arrow[ur, dotted] & \Gamma \end{tikzcd}$$

(exhibiting f as weakly orthogonal to all dependent projections; see Section **??** below, and cf. [**GG08**]), together with some stability conditions on the resulting fillers.

Definition 2.2. A *Frobenius elim-structure* on a map $f\colon \Xi \longrightarrow \Theta$ is a choice of elim-structure E_Δ on $(f.\Delta)\colon \Xi.(f^*\Delta) \longrightarrow \Theta.\Delta$, for each $\Delta \in p^{\mathsf{cxt}}(\Theta)$.

Syntactically this corresponds to an extra parameter in all the contexts of the rule:

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

Definition 2.3. An Id-*structure* on a (plain or stratified) comprehension category $(\mathcal{C}, p, \mathcal{P})$ consists of the following data for each context $\Gamma \in \mathcal{C}$ and type $A \in p(\Gamma)$:

(1) a type $\mathrm{Id}_A \in p(\Gamma.A.A)$;
(2) a map $r_A\colon \Gamma.A \longrightarrow \Gamma.A.A.\mathrm{Id}_A$ lifting the diagonal (contraction) map $\delta_A\colon \Gamma.A \longrightarrow \Gamma.A.A$ over Γ

$$
\begin{tikzcd}
 & \Gamma.A.A.\mathrm{Id}_A \arrow[d, two heads] \\
\Gamma.A \arrow[ur, "r_A"] \arrow[r, "\delta_A"] \arrow[dr] & \Gamma.A.A \arrow[d, two heads] \\
 & \Gamma
\end{tikzcd}
$$

(3) a Frobenius elim-structure J_A on r_A,

$$
\begin{tikzcd}
\Gamma.A.\Delta \arrow[rr, "d"] \arrow[dr, "r_A.\Delta"'] & & \Gamma.A.A.\mathrm{Id}_A.C \arrow[dl, two heads] \\
 & \Gamma.A.A.\mathrm{Id}_A \arrow[ur, dotted, "J_{A,\Delta}(C,d)"] &
\end{tikzcd}
$$

all stably in Γ, in that for $A \in p(\Gamma)$ and $f\colon \Theta \longrightarrow \Gamma$,

(1) $(f.A.A)^*\mathrm{Id}_A = \mathrm{Id}_{f^*A} \in p(\Theta.f^*A.f^*A)$:

$$
\begin{tikzcd}
\mathrm{Id}_{f^*A} & Id_A \arrow[l, mapsto] \\
\Theta.f^*A.f^*A \arrow[r, "f.A.A"] & \Gamma.A.A
\end{tikzcd}
$$

(2) $f^*(r_A) = r_{f^*A}$; equivalently, the following square commutes:

$$
\begin{tikzcd}
\Theta.f^*A \arrow[r] \arrow[d, "r_{f^*A}"] & \Gamma.A \arrow[d, "r_A"] \\
\Theta.f^*A.f^*A.\mathrm{Id}_{f^*A} \arrow[r] & \Gamma.A.A.\mathrm{Id}_A
\end{tikzcd}
$$

(3) and, for all suitable Δ, C, d, we have $f^*(J_{A,\Delta}(C,d)) = J_{f^*A,f^*\Delta}(f^*C, f^*d)$; in other words, the square

$$
\begin{tikzcd}
\{\text{triangles over } r_A.\Delta\} \arrow[r, "f^*"] \arrow[d, "J_{A,\Delta}"] & \{\text{triangles over} r_{f^*A}.f^*\Delta\} \arrow[d, "J_{f^*A,f^*\Delta}"] \\
\{\text{filled triangles}\} \arrow[r, "f^*"] & \{\text{filled triangles}\}
\end{tikzcd}
$$

commutes.

(TO DO: sleep on this for a while, try to find a nice way of wrapping this up, eg fibrationally or similar!)

TO DO: define the various categories $\mathbf{CwA}^{\mathrm{Id}}$, etc.

Proposition 2.4. *If* T *is any DTT with* Id*-types, then* $\mathbf{cl}(\mathsf{T})$ *admits a canonical* Id*-structure. Conversely, if* T *is any (plain or stratified) category with attributes, then* $\mathbf{th}(\mathcal{C})$ *admits an interpretation of the* Id*-rules; and the maps* $\epsilon_{\mathcal{C}}\colon \mathbf{cl}(\mathbf{th}(\mathcal{C})) \longrightarrow \mathcal{C}$ *and* $\eta\colon \mathsf{T} \longrightarrow \mathsf{T}'$ *preserve the resulting* Id*-structure.*

In particular, the equivalence $\mathbf{Th} \simeq \mathbf{CwA}_{strat}$ *lifts to an equivalence* $\mathbf{Th}^{\mathrm{Id}} \simeq \mathbf{CwA}^{\mathrm{Id}}$.

Proof. Straighforward verification. □

Proposition 2.5 (Identity contexts). *An* Id*-structure on* $(\mathcal{C}, p, \mathcal{P})$ *lifts to one on* $(\mathcal{C}, p^{\mathsf{cxt}}, \mathcal{P}^{\mathsf{cxt}})$.

Note the interesting type-theoretic content: this shows that from identity *types* for dependent types, we can build identity *contexts* for dependent contexts, satisfying all the same rules.

Proof. We just sketch the proof here; see [**Gar09a**, 2.3.1] for details. By Proposition 2.4, we may work type-theoretically.

(TODO: is this worth the notation that it requires developing?) □

Thus the $(-)^{\mathsf{cxt}} \colon \mathbf{CwA} \longrightarrow \mathbf{CwA}$ lifts to a functor $\mathbf{CwA}^{\mathrm{Id}} \longrightarrow CwAId$. Note, however, that the monad structure does not lift on the nose: its multiplication fails to preserve Id-structure.

The coslice construction.

Proposition 2.6 (Co-slice categories with attributes). *If* $\mathcal{C}$ *is a cwa, and* Θ *any context of* $\mathcal{C}$*, then there is a natural cwa structure on the co-slice category* $\Theta/\mathcal{C}$.

If $\mathcal{C}$ *was stratified, then so is* $\Theta/\mathcal{C}$.

If $\mathcal{C}$ *had unit types, sum types, or* Id*-types, then so does* $\Theta/\mathcal{C}$.

In all of these cases, the forgetful functor $\mathrm{cod} \colon \Theta/\mathcal{C} \longrightarrow \mathcal{C}$ *preserves all the given structure.*

Proof. Recall that an object of $\Theta/\mathcal{C}$ is a pair (Γ, k), where $\Gamma \in \mathcal{C}$ and $\vec{g} \colon \Theta \longrightarrow \Gamma$ ("a Θ-pointed object of $\mathcal{C}$"); a map $f \colon (\Gamma', \vec{g}') \longrightarrow (\Gamma, \vec{g})$ is a map in $\mathcal{C}$ preserving the pointings:

$$\begin{tikzcd} & \Gamma' \arrow[r, "f"] & \Gamma \\ \Theta \arrow[ur, "\vec{g}'"] \arrow[urr, "\vec{g}"'] & & \end{tikzcd}$$

We define the attributes structure on $\Theta/\mathcal{C}$ by taking an object of $p(\Gamma, \vec{g})$ to be a type $A \in p(\Gamma)$ together with a map $\vec{g}.a : \Theta \longrightarrow \Gamma.A$ over $\vec{g}$

$$\begin{tikzcd} & \Gamma.A \arrow[d] \\ \Theta \arrow[ur, "\vec{g}.a"] \arrow[r, "\vec{g}"'] & \Gamma \end{tikzcd}$$

and defining $(\Gamma, \vec{g}).(A, \vec{g}.a) = (\Gamma.A, \vec{g}.a)$.

All the remaining structure is determined straightforwardly by the requirement that is preserved by $\mathrm{cod} \colon \Theta/\mathcal{C} \longrightarrow \mathcal{C}$. We give here the details just for Id-structure, which is the case we will need later.

