INTERNAL LANGUAGES OF HIGHER CATEGORIES

CHRIS KAPULKIN

I recently posted the following preprint to the arXiv:

- Locally cartesian closed quasicategories from type theory, arXiv:1507.02648.

Btw if you're not the kind of person who likes to read mathematical papers, I also gave a talk about the above mentioned work in Oxford, so you may prefer to watch it instead. (-:

I see this work as contributing to the idea/program of *HoTT as the internal language of higher categories*. In the last few of weeks, there has been a lot of talk about it, prompted by somewhat provocative posts on Michael Harris' blog.

My goal in this post is to survey the state of the art in the area, as I know it. In particular, I am not going to argue that internal languages are a solution to many of the problems of higher category theory or that they are not. Instead, I just want to explain the basic idea of internal languages and what we know about them as far as HoTT and higher category theory are concerned.

Disclaimer. The syntactic rules of dependent type theory look a lot like a multi-sorted essentially algebraic theory. If you think of sorts called *types* and *terms* then you can think of rules like Σ-types and Π-types as *algebraic operations* defined on these sorts. Although the syntactic presentation of type theory does not quite give an algebraic theory (because of complexities such as variable binding), it is possible to formulate dependent type theory as an essentially algebraic theory. However, actually showing that these two presentations are equivalent has proven complicated and it's a subject of ongoing work. Thus, for the purpose of this post, I will take dependent type theories to be defined in terms of contextual categories (a.k.a. C-systems), which are the models for this algebraic theory (thus leaving aside the Initiality Conjecture). Ultimately, we would certainly like to know that these statements hold for syntactically-presented type theories; but that is a very different question from the ∞-categorical aspects I will discuss here.

A final comment before we begin: this post derives greatly from my (many) conversations with Peter Lumsdaine. In particular, the two of us together went through the existing literature to understand precisely what's known and what's not. So big thanks to Peter for all his help!

Internal languages of categories.

First off, what is the internal language? Without being too technical, let me say that it is typically understood as a correspondence:

$$\begin{tikzcd} \{\text{type theories}\} \arrow[r, shift left, "\mathrm{Cl}"] & \{\text{categories}\} \arrow[l, shift left, "\mathrm{Lang}"] \end{tikzcd}$$

On the right hand side of this correspondence, we have a category of categories with some extra structure and functors preserving this structure. On the left hand side, we have certain type theories, which are extensions of a fixed core one, and their interpretations (which are, roughly speaking, maps taking types to types and terms to terms, preserving typing judgements and the

Date: July 28, 2015.

constructors of the core theory). Notice that the core theory is the initial object in the category of theories in the above picture.

The functor Cl takes a theory to its initial model, built directly out of syntax of the theory: the objects are contexts and the morphisms are (sequences of) terms of the theory (this category is often called the *classifying category*, hence the notation Cl). The functor Lang takes a category to the theory whose types are generated in a suitable sense by the objects and whose terms are generated by the morphisms of the category. In particular, constructing $\mathrm{Lang}(C)$ for some category C from the right hand side is the same as establishing a model of the core theory in C.

Finally, these functors are supposed to form some kind of an adjoint equivalence (with $\mathrm{Cl} \dashv \mathrm{Lang}$), be it an equivalence of categories, bi-equivalence, or ∞-equivalence, depending on whether the two sides of the correspondence are categories, 2-categories, or ∞-categories.

The cleanest example of this phenomenon is the correspondence between λ-theories (that is, theories is simply typed λ-calculus) and cartesian closed categories:

$$\left\{\begin{array}{c}\text{theories in}\\ \lambda\text{-calculus}\end{array}\right\} \underset{\text{Lang}}{\overset{\text{Cl}}{\rightleftarrows}} \left\{\begin{array}{c}\text{cartesian closed}\\ \text{categories}\end{array}\right\}$$

which you can read about in Part I of the standard text by Jim Lambek and Phil Scott.

Extensional Martin-Löf Type Theory.

Unfortunately, as soon as we move to dependent type theory, things are getting more complicated. Starting with the work of Robert Seely, it has become clear that one should expect Extensional Martin-Löf Type Theory (with dependent products, dependent sums, and extensional Identity types) to be the internal language of locally cartesian closed categories:

$$\left\{\begin{array}{c}\text{Extensional Martin-Löf}\\ \text{type theories}\end{array}\right\} \underset{\text{Lang}}{\overset{\text{Cl}}{\rightleftarrows}} \left\{\begin{array}{c}\text{locally cartesian}\\ \text{closed categories}\end{array}\right\}$$

Seely overlooked however an important coherence problem: since types are now allowed to depend on other types, we need to make coherent choices of pullbacks. The reason for that is that type-theoretically substitution (into a type) is a strictly functorial operation, whereas its categorical counterpart, pullback, without making any choices, is functorial only up to isomorphism. (If you find the last sentence slightly too brief, I recommend Peter Lumsdaine's talk explaining the problem and known solutions.) The fix was later found by Martin Hofmann; but the resulting pair of functors does not form an equivalence of categories, only a biequivalence of 2-categories, as was proven in 2011 by Pierre Clairambault and Peter Dybjer.

Intensional Martin-Löf Type Theory and locally cartesian closed ∞-categories.

