Review of the doctoral thesis by András Kovács: type-theoretic signatures for algebraic theories and inductive types

Christian Sattler

September 2, 2022

This thesis is an important contribution to the type-theoretic study of algebraic theories, in the broadest sense. It proposes a novel framework for describing signatures of such theories, defines their semantics in terms of categories of algebras, and proves several new and important results about the existence of initial objects in these categories, that is, inductive types (in the broadest sense, including quotient and higher inductive-inductive types).

The chosen topic is of extreme importance both for the theory of formal systems based on type theory and the practical implementation of proof assistants built on dependent type theory. In predicative, constructive formal systems, inductive constructions generally fall into two classes:

- rigid combinators such as W-types and quotient types,
- limited and often ad hoc schemes for or types of codes parametrizing inductive constructions.

The theory of signatures proposes by the thesis provides a much more flexible framework for the specification of inductive types, while at the same time ensuring semantic adequacy via the careful study of the universal property inherent to these types in a category of algebras. Such adequacy results are currently lacking in popular proof assistants such as Agda.

The research methods are up to date and conform to the standards of leading research in type theory. Features and limitations of approaches are discussed in detail. Comparisons to pre-existing notions, even outside the immediate area of type theory, are made and related work is discussed in a satisfactory manner.

The thesis is well-structured are written clearly and in a pedagogical style. It consists of a series of chapters that progressively introduce more complicated kinds of algebraic theories and settings, starting from single-sorted algebraic theories in the traditional sense (with no use of dependent types in the signatures) and going all the way to higher inductive-inductive types in homotopy type theory. Each chapter investigates the models and initial objects (inductive types) for signatures of the kind of algebraic theory under consideration.

We proceed with a short summary of each chapter and a discussion of the novel results contained.

Chapter 2

After the introduction, a pedagogical “toy case” or motivational chapter (Chapter 2) illuminates the approach of the full framework developed in later chapters. By intentionally restricting to closed, single-sorted, finitary signatures, complex dependency issues are removed from the picture. Targeting the simplest case, the reader is allowed to focus on the essential new concept of a theory of signatures, the initial object in a category of cwfs (categories with families) with carefully chosen type formers. This simplified setting still allows the development of toy instances of the technology that will be fully developed in the later chapters such as:

(1) notions of algebras, algebra morphisms, displayed algebras, and sections thereof,
(2) the equivalence between initiality and induction,
(3) the construction of initial algebra from the existence of the syntax of the theory of signatures.

All of these are novel results in this framework.

Of note, the constructions in Chapter 2 are fully formalized in Agda. I have checked the formalization.

Chapter 3

This chapter redevelops the concepts of the previous chapter in a systematic manner and more general setting. A distinction between the metatheory and an object theory relative to which an algebraic theory is to be interpreted is made via the use of two-level type theory (2LTT). The setting of 2LTT forms the backbone of a large part of the thesis material.

Chapter 4

This is the first chapter with a fully fledged theory of signature, the so-called finitary quotient inductive-inductive case. This chapter occupies the largest part of the thesis. The notions of algebras, algebra morphisms, etc. of (1) are bundled up into a finite limit cwf model of the novel theory of signatures. The level of abstraction attained in this manner highlights the quality of the research methods.

The development of the novel model of finite limit cwfs consumes a large part of the chapter and took me some time to check. The same holds for term construction for (3), which only works if the inner theory of 2LTT replicates the features of the outer one. The author is careful here about the use of universe levels to ensure consistency of the construction. Elsewhere, the consideration of size and universe levels is elided, which I consider appropriate as there are no size problems to be expected. Again, the versions of (1), (2), (3) for the theory of signatures of this chapter are novel results.

There is an interesting discussion on the possibility of "bootstrapping" the category of models of the theory of signatures. For this, the author considers the theory of (closed) signatures to be itself specified by a signature. (The restriction to closed signatures is lifted in Chapter 5.)