Given a type $(A, \vec{g}.a) \in p(\Gamma, \vec{g})$, the underlying object of $\mathrm{Id}_{(A, \vec{g}.a)}$ has to be Id_A; and in order to make the intro map r_A commute with pointings, we must give Id_A

the pointing $r_A \cdot (\vec{g}.a)$:

$$
\begin{tikzcd}
 & \Gamma.A \arrow[r, "r_A"] \arrow[dr] & \Gamma.A.A.\mathrm{Id}_A \arrow[d] \\
\Theta \arrow[ur, "\vec{g}.a"] \arrow[urr, dotted] \arrow[rr, "\vec{g}"] & & \Gamma
\end{tikzcd}
$$

This gives us $\mathrm{Id}_{(A,\vec{g}.a)}$ and $r_{(A,\vec{g}.a)}$. To lift the elim structure, we just need to check that the map $J_{A,\Delta}(C,d)$ will commute with all pointings involved whenever d does, but this follows straightforwardly from the equation $J_{A,\Delta}(C,d) \cdot r_A.\Delta = d$.

$$
\begin{tikzcd}
 & \Gamma.A.\Delta \arrow[r, "d"] \arrow[d, "r_A.\Delta"'] & \Gamma.A.A.\mathrm{Id}_A.\Delta.C \arrow[dl, bend left] \\
\Theta \arrow[ur] \arrow[r] & \Gamma.A.A.\mathrm{Id}_A.\Delta \arrow[ur, dotted, "J_{A,\Delta}(C,d)"] &
\end{tikzcd}
$$

$\square$

CHAPTER 3

Globular structures from type theory

1. The fundamental weak omega-groupoid of a type

TODO: either here or in introduction, give historical background: the move from seeing it as a transitive relation, to a groupoid-up-to-prop-eq, to ("the parallels with the fundamental group of a space are inescapable") a weak ω-groupoid.

Hacking. To understand how the ω-groupoid structure arises, it is instructive to make a start on its construction by hand.

(A few notes on derivations: read bottom-up; and obvious premises are omitted. TODO: maybe put this earlier?)

The first term often derived for identity types is a witness for the transitivity of equality on X—which for us, of course, will give binary composition of 1-cells in $\Pi_\omega(X)$:

$$x, y, z : X, u : \mathrm{Id}_X(x, y), v : \mathrm{Id}(y, z) \vdash c_X(v, u) : \mathrm{Id}(x, z)$$

How do we derive such a term? We can just use Id-elim:

TODO: give the derivation.

Of course, we could also have eliminated on v first instead of u. Alternatively, we didn't need to stop after one use of Id-elim: we could have applied it to the second variable as well:

TODO: give this version too!

Exercise 1.1. Derive terms witnessing one or more of the following:

(1) "symmetry of equality", i.e. (weak) inverses of 1-cells:

$$x, y : X, u : \mathrm{Id}(x, y) \vdash ? : \mathrm{Id}(y, x)$$

(2) associativity, for any one of the binary compositions terms above (or for a mixture of them):

$$\begin{aligned} &x, y, z, w : X, t : \mathrm{Id}(x, y), u : \mathrm{Id}(y, z), v : \mathrm{Id}(z, w) \\ &\qquad\qquad\qquad\qquad \vdash \ ? : \mathrm{Id}_{\mathrm{Id}_X}(\ c(c(v, u), t)\ ,\ c(v, c(u, t))\ ) \end{aligned}$$

(For bonus points, find a one-size-fits-all approach which works with any of the binary compositions operations.)

(3) horizontal composition of 2-cells:

$$x, y, z : X, u, u' : \mathrm{Id}(x, y), p : \mathrm{Id}(u, u'), v, v' : \mathrm{Id}(y, z), q : \mathrm{Id}(v, v') \ \vdash \ ? :?$$

Outline of the proof. Having tackled a few examples, we can now generalise the tactics involved, giving the main points of the full proof:

(1) We constructed our composition operations as maps in the syntactic category, from contexts built out of the identity types of X back into those

identity types themselves. This suggests that these are operations in some kind of *endomorphism operad in the syntactic category.*

(2) We used no special information about the type X; in other words, we effectively were deriving composition operations just over a *generic type*, which can then be implemented on any other type. (On this point, [**GvdB08**] take a different approach; see ¶**??** below.)

(3) We now want to see that this composition operad is *contractible*! What does this mean? The exercises above, particularly the last two, are typical instances of the liftings that contractibility requires: given two parallel composition operations, we have to find another connecting them one dimension up; that is, given two terms σ, τ we have to derive a term of idenity type $\mathrm{Id}(\sigma, \tau)$ between them.

(4) How do we do that? Our context is built out of identity types, so we *repeatedly apply* Id-*elim* and hope that our target type simplifies until we can inhabit it. As in the first derivation above, one might sometimes stop early; but as Ex.2 shows, the most generally powerful approach is to eliminate away *all* higher variables in the context, until only a single variable $x : X$ remains; so it is enough to show that σ and τ are propositionally equal once x and its (higher) reflexivity terms $r(x), r(r(x)), \ldots$ have been plugged in for all the variables of σ and τ.

(5) In all the examples above, σ and τ then computed down to higher reflexivity terms themselves, and so were definitionally equal. In fact this turns out to be typical: *the context $x : X$ is an initial object*, so σ and τ must always simplify in this way, and we are done.

Globular contexts, and their endomorphism operads. For any cwa $\mathcal{C}$, the class of dependent projections is closed under pullbacks, so by ¶**??**, we can form the monoidal globular category $\mathbf{FibSpans}(\mathcal{C})$; and maps of cwa's preserve dependent projections and pullbacks of them, so we have a functor $\mathbf{FibSpans} : \mathbf{CwA} \longrightarrow \mathbf{MonGlobCat}$.

A globular object of $\mathbf{FibSpans}(\mathcal{C})$ is then just a globular object $\Gamma_\bullet : \mathbb{G} \longrightarrow \mathcal{C}$ in $\mathcal{C}$, in which all the maps s, t are dependent projections:

$$\Gamma_0 \overset{t}{\underset{s}{\twoheadleftarrow\!\!\twoheadleftarrow}} \Gamma_1 \overset{t}{\underset{s}{\twoheadleftarrow\!\!\twoheadleftarrow}} \Gamma_2 \overset{t}{\underset{s}{\twoheadleftarrow\!\!\twoheadleftarrow}} \cdots$$

Mention abuses of notation: $\mathrm{End}_{\mathcal{C}}(A_\bullet)$, $\mathrm{End}_{\mathsf{T}}(A_\bullet)$.

The theory of a generic type. TODO: Construct $\mathsf{ML}^{\mathrm{Id}}[X]$; state universal property.

Definition 1.2. Let $\mathsf{ML}^{\mathrm{Id}}[X] \in \mathbf{Th}^{\mathrm{Id}}$ be the extension of $\mathsf{ML}^{\mathrm{Id}}$ by a single new type

$$\frac{}{\diamond \vdash X \ \mathsf{type}} \ X\text{-FORM}$$

and no further terms or axioms.

$\mathsf{ML}^{\mathrm{Id}}[X]$ is thus the free theory on a closed type (in $\mathbf{Th}^{\mathrm{Id}}$): any closed type A of a theory T uniquely determines a translation $F_A : \mathsf{ML}^{\mathrm{Id}}[X] \longrightarrow \mathsf{T}$ with $F_A(X) = A$.

By the adjunctions of **¶??**, $\mathbf{cl}_{\mathrm{strat}}(\mathsf{ML}^{\mathrm{Id}}[X])$ is thus the free object of $\mathbf{CwA}^{\mathrm{Id}}_{\mathrm{strat}}$ with a distinguished closed type, and $\mathbf{cl}(\mathsf{ML}^{\mathrm{Id}}[X])$ is free among CwA's $\mathcal{C} \in \mathbf{CwA}^{\mathrm{Id}}$ with a distinguished context Γ and type $A \in p(\Gamma)$.

