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

Universal-algebraic aspects

Possibly fold this chapter into the next??

1. A category of Type Theories

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

1.2. 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 *embaras de richesse* of frameworks: for instance [**Jac93**], [**Pit00**], [**Hof97**] and [**?**] 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: do this! Mention all the definitions above, at least briefly; and discuss stratified/reachable versions.)

Definition 1.3. A *split full comprehension category*[**Jac93**] (FSCC) is a category $\mathcal{C}$ together with a split fibration $p : \mathcal{E} \to \mathcal{C}$ and a factorisation $p = \mathrm{cod} \cdot \mathcal{P} : \mathcal{E} \to \mathcal{C}^{\to} \to \mathcal{C}$, such that (a) $\mathcal{P}$ maps cartesion 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 $F : \mathcal{C} \to \mathcal{C}'$ and a map of fibrations $p \to p'$ over F, commuting with the factorisations $\mathcal{P}$, $\mathcal{P}'$. A *map of comprehension structures* on $\mathcal{C}$ is just the case $F = 1_C$.

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

Example 1.4. For any type theory $\mathcal{T}$, its category of context $\mathcal{C}(\mathcal{T})$ 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 $\Gamma, x : A \to \Gamma$.

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.

1.5. Categories with attributes

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

1.6. 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.

1.7. 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 1.8. An FSCS *with units* on a category $\mathcal{C}$ is an FSCS $(p, \mathcal{P}_0)$ together with a strict map of FSCS's $1 : (1_B, id) \to (p, \mathcal{P}_0)$.

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

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 1.9 (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 $f^* : p^{\mathsf{cxt}}(\Gamma) \to p^{\mathsf{cxt}}(\Delta)$ is similarly defined in terms of pullback in p.

This is the object part of an evident functor $(-)*$ 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.)

1.10 (The nice slice). Even if $(\mathcal{C}, p, \mathcal{P})$ was stratified, $(\mathcal{C}, p^{\mathsf{cxt}}, \mathcal{P}^{\mathsf{cxt}})$ will generally not be: extending a context by a dependent context may increase its length by more than 1!

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 corresponds type-theoretically to working in context Γ.

Definition 1.11. An *elim-structure* on a map $f : \Xi \to \Theta$ is a function E, assigning to each $C \in p(\Theta)$ and each map $d : \Xi \to \Theta.C$ over Θ a section $E(C, d) : \Theta \to \Theta.C$ 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

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

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

Definition 1.12. A *Frobenius elim-structure* on a map $f : \Xi \to \Theta$ is a choice of elim-structure E_Δ on $(f.\Delta) : \Xi.(f^*\Delta) \to \Theta.\Delta$, 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 1.13. 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 $r_A \colon \Gamma.A \to \Gamma.A.A.\mathrm{Id}_A$ lifting the diagonal (contraction) map $\delta_A \colon \Gamma.A \to \Gamma.A.A$ over Γ

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

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

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

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

(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 1.14. *If $\mathcal{T}$ is any DTT with* Id*-types, then $\mathbf{cl}(\mathcal{T})$ admits a canonical* Id*-structure. Conversely, if $\mathcal{T}$ is any (plain or stratified) category with attributes, then $\mathbf{th}(\mathcal{C})$ admits an interpretation of the* Id*-rules; and the maps $\epsilon_{\mathcal{C}} \colon \mathbf{cl}(\mathbf{th}(\mathcal{C})) \to \mathcal{C}$ and $\eta \colon \mathcal{T} \to \mathcal{T}'$ 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 1.15 (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 [**?**, 2.3.1] for details. By Proposition 1.14, we may work type-theoretically.
(TODO: is this worth the notation that it requires developing?) □

1.16. The coslice construction

2. Internal algebras for operads

(This should probably be a separate chapter.)

2.1. Give fuller account of what I rush through in my previous paper: show correspondence btn different notions of algebras for an operad! (a) models of ess. alg. (Lawvere) theory (poss with extra structure: "P-maps"); (b) Batanin: monoidal globular categories (as used + nicely expounded in [GvdB]); (c) Leinster: (weak) T-structured categories.
Lovely rarely-cited WEBER paper gives source for most of this! Possibly even *everything* I need is there, in which case possibly move this section to appendix, and fold the first part of this chapter into the next chapter???

DEFINITION 2.2 (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 2.3. *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 2.4. $\mathrm{End}_{\mathcal{E}}(X_\bullet)$ *is functorial in* $\mathcal{E}$*: a functor* $F : \mathcal{E} \to \mathcal{F}$ *preserving appropriate limits induces a map* $\mathrm{End}_{\mathcal{E}}(X_\bullet) \to \mathrm{End}_{\mathcal{F}}(FX_\bullet)$.

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

Definition 2.5. An *algebra* for an operad P on an object $X_\bullet$ of $\mathcal{E}$ is an operad map $\xi\colon P \to \mathrm{End}_{\mathcal{E}}(X_\bullet)$ (the *action* of P on $X_\bullet$); a map of P-algebras is a globular map $\vec{f}\colon X_\bullet \to Y_\bullet$ 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 $\mathcal{V}$-$\mathbf{Cat}(\mathbf{Coll})(P, [\mathbb{G}, \mathcal{E}])$.

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

CHAPTER 3

Globular structures from type theory

1. The fundamental weak omega-groupoid of a type

Update of my prev paper (+ more detailed comparison with Richard + Benno): type gives internal weak omega-groupoid in the classifying category.

Put here: an informal overview as in the paper!

1.1. functor $\mathbf{FibSpans} : \mathbf{Th}^{\mathrm{Id}} \to \mathbf{MonGlobCat}$: imitate the original Batanin "higher spans" construction, but using doubly-dependent types instead of spans.

1.2. now for $\mathcal{C} \in \mathbf{Th}^{\mathrm{Id}}$, get for each context $X \in \mathcal{C}$ a globular element of $fibSpans(\mathcal{C})$.

2. The classifying weak omega-category of a type theory

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?

0.1. 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 $\mathbb{C}\lessdot_\omega(\mathcal{T})$, 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 $Clw(\mathcal{T})$!

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

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

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

There are two obvious options. Firstly, we could construct $\mathbb{C}\lessdot_\omega^{qcat}(\mathcal{T})$ directly from the theory $\mathcal{T}$. [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. (The intriguing approach of re-axiomatising Id-types to be “naturally simplicial in shape” has, however, 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

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

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.

0.3. 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 [**Gar09**], [**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 $D(\operatorname{ob}\mathbb{G}) \to \mathbb{G} \to \widehat{\mathbb{G}} \to \mathbf{wk}\text{-}n\text{-}\mathbf{Cat}$ 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 $\mathcal{I} \to \mathcal{E}^{\mathbf{2}}$. 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. $\mathbb{F}_t\text{-}\mathbf{Alg} \Leftrightarrow \mathbb{F}\text{-}\mathbf{Alg} \times_{\mathcal{E}^{\mathbf{2}}} \mathbb{W}\text{-}\mathbf{Alg}$. Give: $\mathbb{W} \to \mathcal{E}^{\mathbf{2}}$ "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 [**?**]. 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 $\mathsf{term}^n \to \mathsf{term}$, $\mathsf{term}^n \to \mathsf{type}$, 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:

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

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

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

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

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

Rules for context maps:

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

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

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. Weak higher categories

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.

2.4. Equivalent (by Weber): Batanin presentation!

Definition 2.5.

2.6. Contractibility.

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.

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

[Gar09] ———, *Understanding the small object argument*, Appl. Categ. Structures **17** (2009), no. 3, 247–285. MR MR2506256 (2010a:18005)

[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.

[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.