The chapter also contains an important and novel result on the reduction of FQIITs to W-types and quotients.

Chapter 5

This chapter lifts the finiteness restriction in the theory of signatures considered in the previous chapter. The price to pay is a more complicated semantics based on so-called weak morphisms of cwfs that do not preserve the empty context and context extension strictly. The resulting theory of signatures is now powerful enough to cover all practically desired examples of algebraic theories. In particular, it is powerful enough to bootstrap its own theory of signatures. The framework is again novel and the author again manages to prove the novel versions of (1), (2), (3) in this setting.

The author gives informal reasons for why the reduction to combinators such as W-types and quotients does not seem possible in this setting.

Chapter 6

The last chapter examines theories of signatures in a setting where the equality of the inner theory no longer reflects that of the outer one. Such is the case in models of homotopy type theory (univalent foundations). The quotient features in the previous setting turns into the possibility for so-called higher constructors in inductive types. The dichotomy between the two equalities leads to schism in the notion of theory of signatures: the issue is whether morphisms of algebras should preserve structure up to inner (weak) or outer (strict) equality. The author discusses both kinds of signatures and develops semantics (1) for both of them. All of these are novel contributions.

Concluding remarks

The thesis is well-written and generally complies with the spelling and grammatical rules of professional English language. The writing is clear and intelligible.

The two attached publications:

- Signatures and induction principles for higher inductive-inductive types,
- Constructing quotient inductive-inductive types

adequately contain new findings that are included in the dissertation. The thesis contains further generalizations building on the material of these and other publications of the author. The additions are clearly marked.

The English summary is well-written and gives a coherent summary of the content of the thesis.

The thesis can be accepted in its current form. I recommend that the dissertation receive the qualification **summa cum laude**.

Specific comments

Below, I note some specific questions and comments for the author. The below comments fall into several classes:

- typos and grammar mistakes,
- remarks on notation,
- questions where I do not understand something,
- mathematical remarks relevant to the presentation,
- side-remarks that are connected to the material, but are not intended as writing suggestions.

None of these comments prevent the thesis from being accepted in the current form.

New comments

- In Subsection 3.2.3, you write:

  > Σ is equivalent to negative Σ, although it only supports propositional η-rules. In contrast, positive identity is usually not equivalent to negative identity.

  I no longer understand this. Case distinction over what you could mean by equivalence here:

  - If both positive and negative type formers exist, they are equivalent (up to propositional equality). This holds both for Σ-types and identity types.
  - If we have the positive type former, we can derive the negative type former. This holds neither for Σ-types nor identity types.
  - That leaves: theories with the positive type former are in some sense equivalent to theories with the negative type former. Working this out precisely is current research, I guess.

- Adding a notion of displayed object to finitely complete categories does not immediately give your notion of finitely complete cwfs. What we get is, for each object A in $\mathbb{C}$:

  - a type DisplayedObj(A) with a map extension : DisplayedObj(A) → Obj(C ↓ A). (Note that there is no action under substitution in A.)

  This notion is probably sufficient to model the induction principle up to isomorphism. To get the exact induction principle, we also have to add a type of sections of B : DisplayedObj(A):

  - a type Section(B) with an isomorphism Section(B) ≃ section of extension(B).

  This seems to be enough for the basic interpretation. To able to have equivalence of initiality and induction, we need to add that extension induces an equivalences between the category of displayed objects over A and C ↓ A. This means: for each map into A, a displayed object over it with isomorphic extension. So constant families in each slice (also known as “fibrant replacement”). Note that this gives a substitutional action in A, which however is not strictly functorial.

  Note that if we add strictly functorial substitutional action in A to the above, this would be isomorphic to finitely complete cwfs (minus a terminal object). However, I think strict functoriality of the substitution of displayed objects is not actually needed (although it may

be nice to have). At least, the equivalence between initiality and induction also works without it.