In particular, for any such $\mathcal{C}$, $\diamond$, A, this gives a monoidal globular functor $\mathbf{FibSpans}(\mathbf{cl}(\mathsf{ML}^{\mathrm{Id}}[X])) \longrightarrow \mathbf{FibSpans}(\mathcal{C})$, sending $X_\bullet$ to $A_\bullet$, and hence an operad map $\mathrm{End}_{\mathsf{ML}^{\mathrm{Id}}[X]}(X_\bullet) \longrightarrow \mathrm{End}_{\mathcal{C}}(A_\bullet)$.

So $\mathrm{End}_{\mathsf{ML}^{\mathrm{Id}}[X]}(X_\bullet)$ acts on all other types $A \in p(\Gamma)$, naturally in $(\mathcal{C}, p, \mathcal{P})$ and Γ. Syntactically, this is the fact that a compositin law on X derived in $\mathsf{ML}^{\mathrm{Id}}[X]$ could just as well have been derived for any ofther type A in any theory with Id-types. In view of this, we will give the operad a less unwieldy name:

Definition 1.3. We will write $P_{\mathsf{ML}^{\mathrm{Id}}}$ for the globular operad $\mathrm{End}_{\mathsf{ML}^{\mathrm{Id}}[X]}(X_\bullet)$ of *generically definable composition laws.*

1.4. According to **¶??**, a composition law in $P_{\mathsf{ML}^{\mathrm{Id}}}(\pi)$ consists of a natural transformation $(S_0, T_0, \ldots T_{n-1}, R)$ between pylon diagrams as in Figure 1.4, where Γ_n denotes the context

$$\Gamma_n \;:=\; x_0, y_0 : X, x_1, y_1 : \mathrm{Id}(x_0, y_0), \ldots \; z : \mathrm{Id}(x_{n-1}, y_{n-1})$$

and $\Gamma_{\widehat{\pi}}$ has... [TODO: complete description!]

$$\begin{tikzcd}
\Gamma_{\widehat{\pi}} \arrow[d] \arrow[ddr] \arrow[rr, dotted, "R"] & & \Gamma_n \arrow[d] \arrow[ddr] & \\
\Gamma_{\widehat{s\pi}} \arrow[rr, "S_{n-1}"] \arrow[dd] \arrow[dddr] & & \Gamma_{n-1} \arrow[dd] \arrow[dddr] & \\
& \Gamma_{\widehat{t\pi}} \arrow[rr, "T_{n-1}"] \arrow[dl] \arrow[dd] & & \Gamma_{n-1} \arrow[dl] \arrow[dd] \\
\Gamma_{\widehat{s^2\pi}} \arrow[rr, "S_{n-2}"] & & \Gamma_{n-2} & \\
& \Gamma_{\widehat{t^2\pi}} \arrow[rr, "T_{n-2}"] & & \Gamma_{n-2} \\
\vdots & & \vdots & \\
& \vdots & & \vdots \\
\Gamma_{\widehat{s_1\pi}} \arrow[rr, "S_1"] \arrow[dd] \arrow[dddr] & & \Gamma_1 \arrow[dd] \arrow[dddr] & \\
& \Gamma_{\widehat{t_1\pi}} \arrow[rr, "T_1"] \arrow[dl] \arrow[dd] & & \Gamma_1 \arrow[dl] \arrow[dd] \\
\Gamma_{\widehat{s_0\pi}} \arrow[rr, "S_0"] & & \Gamma_0 & \\
& \Gamma_{\widehat{t_0\pi}} \arrow[rr, "T_0"] & & \Gamma_0
\end{tikzcd}$$

Proposition 1.5. *X is an initial object in $\mathfrak{g}_0$.*

Proof. By the universal property of $\mathfrak{g}_0$, there is a unique translation F_{1_X} from $\mathfrak{g}_0$ to the coslice $X/\mathfrak{g}_0$ sending X to $1_X : X \longrightarrow X$. Again by the universal property, the composite $\mathfrak{g}_0 \longrightarrow X/\mathfrak{g}_0 \longrightarrow \mathfrak{g}_0$ is equal to $1_{\mathfrak{g}_0}$, since it fixes X.

Thus the values of F_{1_X} on objects give for each $\Gamma \in \mathfrak{g}_0$ a natural map $\vec{r}^{\Gamma} := F_{1_X}(\Gamma) \colon X \longrightarrow \Gamma$, with $\vec{r}^{X} = 1_X$. From this, we see that X is an initial object by considering the naturality square

$$\begin{tikzcd}
X \arrow[r, "\vec{r}^X = 1_X"] \arrow[d, "1_X"'] & X \arrow[d, "f"] \\
X \arrow[r, "\vec{r}^\Gamma"] & \Gamma
\end{tikzcd}$$

for any map $f : X \longrightarrow \Gamma$. □

Syntactically, deducing F_{1_X} from the universal property of $\mathfrak{g}_o$ corresponds to inductively constructing, for each context Γ, a context morphism

$$x : X \vdash \vec{r}^\Gamma(x) : \Gamma$$

such that every other context morphism preserves the point:

$$\frac{\vec{y} : \Gamma \ \vdash \ \vec{f}(\vec{y}) : \Delta}{x : X \ \vdash \ \vec{f}(\vec{r}^\Gamma(x)) = \vec{r}^\Delta(x) : \Delta}$$

(See [**Lum09**] for a full presentation of this syntactic approach.)

Concretely, since all contexts of $\mathfrak{g}_0$ are constructed from X and identity types over it, $\vec{r}^\Gamma(x)$ is built by plugging x in for each variable of type X in Γ, and reflexivity terms for all variables of Id-types.

Corollary 1.6. *The operad $P_{\mathsf{ML}^{\mathrm{Id}}}$ is contractible.*

Proof. By **¶??**, it suffices to show that for any pasting diagram π and map $\Gamma_{\widehat{\pi}} \longrightarrow \Gamma_{\delta(n)}$, we can complete the diagram to a triangle

$$\begin{tikzcd}
 & \Gamma_{\mathbf{y}(n)} \arrow[d, two heads] \\
\Gamma_{\widehat{\pi}} \arrow[r] \arrow[ur, dotted] & \Gamma_{\delta(n)}
\end{tikzcd}$$

But by the initiality of X, the diagram can always be completed to a square

$$\begin{tikzcd}
X \arrow[r] \arrow[d] & \Gamma_{\mathbf{y}(n)} \arrow[d, two heads] \\
\Gamma_{\widehat{\pi}} \arrow[r] \arrow[ur, dotted] & \Gamma_{\delta(n)}
\end{tikzcd}$$

and since $\vec{r}^{\Gamma_{\widehat{\pi}}}$ is an I-map, we can find a diagonal filler for this square, completing the original triangle as desired. □

Interpreted syntactically, this is precisely the process described in **¶??**. The "problem" we are given, lifting a map into $\Gamma_{\delta(n)}$ to one into $\Gamma_{\mathbf{y}(n)}$, is just the problem of deriving a term $\vec{y} : \Gamma_{\widehat{\pi}} \ \vdash \ ? : \mathrm{Id}(S_{n-1}(\vec{y}), T_{n-1}(\vec{y}))$. The fact that $\vec{r}^{\Gamma_{\widehat{\pi}}}$ is an I-map tells us that (by repeated application of Id-ELIM and structural isomorphisms) it suffices to derive the term in the case where $\vec{r}^{\Gamma_{\widehat{\pi}}}(x)$ has been substituted for $\vec{y}$. Finally, the initiality of X ensures that after this substitution, (the last component of) the context morphism $\vec{r}^{\Gamma_{\mathbf{y}(n)}}$ fits the bill—specifically, S_{n-1} and T_{n-1} must compute down to be definitionally equal, and so we can use a reflexivity term.

Compare: [**GvdB08**], with tailor-made operad from $A/\mathcal{C}$.

Add: groupoidality!