Next let us consider Intensional Martin-Löf Type Theory with dependent products and sums, and the identity types; additionally, we will assume the (definitional) eta rule for dependent functions and functional extensionality.

Such type theory has been, at least informally, conjectured to be the internal language of locally cartesian closed ∞-categories and thus, we expect the following correspondence:

$$\left\{\begin{array}{c}\text{Intensional Martin-Löf}\\ \text{type theories}\end{array}\right\} \underset{\text{Lang}}{\overset{\text{Cl}}{\rightleftarrows}} \left\{\begin{array}{c}\text{locally cartesian}\\ \text{closed }\infty\text{-categories}\end{array}\right\}$$

where the functors Cl and Lang should form an adjunction (an adjoint $(\infty, 1)$-equivalence? or maybe even $(\infty, 2)$-equivalence?) between the type-theoretic and categorical sides.

Before I summarize the state of the art, let me briefly describe what the two functors ought to be. Starting with a type theory, one can take its underlying category of contexts and regard it as category with weak equivalences (where the weak equivalences are syntactically-defined equivalences), to which one can then apply the simplicial localization. This gives a well-defined functor from type theories to ∞-categories. Of course, it is a priori not clear what the (essential) image of such a functor would be.

Conversely, given a locally cartesian closed ∞-category C, one can look for a category with weak equivalences (called the *presentation* of C) whose simplicial localization is C and then try to establish the structure of a categorical model of type theory on such a category.

What do we know? The verification that Cl takes values in locally cartesian closed ∞-categories can be found in my paper. The other functor is known only partially. More precisely, if C is a *locally presentable* locally cartesian closed ∞-category, then one can construct $\mathrm{Lang}(C)$. As mentioned above, the construction is done in two steps. The first, that is presenting such an ∞-category by a type-theoretic model category (which is, in particular, a category with weak equivalences) was given by Denis-Charles Cisinski and Mike Shulman in these blog comments, and independently in Theorem 7.10 of this paper by David Gepner and Joachim Kock. The second step (the structure of a model of type theory) is precisely Example 4.2.5.3 of the "local universe model" paper by Peter Lumsdaine and Michael Warren.

What don't we know? First off, how to define $\mathrm{Lang}(C)$ when C is not locally presentable and whether the existing definition of $\mathrm{Lang}(C)$ for locally presentable quasicategories is even functorial? We also need to understand what the *homotopy theory of type theories* is (if we're hoping for an equivalence of homotopy theories, we need to understand the homotopy theory of the left hand side!)? In particular, what are the weak equivalences of type theories? Next in line: what is the relation between Cl and Lang? Are they adjoint and if so, can we hope that they will yield an equivalence of the corresponding ∞-categories?

Univalence, Higher Inductive Types, and (elementary) ∞-toposes.

Probably the most interesting part of this program is the connection between Homotopy Type Theory and higher topos theory (HoTT vs HTT?). Conjecturally, we should have a correspondence:

$$\begin{tikzcd}
\left\{\begin{array}{c}\text{Homotopy}\\ \text{Type Theories}\end{array}\right\} \arrow[r, shift left, "\mathrm{Cl}"] & \left\{\begin{array}{c}\text{elementary}\\ \infty\text{-toposes}\end{array}\right\} \arrow[l, shift left, "\mathrm{Lang}"]
\end{tikzcd}$$

This is however not a well-defined problem as it depends on one's answer to the following two questions:

(1) What is HoTT? Obviously, it should be a system that extends Intensional Martin-Löf Type Theory, includes at least one, but possibly infinitely many univalent universes, and some Higher Inductive Types, but what exactly may largely depend on the answer to the next question...

(2) What is an elementary ∞-topos? While there exist some proposals (for example, that presented by André Joyal in 2014), this question also awaits a definite answer. By analogy with the 1-categorical case, every Grothendieck ∞-topos should be an elementary ∞-topos, but not the other way round. Moreover, the axioms of an elementary ∞-topos should imply (maybe even explicitly include) that it is locally cartesian closed and that it has finite colimits, but should not imply local presentability.

What do we know? As of today, only partial constructions of the functor Lang exist. More precisely, there are:

- Theorem 6.4 of Mike Shulman's paper contains the construction of $\mathrm{Lang}(C)$ if C is a Grothendieck ∞-topos that admits a presentation as simplicial presheaves on an elegant Reedy category and HoTT is taken the extension of Intensional Martin-Löf Type Theory but as many univalent universes á la Tarski as there are inaccessible cardinals greater than the cardinality of the site.
- Remark 1.1 of the same paper can be interpreted as saying that if one considers HoTT with weak (univalent) universes instead, then the construction of $\mathrm{Lang}(C)$ works for an arbitrary ∞-topos C.
- A forthcoming paper by Peter Lumsdaine and Mike Shulman will supplement the above two points: for some reasonable range of higher toposes, the resulting type theory $\mathrm{Lang}(C)$ will be also shown to possess certain Higher Inductive Types (e.g. homotopy pushouts and truncations), although the details remain to be seen.

What don't we know? It still remains to define $\mathrm{Lang}(C)$ outside of the presentable setting, as well as give the construction of Cl in this case (or rather, check that the obvious functor from type theories to ∞-categories takes values in higher toposes). The formal relation between these functors (which are yet to be defined) remains wide open.