There is a natural notion of weak morphismn that is effectively just a finitely continuous functor with a compatible action on displayed objects. I wonder if this allows for a modified version of the semantics in Chapter 5 that removes the splitness requirement for the cloven isofibrations. I think that there is some version of the semantics that does not require the splitness condition. But maybe more changes would be needed.

- There is something wrong with the grammar of this sentence on page 102:

  > Recently, Fiore, Pitts and Steenkamp showed that a class of infinitary quotient inductive types, called QWI-types, can be reduced to inductive types, quotients the axiom of weakly initial sets of covers (WISC) [FPS21].

- There is a typo on page 146:

  > Alternatively, if have large Σ types in ToS

- In Example 28, the naturality laws for return and bind are missing. Or do they follow?

Regarding previous comments

- I think something got wrangled in this change. A counterexample to what (closed implying finitary in the presence of metatheoretic quantification)?

```
@@ -649,16 +688,16 @@ $t[\id] = t$ for all $t$. Substitution composition is as follows.
 \end{myexample}
 In short, the current ToS allows signatures which are
 \begin{itemize}
-\item \emph{Single-sorted}: this means that we have a single type constructior, correspon
-\item \emph{Closed}: signatures cannot refer to any externally existing type. For example
+\item \emph{Single-sorted}: this means that we have a single type constructor, correspondi
+\item \emph{Closed}: signatures cannot refer to any externally existing type. For example
 \item \emph{Finitary}: inductive types corresponding to signatures are always
-  finitely branching trees. Being closed implies being finitary, since an
-  infinitely branching node would require some external type to index subtrees
-  with. For example, $\mi{node} : (\mathbb{N} \to \iota) \to \iota$ would
-  specify an infinite branching (if such type was allowed in ToS).
+  finitely branching trees. For a counterexample, assuming $\mbb{N}$ as the
+  metatheoretical type of natural numbers, $\mi{node} : (\mathbb{N} \to \iota)
+  \to \iota$ would specify an infinite branching (if such type was allowed in
+  the ToS).
 \end{itemize}

-\emph{Remark.} We omit $\lambda$-expressions from ToS for the sake of
+\emph{Remark.} We omit $\lambda$-expressions from the ToS for the sake of
 simplicity: this causes terms to be always in normal form (neutral, to be
 precise), and thus we can skip talking about conversion rules. Later, starting
 from Chapter \ref{chap:fqiit} we include proper $\beta\eta$-rules in theories of
```

  This relates to my previous comment:

  > ### 2.1
  >
  > - Page 9:
  >
  >   > Being closed implies being finitary.
  >
  >   I disagree. Is it not possible to have closed, infinitary signatures by taking a metatheoretic branching type (rather than a type in the object theory)? For example, semi-lattices with externally countable joins (with suprema over externally indexed lists v_0, v_1, ...). In a setting with 2LTT, allowing for closed and non-finitary signatures would correspond in the theory of signatures to closure

of the universe U under externally-indexed products (external dependent function type). It seems this is not covered by the theories of signatures you discuss in later chapters. Do you comment on this somewhere?

Another kind of closed, non-finitary signature (according to your usage of finitary in Chapter 4) would be quotient signatures with branching type given by the equality type of a type that is being defined. But I understand you are not considering quotients here and that use of equality can be encoded away anyway.

- You forgot to fix this typo:

  - [Typo] Page 17: “f function” should be “function f”.

  Probably because I gave you the wrong page number. It’s at the start of page 13.