2. The classifying weak ω-category of a theory

Overview: what are we trying to build?

Representability: the type theoretic globes. More abstract view: set up Kan situation! Type-theoretic globes.

Make analogy with **Top**. How much detail to go into? Separate subsection*? Compare: [**Bat98**, §9],

Definition 2.1. Let $\mathfrak{g}_n$, the *type-theoretic n-globe*, be the theory defined by the following axioms:

$$n = 0: \qquad \frac{}{\diamond \vdash X \ \mathsf{type}}$$

$$n > 0: \qquad \frac{}{\diamond \vdash X^s \ \mathsf{type}} \qquad \frac{}{\diamond \vdash X^t \ \mathsf{type}}$$

$$n = 1: \qquad \frac{}{x : X_s \vdash p_1(x) : X_t}$$

$$n > 1: \qquad \frac{}{x : X_s \vdash p_1^s(x) : X_t} \qquad \frac{}{x : X_s \vdash p_1^t(x) : X_t}$$

$$(1 < i < n) \qquad \frac{}{x : X_s \vdash p_i^s(x) : \mathrm{Id}(p_{i-1}^s, p_{i-1}^t)} \qquad \frac{}{x : X_s \vdash p_i^t(x) : \mathrm{Id}(p_{i-1}^s, p_{i-1}^t)}$$

$$\frac{}{x : X_s \vdash p_n(x) : \mathrm{Id}(p_{n-1}^s, p_{n-1}^t)}$$

Then $\mathbf{cl}_\omega^-(\mathbb{T}) := \mathbf{Th}^{\mathrm{Id}}(\mathfrak{g}_\bullet, \mathbb{T})$ is the underlying globular set of our desired classifying weak ω-category of types of $\mathbb{T}$.

Indeed, since $\mathbf{Th}^{\mathrm{Id}}$ is co-complete, Kan extension of $\mathfrak{g}_\bullet$ along the Yoneda embedding gives a left adjoint $\mathbb{T}[-] \dashv \mathbf{cl}_\omega^-$. Analogously to the geometric realisation $|X|$ of a globular (or simplicial) set, we call $\mathbb{T}[X]$ the *type-theoretic realisation* of X, a theory with axioms implementing 0-cells of X as closed types, 1-cells as open terms mapping between these, and higher cells as terms of Id-type. (This follows from the usual colimit formula for Kan extensions[**?**, ??] and the description of colimits in **??** above.) In particular, $\mathbb{T}[y(n)] = \mathfrak{g}_n$.

$$\begin{tikzcd}
\widehat{\mathbb{G}} \arrow[r, bend right=10, "{\mathbb{T}[-] := \mathrm{Lan}_{\mathbf{y}} \mathfrak{g}_\bullet}"'] \arrow[r, phantom, "\top"] & \mathbf{Th}^{\mathrm{Id}} \arrow[l, bend right=10, "{\mathbf{cl}_\omega^- := \mathbf{Th}^{\mathrm{Id}}(\mathfrak{g}_\bullet, -)}"'] \\
\mathbb{G} \arrow[u, hook, "\mathbf{y}"] \arrow[ur, bend right, "\mathfrak{g}_\bullet"']
\end{tikzcd}$$

2.2. So, by 4.2, to put a natural P-algebra structure on $\mathbf{cl}_\omega^-$ is equivalent to putting a co-P-algebra structure on $\mathfrak{g}_\bullet$. In particular, its coendomorphism operad $\mathrm{End}_{(\mathbf{Th}^{\mathrm{Id}})^{\mathrm{op}}}(\mathfrak{g}_\bullet)$ acts naturally on $\mathbf{cl}^-\omega$; in other words, we can lift if to a functor into $\mathrm{End}_{(\mathbf{Th}^{\mathrm{Id}})^{\mathrm{op}}}(\mathfrak{g}_\bullet)$:

$$\begin{tikzcd} & \left[\mathrm{End}_{(\mathbf{Th}^{\mathrm{Id}})^{\mathrm{op}}}\mathfrak{g}_\bullet\right]\text{-}\mathbf{Alg} \arrow[d, "U"] \\ \mathbf{Th}^{\mathrm{Id}} \arrow[ur, "\mathbf{cl}^{-}\omega"] \arrow[r, "\mathbf{Th}^{\mathrm{Id}}(\mathfrak{g}_\bullet{,}-)"] & \widehat{\mathbb{G}} \end{tikzcd}$$

With this in mind, we investigate this operad. As usual, an operad element of dimension n consists of a natural transformation $(S_0, T_0, \ldots, S_{n-1}, T_{n-1}, R)$ between pylon diagrams, but this time going from the pylon representing the n-globe to one representing a pasting diagram as in 2.3 below.

Before proceeding to investigate contractibility in general, however, it is helpful to pause for a few observations on the low dimensions of this operad.

TODO: points to make: by arguments like before, or by normalisation, X_0 is again initial in the theory of any one-dimensional pasting diagram (i.e. path), and so the only elements of the operad in dimensions 0 and 1 are the ones you'd expect: the identity in dim 0, and a unique k-ary composition for each k in dim 1. But, more importantly, we also get that any map out of a globe (or any colimit of globes) into any $\mathbb{T}[\hat{\pi}]$ (for 1-dimensional π) is determined by what it does to in dimensions 0 and 1.

2.3 (Contractibility)**.** Figure 2.3 depicts the lifting condition required for contractibility of $\mathrm{End}_{(\mathbf{Th}^{\mathrm{Id}})^{\mathrm{op}}}(\mathfrak{g}_\bullet)$: this amounts to showing that whenever $(S_0, T_0, \ldots, S_{n-1}, T_{n-1}$ are given making the lower dimensions commute, some R can be found to complete the diagram.

$$\begin{tikzcd}
& \mathfrak{g}_n \arrow[rr, dotted, "R"] & & \mathbb{T}[\widehat{\pi}] \\
\mathfrak{g}_{n-1} \arrow[ur] \arrow[rr, "S_{n-1}"] & & \mathbb{T}[\widehat{s\pi}] \arrow[ur] & \\
& \mathfrak{g}_{n-1} \arrow[uu] \arrow[rr, "T_{n-1}"] & & \mathbb{T}[\widehat{t\pi}] \arrow[uu] \\
\mathfrak{g}_{n-2} \arrow[uu] \arrow[ur] \arrow[rr, "S_{n-2}"] & & \mathbb{T}[\widehat{s^2\pi}] \arrow[uu] \arrow[ur] & \\
& \mathfrak{g}_{n-2} \arrow[uu] \arrow[uuul] \arrow[rr, "T_{n-2}"] & & \mathbb{T}[\widehat{t^2\pi}] \arrow[uu] \arrow[uuul] \\
\vdots & \vdots & \vdots & \vdots \\
\mathfrak{g}_1 \arrow[rr, "S_1"] & & \mathbb{T}[\widehat{s_1\pi}] & \\
& \mathfrak{g}_1 \arrow[rr, "T_1"] & & \mathbb{T}[\widehat{t_1\pi}] \\
\mathfrak{g}_0 \arrow[uu] \arrow[ur] \arrow[rr, "S_0"] & & \mathbb{T}[\widehat{s_0\pi}] \arrow[uu] \arrow[ur] & \\
& \mathfrak{g}_0 \arrow[uu] \arrow[uuul] \arrow[rr, "T_0"] & & \mathbb{T}[\widehat{t_0\pi}] \arrow[uu] \arrow[uuul]
\end{tikzcd}$$

We can restate this as an extension problem:

$$\begin{tikzcd} \mathbb{T}[\delta(n)] \arrow[r, "(\vec{S}{,}\vec{T})"] \arrow[d, hook] & \mathbb{T}[\delta\widehat{\pi}] \arrow[r, hook] & \mathbb{T}[\widehat{\pi}] \\ \mathfrak{g}_n = \mathbb{T}[\mathbf{y}(n)] \arrow[urr, dotted, "R"] & & \end{tikzcd}$$

By the observations above (TODO: work out where to put them!) this "corner" completes to a commutative square, so we in fact have a square-filling problem:

$$\begin{tikzcd}
\mathbb{T}[\delta(n)] \arrow[r, "(\vec{S},\vec{T})"] \arrow[d, hook] & \mathbb{T}[\delta\widehat{\pi}] \arrow[r, hook] & \mathbb{T}[\widehat{\pi}] \arrow[d] \\
\mathfrak{g}_n = \mathbb{T}[\mathbf{y}(n)] \arrow[r, two heads] \arrow[urr, dotted, "R"] & \mathfrak{g}_1 \arrow[r] & \mathbb{T}[\widehat{s_1\pi} = \widehat{t_1\pi}]
\end{tikzcd}$$

(TODO: rephrase this to avoid duplicating the diagram, while still keeping the train of thought clear!)

