A note on (locally) presentable categories

Ambroise LAFONT

September 20, 2019

Contents

A *locally presentable category* (or just *presentable* category) is a category which is not too big and not too small: it is cocomplete, but is the generated by a small set of objects. The main reference for this beast is [1].

A morphism between presentable categories is defined as cocontinuous functor in Anel's and Joyal's Topo-logie.

Presentable categories can be defined as cocomplete *accessible categories*, or as complete accessible categories (after nlab, the page on accessible categories and locally presentable categories).

An accessible category is defined as a λ-accessible category (you don't need to know what it is at this stage) for some regular cardinal λ. This a notion of λ-presentable category, and a presentable categories is a λ-presentable category, for some regular cardinal λ. Any λ-presentable category is also μ-presentable for any regular cardinal $\mu > \lambda$ (this is not the case for accessible categories!).

Presentable categories satisfy nice properties.

Definition 0.1 *A functor is said* ***accessible*** *if it is* ***λ-accessible*** *for some regular cardinal λ, that is, if it preserves λ-directed colimits.*

Theorem 0.2 *[1, 1.66] A functor between presentable categories is right adjoint iff it is continuous and accessible.*

A similar statement is true of left adjointness, according to nlab.

Theorem 0.3 (nlab) *Let T be an accessible monad on a presentable category C. Then C^T is presentable.*

1 Presentable categories

Let λ be a regular cardinal. A λ-presentable category is a category C which is a free λ-cocompletion $\mathrm{Cont}_\lambda(\mathrm{Pres}_\lambda(C))$ of some full subcategory $\mathrm{Pres}_\lambda(C)$ of *λ-presentable objects*.

Here, by free λ-cocompletion, we think informally of an adjunction between cocomplete (not λ-cocomplete!) categories and the category of categories and λ-cocontinous functors (i.e., functors preserving any λ-small colimit, not only the directed ones).

2 Finitely presentable categories

Any presheaf category is finitely presentable [1]: the representable functors form the strong generator, and thus the presentable objects are the finite limits of these representable functors.

3 Gabriel-Ulmer

This section is based on Exercise 1.s and Remark 1.46 of [1].

Theorem 3.1 *For each regular cardinal λ, we denote*

- *$c_\lambda CAT$ the category of small categories λ-complete and functor preserving these limits;*
- *$lp_\lambda CAT$ the category of λ-presentable categories and continuous and λ-accessible functors.*

Then, the following functors are equivalences:

$$\begin{aligned} R : c_\lambda CAT &\to lp_\lambda CAT \\ C &\mapsto Cont_\lambda C^o \\ L : lp_\lambda CAT &\to c_\lambda CAT \\ C &\mapsto Pres_\lambda(C)^o \end{aligned}$$

Actually, they even induce a biequivalence ($c_\lambda CAT$ and $lp_\lambda CAT$ can be equipped with a 2-categorical structure).

Question 3.2 *Is it a strict 2-equivalence? I guess no, otherwise it would have been said.*

This theorem induces two (equivalent) contravariant functors from the poset of regular cardinals (seen as a category) to the category of small categories. Performing the Grothendieck construction yield an equivalence between some category of presentable categories and some category of small categories (fibered over the category of regular cardinals).

4 Sketches

Definition 4.1 *A* ***sketch*** *is a small category C with a family of (small) diagrams $F_i : D_i \to C$ together with a choice of a cone or a cocone for each of this diagram.*

Definition 4.2 *The* ***category of models of a sketch*** $(C, (F_i, c_i)_i)$ *is the full subcategory of $[C, Set]$ mapping any mapping any chosen cones and cocones to limits and colimits.*

Theorem 4.3 *Any accessible category is equivalent to the category of models of a sketch.*

Theorem 4.4 *Any presentable category is equivalent to the category of models of a* ***limit sketch****, that is, a sketch with only cones (and no cocones).*

Question 4.5 *How do you retrieve the regular cardinal from this description? Probably, it is some cardinal bound over the size of the diagrams?*

Theorem 4.6 *(nlab) A category is λ-accessible iff it is equivalent to a full subcategory of a presheaf category closed under λ-filtered colimits.*

Theorem 4.7 *A category is if it is equivalent to a reflective subcategory of a presheaf category.*

Note that a λ-accessible category is a λ-presentable in case the embedding into the presheaf category has a left adjoint.

Theorem 4.8 *[1, Corollary 2.62] For any sketch, there exists another one whose cones/cocones are all limiting/colimiting and has an equivalent category of models.*

By Gabriel-Ulmer duality, we already know that for limit sketches: you can take the λ-complete category corresponding to the presentable category and take all the λ-small limits as cones.

Question 4.9 *What does exactly this construction on a sketch S do?*

5 GATs and EATs

generalized algebraic theories (or gats) are defined in [2], and also more formally (and recently) in [3]

5.1 QIITs are (particular) GATs

It is not exactly clear what are GATs introduced by [2], in particular what is the status of equations of sorts. GATs are like QIITs but with a possible infinite context. More concretely, a GAT consists of a possibly infinite set of axiomatic judgments, and each judgement must be wellformed (with respect to a finite subset of axioms). Infinitary QIITs allow to capture some of them, but not all.

Question 5.1 *Can we find a counter example?*

Also, in QIITs, we can have equality constructors taking equalities in arguments, that we cannot do directly with GATs. But fortunately, such constructors can be encoded by introducing new types in the universe which are the limits of other ones (such incorporating the equations).

Question 5.2 *[2] claims that GATs are equivalent to EATs, but how to deal with equations of sorts? How are they converted in the EAT version ?*

Cartmell also proves that they are equivalent to contextual categories, and for this he needs these sorts equations (see (iii) at the top of p107 of his PhD manuscript).

Taichi sees two difficulties when trying to replace sort equations with isomorphisms:

- If we replace a sort equation $A = B$ with a sort isomorphism $e : A \simeq B$, we also have to replace occurences of coercion along $A = B$ with applications of e.
- A naive replacement can create a different GAT. For example, consider the GAT consisting of a constant sort A and equation $A = A$ (although this equation is redundant, it is a valid GAT). A model of $(A, A = A)$ is just a set, but a model of $(A, e : A \simeq A)$ is a set equipped with an automorphism. They do not seem to be equivalent.

I expect that 1. is possible, although I confess I don't know how to do it in general. For the second point, I would suggest the following: given any sort equation $A = A$ that you can prove in the theory induced by a GAT with sort equations, you get (I hope) an automorphism on the translation of A (in the translated GAT which turns sort equations into isomorphisms). Then, for each such sort equation, add in the translated GAT that this automorphism should be the identity.

5.2 QIITs are QITs (as GATs are EATs)

Example: $Con : \mathcal{U}, Ty : Con \to \mathcal{U}$ is translated as $Con : \mathcal{U}, \int Ty : \mathcal{U}, con : \int Ty \to Con$ (this is the idea of translating GATs into EATS, as in [2]).

Question 5.3 *Maybe this generalizes to HIITs and HITs?*

References

[1] J. Adamek and J. Rosicky. *Locally Presentable and Accessible Categories.* London Mathematical Society Lecture Note Series. Cambridge University Press, 1994.

[2] John Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209 – 243, 1986.

[3] Richard Garner. Combinatorial structure of type dependency. *Journal of Pure and Applied Algebra*, 219(6):1885 – 1914, 2015.