- The grammar here still makes it seem as if the usual induction principle has propositional β-rules. I recommend replacing “where [...] are” by “but with [...]”:

```
@@ -943,10 +982,10 @@ We define a predicate which holds if an algebra supports induction.
 \end{mydefinition}

 We can observe that $\Inductive\,\{\ms{NatSig}\}\,(X,\,\ms{zero},\,\ms{suc})$
-computes exactly to the usual induction principle for natural numbers. The input
-$\DispAlg$ is a bundle of the induction motive and the methods, and the output
-$\Section$ contains the $X^S$ eliminator function together with its
-$\beta$-rules.
+computes to the usual induction principle for natural numbers, where
+$\beta$-rules are given as propositional equalities. The input $\DispAlg$ is a
+bundle of the induction motive and the methods, and the output $\Section$
+contains the $X^S$ eliminator function together with its $\beta$-rules.

 \section{Term Algebras}
```

  This relates to my previous comment:

  2.2.4

  - After Definition 8, you say that Inductive {NatSig} (X, zero, suc) corresponds exactly to the usual induction principle for natural numbers (including its β-rules). However, the usual induction principle has the β-rules as strict equalities. Here, they are elements of the intensional identity type. (This is addressed by the 2LTT setting of later chapters.)

- I no longer understand the following in Subsection 3.2.3:

  > Σ is equivalent to negative Σ, although it only supports propositional η-rules. In contrast, positive identity is usually not equivalent to negative identity.

  Case distinction over what you could mean by equivalence here:

  - If both positive and negative type formers exist, they are equivalent (up to propositional equality). This holds both for Σ-types and identity types.
  - If we have the positive type former, we can derive the negative type former. This holds neither for Σ-types nor identity types.
  - That leaves: theories with the positive type former are in some sense equivalent to theories with the negative type former. But working this out precisely is current research, I guess.

- In Subsection 3.2.3, you changed:

```
@@ -1814,12 +1847,8 @@ in the number of $\Bool$ arguments. More generally, Scherer presento
 algorithm for conversion checking with strong finite sums and products in simple
 type theory \cite{scherer17deciding}, which also takes exponential time. If we
 move to natural numbers with definitionally unique recursion, conversion
-checking becomes undecidable.
-
```

```
-One solution is to have propositionally unique recursion instead. However, if
-such equations are postulated, that would break the canonicity property in
-intensional type theories, since now we would have equality proofs other than
-$\refl$ in the empty context.
+checking becomes undecidable, since it would require deciding extensional
+equality of $\Nat$ functions.

 The standard solution is to have dependent elimination principles instead: this
 allows inductive reasoning, canonicity and effectively decidable definitional
```

Part of change is presumably related to my previous comment:

> In the discussion of η for Nat, you say that conversion checking is undecidable. You should provide a reference for that.

You now write that that conversion checking with η requires deciding extensional equality of functions involving the natural numbers. But I don't think that's the case, or at least I don't see how these two things are related. Given f, g : Nat → Nat such that f (S^k 0) and g (S^k 0) are definitional equal for every external natural number k, I don't think it follows that f and g are definitionally equal using η for Nat.

- You made some changes in Subsection 3.4.2 that were triggered by my comment:

  > You say that "by definition" all such type formers transfer from $\mathbb{C}$ to (Ty_0, Tm_0), but there is a step missing that you should fill in. What you get from the what you write is a section of a certain presheaf X over $\mathbb{C}$. However, you have defined type formers for (Ty_0, Tm_0) as structure naturally in a context Γ where Γ now is itself a presheaf over $\mathbb{C}$. Or, if you were to apply the HOAS idea also here, it would be a section of a certain presheaf over Psh($\mathbb{C}$). However, the representing object of that presheaf turns out to be X (why?). Essentially, there is a Yoneda-step missing.

  To say that C has e.g. Π-types, we still have to demand a global section (or something else external) of the "HOAS" presheaf over C. This is because we regard C itself as external to presheaves over C, so we can't stop with an internal formula. You had this in your previous version, but removed it.

  In any case, the point of my original comment was that the specification of e.g. Π-types in C as a global section of this presheaf is different a priori from the specification of Π-types for Ty_0 in presheaves over C. To wit, in Subsection 3.3.1, you say that Π_0 takes a context Γ, a type A over Γ, and so on. But here Γ is a context of presheaves over C, that is, a presheaf over C, and not just an object over C. I guess this can be resolved by observing that a global section of a global type is equivalent to coherent sections of all possible substitutions of that type.