Rephrase *syntactically*: completing the translation is just showing that a certain identity type is inhabited. This motivates:

Conjecture 2.4. *In the theories $\mathbb{T}[\pi]/X_0$, any two closed terms of the same type are propositionally equal.*

If this holds, then the operad $\mathrm{End}_{(\mathbf{Th}^{\mathrm{Id}})^{\mathrm{op}}}(\mathfrak{g}_\bullet)$ is contractible, and hence $\mathbf{cl}_\omega$ carries a natural weak ω-category structure as desired. In the following sections, we discuss various approaches to proving this conjecture: by direct syntactic attack on the theories $\mathbb{T}[\pi]$, or by working more globally, using the structures on $(\mathbf{Th}^{\mathrm{Id}})^{\mathrm{op}}$ discussed in **??**. [TODO: put discussion of the structures on $\mathbf{Th}^{\mathrm{Id}^{\mathrm{op}}}$ in later section of this chapter, or in previous chapter? Big-picture-wise, belongs better there; pragmatically, more motivated here.] First, however, we pause to discuss a variant of $\mathbf{cl}_\omega$ using Π-types, which is more easibly seen to be contractible.

Approximation by Π-types.

2.5. Work in $\mathbf{Th}^{\mathrm{Id},\Pi}$. Define alternate globes using closed terms of Π-type and of their Id-types (so Id_Π not $\Pi(\mathrm{Id})$!) instead of open terms.

Show this is *is* contractble: since theories are now just extensions by Id-types, induction works, and (by normalisation) in the base case (i.e. theories of 1-dimensional

A NWFS on $\mathbf{Th}^{\mathrm{Id}}$. Introduce contractible maps; give nwfs by Garner small-object argument!

Conjecture various forms of $\bar{J}$

Theorem: if any of above conjectures holds, then have $\mathbf{cl}_\omega^-$.

2.6. Corollary: have $\mathbf{cl}_\omega$ by using $(-)^{\mathsf{cxt}}$!

CHAPTER 4

Homotopical constructions from globular higher categories

(NON)SECTION AIM: Motivate constructing the simplicial nerve, and hence the model structure. TODO: cut down, too discursive?? Maybe move this to end of previous section, or even to introduction, and here give an overview of simplicial structures / methods?

0.7. The great power of classifying categories in 1-categorical logic come from [‘depends on’?] an analysis of the logical constructors and rules in categorical terms: substitution as pullbacks, Π- and Σ-types as adjoints, and so on. So, a natural first impulse is to try to analyse the universal properties of the type constructors within $\mathbb{C}\lessdot_\omega(\mathsf{T})$, which we would expect to be weak-higher-categorical analogues of the usual logical structure.

Unfortunately, the theory of logical structure on globular higher-categories is not yet well-understood. Of course, we can hope that the developing dictionary with type theory will help understand how such structure should behave! However, there is an alternative model of higher categories for which the relevant theory is already much further advanced: Joyal’s quasi-categories. Quasi-categories are not a fully general theory of higher categories: they only model so-called $(\infty,1)$-categories, in which all cells above dimension 1 are (weakly) invertible. However, as we have seen, the classifying categories of type theories are of this form; so quasi-categories seem potentially excellently-suited for our desired analysis, if only we can give quasi-category models for $Clw(\mathsf{T})$!

0.8. In other words, we would like to construct a functor

$$\mathbb{C}\lessdot_\omega^{qcat} \colon \mathbf{Th} \longrightarrow \mathbf{QCat}.$$

TODO: Hmm, this doesn’t work if the reader doesn’t know yet that quascateogries are simplicial things! Work out how to re-organise to fit that in nicely.

There are two obvious options. Firstly, we could construct $\mathbb{C}\lessdot_\omega^{qcat}(\mathsf{T})$ directly from the theory T. [TODO: Ask Michael whether/how much to mention simplicial type theory.] However, Id-types as they stand are inescapably globular; there seems no obvious way to extract simplicial sets from the theory as cleanly and directly as one can extract globular sets. (Alternatively, the intriguing approach of re-axiomatising Id-types to be “naturally simplicial in shape” has been considered by Warren and Gambino [?].)

It thus seems natural to take a different approach: to construct $\mathbb{C}\lessdot_\omega^{qcat}$ in two steps, composing $\mathbb{C}\lessdot_\omega$ from the previous section with a functor

$$\mathcal{N}^{qcat} \colon P\text{-}\mathbf{Alg} \longrightarrow \mathbf{QCat}$$

giving the "quasi-category nerve" of a globular weak n-category. This has the added payoff that such a functor would be of independent interest, since the comparison between globular and simplicial higher categories is as yet little-understood in the fully weak case.

0.9. In Section 3, we will thus construct several candidate nerve functors. Constructing simplicial objects is straightforward; the hard part is proving the requisite horn filling conditions to show that they are quasi-categories.

It is for this that we construct, in Sections 1 and 2, a Quillen model structure on categories of globular higher categories (under certain extra hypotheses). The computational tools provided by such a structure provide precisely what we need to show that the horns arising in our nerve constructions can be filled, and hence that the nerves are indeed quasi-categories.

In fact, we construct an *algebraic* model structure in the sense of [**?**]. This is a Quillen model structure in which both weak factorisation systems are NWS's and there is moreover a comparison map connecting the two. While we will not need any of the extra power of an algebraic model structure, the algebraicity comes almost for free given the form of our proof.

1. Cellular algebraic model structures

SECTION AIM: The general construction of an algebraic model structure from a collection of generating cells. (TODO: read/ask around in case I've missed where something closer to this has already been done; work out terminology for this general construction.)

1.1. Recall what an AWFSand AMSare. (Or put this in appendix?)

1.2. We start by recalling from [**Gar09b**], [**Gar08**] the construction of an AWFS on **str**-n-**Cat** whose right maps are precisely the contractible maps.

...do it by the Garner small-object argument! Algebraic freeness shows the right maps are what we think. And Garner shows that this is also the adjunction with computads, fwiw (maybe leave this out if not needed).

Point out how this comes from the map $D(\mathrm{ob}\,\mathbb{G}) \longrightarrow \mathbb{G} \longrightarrow \widehat{\mathbb{G}} \longrightarrow \mathbf{wk}\text{-}n\text{-}\mathbf{Cat}$ of cells/boundaries.

1.3. In fact, the remainder of the construction of the AMS(though not the proof that it really is one) can be given entirely in terms of this set of generating cofibrations; and, indeed, L' categories will give another example. Thus for the remainder of this section, we will fix a category $\mathcal{E}$ that admits the small object argument [TODO: define this here or elsewhere!], a set $\mathcal{I}$ (considered as a discrete category), and a functor $\mathcal{I} \longrightarrow \mathcal{E}^{\mathbf{2}}$. The functor will remain nameless, but we will write maps in its image as $d_i \hookrightarrow c_i$, for $i \in \mathcal{I}$. (They are to be thought of as inclusions of boundaries into cells.)