- You made some changes in response to this comment:

  > You say that you do not lose anything by having strict constancy. But it makes the statements a bit weaker that assume a flcfw. With the weaker notion, some of these get nicer. In particular, Theorem 3 could be seen as stating the following. If $\mathbb{C}$ is a flcwf, then so is every slice of $\mathbb{C}$.

  You added "with weak K" or "with weak constant families" in a couple of places, for example Theorem 3. Unfortunately, that makes it seem like an extra hypothesis on a flcwf. It's rather "with strict constant families replaced by weak constant families" (which would be a mouthfull). Maybe add something after Definition 41 to clarify?

  Note also that Definition 44 does not require constant families at all. So it feels especially strange to have "with weak constant family" as extra qualifier.

- In Subsection 4.2.4, you changed:

```
@@ -3606,14 +3680,14 @@ Additionally, type substitution is functorial, i.e.\ $\bs{A[\id]} 
and $\bs{A[\sigma \circ \delta]} \equiv \bs{A[\sigma][\delta]}$. This holds
because the underlying set families are defined by function composition.

-\emph{Remark.} Types could be equivalently defined as slices in the category of
-flcws, and type substitution could be given as pullback, but in that case we
```

```
-would run into the well-known strictness issue, that type substitution is
-functorial only up to isomorphism. This is not a critical issue, as there are
-standard solutions for recovering strict substitutions from weak ones
-\cite{kapulkin12simplicial,local-univ,clairambault2014biequivalence}. But if we
-ever need to look inside the definitions in the model, using displayed algebras
-yields a lot less encoding overhead than strictifying pullbacks.
+\emph{Remark.} Types could be equivalently defined as slices, as objects in
+$\bs{\ms{flcwf}}/\bGamma$, and type substitution could be given as pullback, but in
+that case we would run into the well-known strictness issue, that type
+substitution is functorial only up to isomorphism. This is not a critical issue,
+as there are standard solutions for recovering strict substitutions from weak
+ones \cite{kapulkin12simplicial,local-univ,clairambault2014biequivalence}. But
+if we ever need to look inside the definitions in the model, using displayed
+algebras yields less encoding overhead than strictifying pullbacks.

A term $\bs{t : \Tm\,\Gamma\,A}$ is a displayed flcwf section, which again
strictly preserves all structure. We use the same notation for the action of
```

This relates to my previous comment:

- In the remark, instead of “slices in the category of flcwfs”, I guess you mean to say “objects in slices of the category of flcwfs”. I do not think functoriality of substitution is even an issue if one does not use displayed algebras anyway. Then it is more natural to just directly use the finite limit category model, which does not require any strictification.

I you meant to write `objects of slices $\bs{\ms{flcwf}}/\bGamma$` instead of `as slices, as objects in $\bs{\ms{flcwf}}/\bGamma`. A slice of a category is the collection of maps into a single object, it does not refer to single maps.

Past comments (from first review, generally no action needed)

2.2.1

- The inductive definition of algebras would have been nicer if you lifted X : Set outside the induction. Of course, this does not extend to the generalizations in later chapters. The same comment applies to:
  - the definition of morphisms in Subsection 2.2.2 (lifting X_0 X_1 : Set with X^M : $X_0 \to X_1$),
  - the definition of displayed algebras in Subsection 2.2.3 (lifting X : Set with X^D : X → Set),
  - the definition of sections of displayed algebras in Subsection 2.2.4 (lifting X : Set with X^D : X → Set and X^S : (x : X) → X^D x).

3.1

- In Notation 3, the symbol ⊗ is a bit unfortunate. It suggest a general monoidal structure. The names p and q for the projections are interesting. They suggest you view this as a simplified case of context extension. You could comment on this.

3.2.2

- Regarding Notation 4 and 6: I am fine with the named notation, but it would be nice if it could be justified formally. Some non-conclusive thoughts on this.

  Like in your notation for projections of dependent (binary) products, the variable names in a context could be names for the generic terms available in that context. One could introduce the category of weakenings (generated by substitution lifts of p). Given an extended context Γ.A, we could introduce var_name for the family q[w] ∈ Ty(Δ, A') natural in a weakening w : Δ → Γ.A such that A' = A[p w]. By leaving the weakening implicit and coming up with rules for inferring the implicit weakening parameters, we could try to make Agda understand

this notation. Substitution lifting could be justified by introducing the slice construction on cwfs and generalizing pullback along $\sigma : \Delta \to \Gamma$ to an operation $\sigma\hat{}* : \mathrm{Slice}(\Gamma) \to \mathrm{Slice}(\Delta)$ defined on objects that are weakenings into Γ or when σ is a weakening.

Negative types

- It is not quite clear to me how your convention on naturality of type formers is supposed to be interpreted in Definition 15. Clearly, everything is natural in Γ. But what about the first context parameter? Probably you mean it to really come from the set of contexts as a discrete category rather than a category in the usual way. It would be good to clarify the phrase "everything is natural" in Notation 7 by mentioning Γ in some way.
- I like your discussion of the UIP axiom versus extensional identity types. However, if we did not have UIP in the metatheory, then all your coherence equations would have to be amended with higher coherence.

Universes

- The "type former" of Nat in U is really the specification of a type former in the family (Tm(-, U), El) induced by (U, El) with elimination targeting the family (Ty, Tm).

3.3.3

- As far as alternative presentations for 2LTT as used here are concerned, I would personally go for the second of the two alternatives outlined. It keeps the inner type formers and outer type formers as separate as possible and allows the specification of the inner fragment without talking about the outer fragment. So it is clearer how any given object theory extends to a 2LTT. But this is personal preference.
- I like the connection with template coding systems you point out. It would be nice if Agda had something like "template Agda", offering a 2LTT-like interface. I really want to be able to implement, say, coherence of the smash monoidal product, using a higher categorical argument in the outer layer and then be able to get a pure HoTT-term for coherence at any fixed level.

3.4.1

- Side-remark. There is a more fundamental model Cat_{disc,fib} of cwf that the presheaf model can be seen as being derived from. The contexts are categories and the types are displayed categories with unique backwards transports ("discrete fibrations"). The presheaf model over $\mathbb{C}$ is the fibrant slice of Cat_{disc,fib} over $\mathbb{C}$.
- Side-remark. It is possible to define types in the presheaf model so as to obtain Russel universes. But I think it is best to avoid strict equality of sets in theories.

3.4.3

- Martin Hofmann gave the reduced definition of exponentials with representables in the setting of a base category with finite products in one of his publications. I cannot find it at the moment, but it would be nice to cite. You could also say that this a special case of exponential with a tiny object and relate to that notion in category theory. For the general notion, you could mention that Tm_0 over Ty_0 is a "representable morphism" (in the sense of Grothendieck) or "locally representable", which is the property that enables this result, and refer to e.g. Awodey's presentation of cwfs ("natural models") and Paolo's thesis.

4.1

- Side-remark. The existing type-theoretic terminology for inductive types and their signatures is a bit unfortunate. We have a short word for the initial algebra before even talking about the signature and its category of algebras. In comparison, for algebraic theories, the terminology is conceptually clearer. I do not know what do to about this. This is a constant point of confusion for people trying to understand the type-theoretic terminology.

- The choice of theory of signatures (Definition 39) feels a bit ad hoc to me. I guess this is unavoidable. For example, you could have also added unit types or Σ-types without essentially changing the specifiable categories of algebras. And in implementations, one might want to allow user-defined record types etc. You definition is canonical in that you went for a minimal choice.
- Side-remark. Regarding Notation 13, maybe this is justification for keeping both versions of application around, the "categorical" one and the usual one. Personally, I would reserve app for the usual application app(f, a) ∈ Tm(Γ, B[a]) for f ∈ Tm(Γ, Π(A, B)) and a ∈ Tm(Γ, A). I do not know what the call the other one, though (except \overline{…} for transposition of an adjunction).