1.4. Construct $(\mathbb{C}, \mathbb{F}_t)$; construct "canonical squares". Describe how to see them as equivs.

1.5. Construct $\mathbb{W}$, $\mathcal{W}$, $(\mathbb{C}_t, \mathbb{F})$ from this. Infer ξ, by [**?**, Rmk 3.6] (and hence get $\mathcal{C}_t \Rightarrow \mathcal{C}$, $\mathcal{F}_t \Rightarrow \mathcal{F}$).

1.6. Give $TF \Leftrightarrow F \cap W$, i.e. $\mathbb{F}_t$-**Alg** $\Leftrightarrow$ $\mathbb{F}$-**Alg** $\times_{\mathcal{E}^{\mathbf{2}}}$ $\mathbb{W}$-**Alg**. Give: $\mathbb{W} \longrightarrow \mathcal{E}^{\mathbf{2}}$ "creates retracts", so $\mathcal{W}$ closed under retracts.

1.7. Now the hard stuff!

2. An algebraic model structure on globular higher categories?

2.1. Discuss instantiating the theorem of the previous section to (a) L'-categories, (b) P-algebras; prove as many of the lemmas as possible!

3. Simplicial nerves of globular higher categories

3.1. Give aim; different NWFS for different nerves; proof with model structure that these give nerves!

3.2. Prove from model strux that these really do give quasi-categories.
TODO: read up Dugger references properly.

APPENDIX A

Background: Martin-Löf Type Theory

Possibly just make this a presentation of the type theory?

1. Presentation overview

1.1. The type theories we consider are all essentially variants on the basic type theory originally presented in [**ML75**]. See also (TODO: pick out good references!)

The core of the type theory, its basic judgements and structural rules, is given in Section 1.4; this will remain constant in all the theories we consider.

The type-constructors and various other rules that will be used are given in Section 1.5; we will consider theories including various different subsets of these rules.

Section **??** contains some fundamental lemmas concerning the resulting type theories: basic proof-theoretic properties, admissible rules, and so on. TODO: this has changed! Memo to self: don't write overviews of chapters before starting the chapters themselves...

For further background, see [**Pit00**] (a notably thorough and precise presentation), [**Jac99**, §6] (for a thoughtful discussion of this system within the wider type-theoretic context), [**Hof97**], [**NPS90**, Ch.3] (discussion of the use of simply-typed calculus as the "raw language'), and [**ML75**] (the originial presentation of the theory). The presentation uses essentially the formal presentation of [**Pit00**], in the notation of [**Jac99**], slightly modified to include dependent contexts and their morphisms.

(TODO: perhaps fully expound the presentation rather than just recapping?)

Notes (re-organise and/or remove later):

— point of using λ-calculus as raw language: to separate the subtleties of binding and capture-avoiding substitution from those of dependency and well-formedness.

— point of having the raw language already simply-typed, not completely un(i)typed: to maintain decidability of $\simeq$.

— abuse of notation: making variables explicit

— dependent contexts and maps: to formulate Frobenius rule

— dependent maps: really only need $\Gamma \vdash \vec{f} : \Delta \Rightarrow \Delta'$ in case either $\Gamma = \diamond$ (for substitution rules) or $Delta = \diamond$ (for Frobenius rules). However, it's simpler to give this than those two separately, plus natural for syntactic cwa.

— syntactic sugar: $\vec{x} : \Gamma \vdash a(\vec{x}) : A(\vec{x})$ really means $\Gamma \vdash a : A$, where all of these are *closed* metaexpressions, but a, A have arities $\mathsf{term}^n \longrightarrow \mathsf{term}$, $\mathsf{term}^n \longrightarrow \mathsf{type}$, where n is the length of Γ; and so on.

1.2. Raw syntax

Define: "raw language". (NB: sometimes called *metalanguage*, but not meaning the language in which we reason about the type theory; rather, plays a similar rôle to that of strings (or trees) of symbols in simpler systems.)

Define: signature, pre-terms, pre-types, pre-terms, judgements.

1.3. Judgement forms

Basic judgement forms	
$\Gamma \vdash A \ \mathsf{type}$	$\Gamma \vdash a : A$
$\Gamma \vdash A = A' \ \mathsf{type}$	$\Gamma \vdash a = a' : A$

Derived judgement forms	
$\Gamma \vdash \Delta \ \mathsf{cxt}$	$\Gamma \vdash \vec{f} : \Delta \Rightarrow \Delta'$
$\Gamma \vdash \Delta = \Delta' \ \mathsf{cxt}$	$\Gamma \vdash \vec{f} = \vec{f}' : \Delta \Rightarrow \Delta'$

Explain in what sense these are derived judgements!
(Discuss in parens why we use dependent cxts and morphisms.)
Perhaps (if feeling pedantic, or can find way to say it non-pedantically) discuss arities of things in judgements, etc.?

1.4. Rules: Structural core

Rules for contexts:

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

$$\frac{\Gamma \vdash \Delta \ \mathsf{cxt} \qquad \Gamma, \Delta \vdash A \ \mathsf{type}}{\Gamma \vdash \Delta, A \ \mathsf{cxt}}\ \mathsf{cxt\text{-}cons} \qquad \frac{\Gamma \vdash \Delta = \Delta' \ \mathsf{cxt} \qquad \Gamma, \Delta \vdash A = A' \ \mathsf{type}}{\Gamma \vdash \Delta, A = \Delta', A' \ \mathsf{cxt}}\ \mathsf{cxt{=}\text{-}cons}$$

Rules for types:

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

$$\frac{\Gamma \vdash A = B \ \mathsf{type} \qquad \Gamma \vdash B = C \ \mathsf{type}}{\Gamma \vdash A = C \ \mathsf{type}}\ \mathsf{type{=}\text{-}trans} \qquad \frac{\vec{x} : \Gamma \vdash A(\vec{x}) \ \mathsf{type} \qquad \diamond \vdash \vec{f} : \Gamma' \Rightarrow \Gamma}{\vec{y} : \Gamma' \vdash A(\vec{f}(\vec{y})) \ \mathsf{type}}\ \mathsf{type\text{-}subst}$$

$$\frac{\vec{x} : \Gamma \vdash A = A' \ \mathsf{type} \qquad \diamond \vdash \vec{f} = \vec{f}' : \Gamma' \Rightarrow \Gamma}{\vec{y} : \Gamma' \vdash A(\vec{f}(\vec{y})) = A'(\vec{f}'(\vec{y})) \ \mathsf{type}}\ \mathsf{type{=}\text{-}subst}$$

Rules for terms:

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

$$\frac{\begin{array}{c}\Gamma \vdash a : A \\ \Gamma \vdash A = A' \ \mathsf{type}\end{array}}{\Gamma \vdash a : A'} \ \mathsf{term\text{-}coerce} \qquad \frac{\begin{array}{c}\Gamma \vdash a = a' : A \\ \Gamma \vdash A = A' \ \mathsf{type}\end{array}}{\Gamma \vdash a = a' : A'} \ \mathsf{term{=}\text{-}coerce}$$

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

$$\frac{\begin{array}{c}\Gamma \vdash a = b : A \\ \Gamma \vdash b = c : A\end{array}}{\Gamma \vdash a = c : A} \ \mathsf{term{=}\text{-}trans} \qquad \frac{\begin{array}{c}\vec{x} : \Gamma \vdash a(\vec{x}) : A(\vec{x}) \\ \diamond \vdash \vec{f} : \Gamma' \Rightarrow \Gamma\end{array}}{\vec{y} : \Gamma' \vdash a(\vec{f}(\vec{y})) : A(\vec{f}(\vec{y}))} \ \mathsf{term\text{-}subst}$$

$$\frac{\begin{array}{c}\vec{x} : \Gamma \vdash a(\vec{x}) = a'(\vec{x}) : A(\vec{x}) \\ \diamond \vdash \vec{f} = \vec{f}' : \Gamma' \Rightarrow \Gamma\end{array}}{\vec{y} : \Gamma' \vdash a(\vec{f}(\vec{y})) = a'(\vec{f}'(\vec{y})) : A(\vec{f}(\vec{y}))} \ \mathsf{term{=}\text{-}subst}$$

Rules for context maps:

$$\frac{\diamond \vdash \Gamma \ \mathsf{cxt}}{\diamond \vdash \diamond : \Gamma \Rightarrow \diamond} \ \mathsf{cxt\text{-}}\diamond \qquad \frac{\diamond \vdash \Gamma \ \mathsf{cxt}}{\diamond \vdash \diamond = \diamond : \Gamma \Rightarrow \diamond} \ \mathsf{cxt{=}\text{-}}\diamond$$

$$\frac{\begin{array}{c}\Gamma \vdash \vec{f} : \Delta' \Rightarrow \Delta \\ \vec{x} : \Gamma, \vec{y} : \Delta(\vec{x}) \vdash A(\vec{x}, \vec{y}) \ \mathsf{type} \\ \vec{x} : \Gamma, \vec{z} : \Delta'(\vec{x}) \vdash a(\vec{x}, \vec{z}) : A(\vec{x}, \vec{f}(\vec{x}, \vec{z}))\end{array}}{\Gamma \vdash (\vec{f}, a) : \Delta' \Rightarrow (\Delta, A)} \ \mathsf{cxt\text{-}cons} \qquad \frac{\begin{array}{c}\Gamma \vdash \vec{f} : \Delta' \Rightarrow \Delta \\ \vec{x} : \Gamma, \vec{y} : \Delta(\vec{x}) \vdash A(\vec{x}, \vec{y}) \ \mathsf{type} \\ \vec{x} : \Gamma, \vec{z} : \Delta'(\vec{x}) \vdash a(\vec{x}, \vec{z}) : A(\vec{x}, \vec{f}(\vec{x}, \vec{z}))\end{array}}{\Gamma \vdash (\vec{f}, a) : \Delta' \Rightarrow (\Delta, A)} \ \mathsf{cxt\text{-}cons}$$

1.5. Type constructors and miscellaneous rules

Rules for type-constructors go here: Id-, Π-, Σ-, with just a tip of the hat to ℕ-, Bool, . . .

Rules for Id-types:

$$\frac{\Gamma \vdash A\ \mathsf{type}}{\Gamma, x, y : A \vdash \mathrm{Id}_A(x,y)\ \mathsf{type}}\ \text{Id-FORM} \qquad \frac{\Gamma \vdash A\ \mathsf{type}}{\Gamma, x : A \vdash r(x) : \mathrm{Id}_A(x,x)}\ \text{Id-INTRO}$$

$$\frac{\begin{array}{c}\Gamma, x, y : A, p : \mathrm{Id}_A(x,y), \vec{w} : \Delta(x,y,p) \vdash C(x,y,p,\vec{w})\ \mathsf{type} \\ \Gamma, z : A, \vec{v} : \Delta(z,z,r(z)) \vdash d(z,\vec{v}) : C(z,z,r(z),\vec{v})\end{array}}{\Gamma, x, y : A, p : \mathrm{Id}_A(x,y), \vec{w} : \Delta(x,y,p) \vdash J_{A;\ x,y,p.\Delta;\ x,y,p,\vec{w}.C}(z,\vec{v}.\ d(z,\vec{v});\ x,y,p,\vec{w}) : C(x,y,p,\vec{w})}\ \text{Id-ELIM}$$

$$\frac{\begin{array}{c}\Gamma, x, y : A, p : \mathrm{Id}_A(x,y), \vec{w} : \Delta(x,y,p) \vdash C(x,y,p,\vec{w})\ \mathsf{type} \\ \Gamma, z : A, \vec{v} : \Delta(z,z,r(z)) \vdash d(z,\vec{v}) : C(z,z,r(z),\vec{v})\end{array}}{\Gamma, x : A, \vec{w} : \Delta(x,y,p) \vdash J_{A;\ x,y,p.\Delta}(z,\vec{v}.\ d(z,\vec{v});\ x,x,r(x),\vec{w}) = d(x,\vec{w}) : C(x,y,p,\vec{w})}\ \text{Id-COMP}$$

$$\frac{\Gamma \vdash e : \mathrm{Id}_A(a,b)}{\Gamma \vdash a = b : A}\ \text{REFLECTION}$$

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

Include optional things: η- and extensionality rules for function-types; K and the truncation rules for Id-types.

Discuss implications between these?

1.6. Translations

Define translation, category of type theories.

2. Categories with attributes

Define: full split comprehension categories [**Jac93**, 4.10] / categories with attributes. [**Pit00**, §6.4]

TODO: ask around about reachable vs. "stratified" cwa's.

Define: Id-structure, Π-structure, etc. on cwa's. (Reference: [**AW09**].)

2.1. Equivalence with type theories

Recall (from Pitts): *equivalence* (honest 1-equivalence!) between type theories with no constructors and stratified fscc's. TO DO: read Jacobs in full (& chase further. . .) re theories with constructors!

Note: $\mathbf{FSCC}_{\mathrm{strat}}$ is a *full* subcat of **FSCC**, and coreflective (with comonad induced by adjunction from **Th**, so closed under colims!

Extend equivalence above to constructors: $\mathbf{Th}^{\mathrm{Id}}$, $\mathbf{Th}^{\mathrm{Id},\Pi}$, etc.

Fact: since all essentially algebraic (or otherwise), these categories are all cocomplete!

APPENDIX B

Background: globular higher category theory

1. Strict higher categories

1.1. Define **str-n-Cat** and **str-ω-Cat** by enrichment.

1.2. Analyse T: pasting diagrams, Batanin trees, familial representability.
Fact: Cartesian, as can see from familial rep'bility[**Str00**],[**?**].

2. Globular operads: Leinster presentation

Definition 2.1. Contractible map of glob sets.

Definition 2.2. Leinster operad, as cartesian map of cartesian monads.

2.3. Equivalent: local presentation of Leinster operad: object over $T1$, with appropriate structure.

Definition 2.4.

2.5. Contractibility.

3. Globular operads: Batanin/Weber presentation

4. Internal algebras for operads

(This should probably be split up and moved: the globular part moved into the "Globular background" appendix, and then the construction of a monoidal globular category from a comprehension category folded into the construction of the fundamental things.)

Monoidal globular categories. Recall definition from Batanin.
Examples: **Spans**; **Spans**$(\mathcal{C})$, for $\mathcal{C}$ with pullbacks; **FibSpans**$(\mathcal{C})$, for $\mathcal{C}$ with a distinguished class of "fibrations", closed under composition and pullbacks.

4.1 (Pasting diagrams)**.** Example of a mon glob cat: $T1$. Give representation as Batanin trees; also as free monoids.

Notation note: I use $\delta(n)$ for the boundary of $\mathbf{y}(n)$, which is standard, but also $\delta\widehat{\pi}$ for the boundary of the $\widehat{\pi}$ which is less standard: Batanin, Street (TODO: refs?) and others use $\delta\pi := s\pi = t\pi$ (since these are always equal), whereas I always distinguish these (yes, they're equal as pasting diagrams, but in use they're typically 'intensionally' distinct) and my $\delta\widehat{\pi}$ is the pushout of $\widehat{s\pi}$ and $\widehat{t\pi}$ along their common boundary.

Give realisation as globular sets, by (a) iterated pushout, (b) colimit.

Define "one-leaf prunings".

In free monoid terms, defined inductively: removing an (extremal?) () at some depth. In tree terms: removing a leaf (an endpoint leaf?) at some height!

("Any leaf" is more natural—then these really do correspond to places one can apply Id-elim. On the other hand, "extremal leaf" makes pushout/pullback decompositions/lemmas simpler. Ah! Idea: maybe just define π^-, and then mention that could use more general one-leaf prunings.)

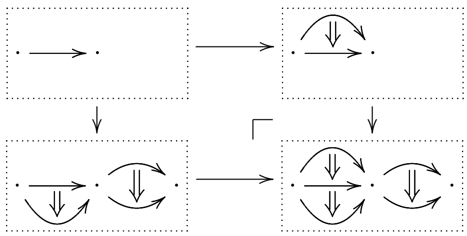
Note: every non-point pasting diagram has some; eg in free monoid terms:

$$((), \pi_1, \ldots, \pi_{r-1})^- = (\pi_1, \ldots, \pi_{r-1})$$
$$(\pi_0, \ldots, \pi_{r-1})^- = (\pi_0^-, \ldots, \pi_{r-1})$$

is a nice straightforward one, corresponding (in tree terms) to following the leftmost branch to its end, and removing the leaf reached.

Show how realisation is pushout:

$$\begin{tikzcd}
y(i) \arrow[r] \arrow[d] & y(i-1) \arrow[d] \\
\widehat{\pi^-} \arrow[r] & \widehat{\pi} \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}$$

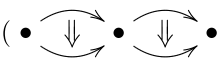


Operads in monoidal categories. Define: globular object. Define: operad! Endomorphism operad!

Endomorphism operads, explicitly. In $\mathbf{FibSpans}(\mathcal{C})$, a globular object $X_\bullet$ is...

Then for $\pi \in T1_n$, the peak of the n-span $(X_\bullet)\pi$ is the object

$$X^\pi := \lim_{c \in \int \hat{\pi}} X_{\dim c}$$

This somewhat cryptic formula is perhaps best illuminated by a couple of examples: if $\pi =$ (), then

$$X_\pi \quad := \quad \lim \left(\begin{tikzcd}
 & X_1 \arrow[dl, "s"'] \arrow[dr, "t"] & & X_1 \arrow[dl, "s"'] \arrow[dr, "t"] & \\
X_0 & X_2 \arrow[u, "s"'] \arrow[d, "t"] & X_0 & X_2 \arrow[u, "s"'] \arrow[d, "t"] & X_0 \\
 & X_1 \arrow[ul, "s"] \arrow[ur, "t"'] & & X_1 \arrow[ul, "s"] \arrow[ur, "t"'] &
\end{tikzcd} \right)$$
$$\cong \quad X_2 \times_{X_0} X_2,$$

giving the object of 0-composable pairs of 2-cells in X. Similarly, if π is the basic n-cell, then $X_\pi = X_n$.

4.2. Describe the gloulbar hom-functor between n-objects [**Bat98**, 3.6] and hence into and out of globular objects. Note: in the case of **Spans**, this is monoidal. (Check: Batanin doesn't show this anywhere, does he? although he does use it! Or, better, is it in Weber somewhere, or easily deducible?)

Really, would like a Yoneda lemma: for a globular object $A_\bullet$ of a cat (with fibrations) $\mathcal{C}$, $\mathrm{End}_{\mathcal{C}}(A_\bullet) \cong \mathrm{End}_{[\mathcal{C}, \mathbf{Sets}]}(\mathbf{y}A_\bullet)$.

Example 4.3. Fundamental ω-groupoid of a space

Connection with Leinster approach. Recall from Weber paper.

Enriched point of view. TODO: the below is from a different earlier approach; fold this into the current presentation!

Definition 4.4 (Endomorphism operads)**.** For $\mathcal{E}$ any category, $X_\bullet$ any globular operad in $\mathcal{E}$, we write $\mathrm{End}_\mathcal{E}(X_\bullet)$ for the operad (construction… either by monoidal globular categories, or by "representable" style of my previous paper; latter is slicker here, but seems very difficult for showing the functoriality). More generally, really want the **Coll**-enriched category structure on (appropriate subcategory of) $[\mathbb{G}, \mathcal{E}]$.

Proposition 4.5. *If $\mathcal{E}$ has enough limits, then for any pasting diagram π,*

$$\mathrm{End}_\mathcal{E}(X_\bullet)(\pi) \cong [\mathbb{G}/n, \mathcal{E}](X \pitchfork \hat{\pi}, X \pitchfork \mathbf{y}(n))$$

where the right hand side consists of "pylon diagrams":

draw the diagram here.

Proof. Straightforward (in either construction of End). □

Proposition 4.6. $\mathrm{End}_\mathcal{E}(X_\bullet)$ *is functorial in $\mathcal{E}$: a functor $F : \mathcal{E} \longrightarrow \mathcal{F}$ preserving appropriate limits induces a map* $\mathrm{End}_\mathcal{E}(X_\bullet) \longrightarrow \mathrm{End}_\mathcal{F}(FX_\bullet)$.

Proof. Straightforward in the "monoidal globular categories" approach. Can't currently see how to do it in the "representable" approach!? □

Definition 4.7. An *algebra* for an operad P on an object $X_\bullet$ of $\mathcal{E}$ is an operad map $\xi\colon P \longrightarrow \mathrm{End}_\mathcal{E}(X_\bullet)$ (the *action* of P on $X_\bullet$); a map of P-algebras is a globular map $\vec{f}\colon X_\bullet \longrightarrow Y_\bullet$ commuting with the action maps, i.e. such that the square

$$\begin{tikzcd} P \arrow[r, "\xi"] \arrow[d, "\upsilon"] & {[X_\bullet, X_\bullet]} \arrow[d, "f\cdot"] \\ {[Y_\bullet, Y_\bullet]} \arrow[r, "\cdot f"] & {[X_\bullet, Y_\bullet]} \end{tikzcd}$$

commutes.

In enriched terms, the resulting category $\mathbf{Alg}_\mathcal{E}(P)$ is $\mathcal{V}\text{-}\mathbf{Cat}(\mathbf{Coll})(P, [\mathbb{G}, \mathcal{E}])$.

We can also consider P as defining a certain finite-limit sketch $\mathrm{Sk}(P)$, and compare the internal algebras defined here with models of this sketch in $\mathcal{E}$.

Proposition 4.8. $\mathbf{Alg}_\mathcal{E}(P) \simeq \mathbf{CmpSpanMod}_\mathcal{E}(\mathrm{Sk}(P))$

APPENDIX C

Background: “Homotopical” higher categories

1. Algebraic Model Structures

1.1. A natural weak factorisation system is dots

1.2. 2. Quasi-categories

Bibliography

[AW09] Steve Awodey and Michael A. Warren, *Homotopy theoretic models of identity types*, Math. Proc. Cambridge Philos. Soc. **146** (2009), no. 1, 45–55.

[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.

[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 MR1474535 (98e:03040)

[Gar08] Richard Garner, *Homomorphisms of higher categories*, submitted, 2008.

[Gar09a] ———, *Two-dimensional models of type theory*, Math. Structures Comput. Sci. **19** (2009), no. 4, 687–736.

[Gar09b] ———, *Understanding the small object argument*, Appl. Categ. Structures **17** (2009), no. 3, 247–285. MR MR2506256 (2010a:18005)

[GG08] Nicola Gambino and Richard Garner, *The identity type weak factorisation system*, Theoret. Comput. Sci. **409** (2008), no. 1, 94–109.

[GvdB08] Richard Garner and Benno van den Berg, *Types are weak ω-groupoids*, submitted, 2008.

[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. MR MR1629519

[Jac93] Bart Jacobs, *Comprehension categories and the semantics of type dependency*, Theoret. Comput. Sci. **107** (1993), no. 2, 169–207. MR MR1201808 (94i:18009)

[Jac99] ———, *Categorical logic and type theory*, Studies in Logic and the Foundations of Mathematics, vol. 141, North-Holland Publishing Co., Amsterdam, 1999.

[Lum09] Peter LeFanu Lumsdaine, *Weak ω-categories from intensional type theory*, Typed lambda calculi and applications (Berlin), Lecture Notes in Computer Science, vol. 5608, Springer, 2009, pp. 172–187.

[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 MR0387009 (52 #7856)

[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.

[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.

[Str00] Ross Street, *The petit topos of globular sets*, Journal of Pure and Applied Algebra **154** (2000), 299–315.