4.2.2

- Thank you for "bundling" things now. Personally, I would have preferred the bundled approach from the start. The unbundled style is unfortunately encouraged by our current proof assistants. (A related pet peeve is always case splitting and splitting on each hole as far as the proof assistant can manage, leading to proofs that are not abstract (β-reduced, ε-expanded, ….)

4.2.3

- I like to see Theorem 3 as the combination of two statements:
  - For every flcwf (with only weak K), the canonical fully faithful functor frm the global type category to the category is an equivalence. This can actually be regarded as the definition of weak K.
  - Flcwfs (with only weak K) are stable under slicing.
- With Theorem 3, Theorem 1 is equivalent to its categorical version: in a finitely complete category, an object Γ is initial exactly if the terminal object in the slice over Γ is (weakly) initial. If you wanted, you could add the latter condition(s).

4.2.6

- The two provided options are only two extremes. One could also imagine bundling up certain groups of components. Of course, none of this changes the end result.

4.3.1

- This regards the discussion of types in the groupoid model in [HS96] as functors Γ → Gpd and the displayed style. There is actually an "extended" groupoid model in which the types are displayed (cloven) fibrations, similar to the groupoid model. See Shulman's inverse diagram paper for this generalization. (Being a cloven fibration just means having lifts of morphisms given a lift of one point.) Unfortunately, the only universes in that model are for the cloven split fibrations, i.e. the functors Γ → Gpd.

4.5.1

- In Definition 55, it is interesting that you do not require naturality of the element of Con_M in M. I guess that is how you can get away with not defining morphisms of flcwfs in the bootstrapping. With naturality, you could show that bootstrapping signatures correspond to contexts in the theory of signatures, i.e. the initial ToS model.

  Without naturality, you are also forced to to follow the "bundling approach" to define the semantics. If you were to unbundle and define algebras, algebra morphisms for the theory of signatures, etc. separately, then naturality would be needed to show that the components "match up", e.g. to define source and target of an algebra morphism. Another benefit of the "bundle" that is perhaps worth mentioning.

5.1

- You now close U under a bunch of operations whereas is was not closed at all before for finitary quotient-inductive-inductive signatures. I guess one can also consider e.g. closure

under Ty_0-indexed products separately from closure under identity types. I guess the name "finitary" makes sense for a finite set of recursive parameters (which then forces U to not essentially not be closed under anything).

5.2.2

- Some bold-face letters around Theorem 7 are easy to mistake with the non-bold versions. This is unfortunate because they are completely unrelated. But with the amount of notation in the thesis, such problems are probably unavoidable.
- It is clear what naturality in the context is in Theorem 7. But since you give an explanation, it could be a bit more precise. You reindex both sides by σ (bold) applied to σ (non-bold).
- There is standard trick to get rid of some coherence isomorphisms when considering weak cwf morphisms, their composition, etc. One can phrase preservation of the empty context and context extension using preservation of universal properties. That is, if some context has the universal property of the empty context, then so does its image. Similarly for a candidate for context extension (consisting of an extended context, a projection morphism, and a term). Then composition of weak cwf morphisms is just given by composition. But that still leaves the mess with non-strict preservation of type formers (Σ, Id, K).
- In Chapter 4, it seems the category of flcwfs was itself the category of algebras of a signature. This seems to fail with weak morphisms. Or can you still recover it? Of course, you can describe the weak morphisms themselves by a signature (as you say do in Section 5.3).

Other stuff that comes to mind

- There was the example of an inductively defined operation M on types that has a constructor $\mathrm{M}\,(\mathrm{M}\,\mathrm{A}) \to \mathrm{M}\,\mathrm{A}$. You say this is not covered by your framework. Can you imagine an extension that does?
- The design choices for the type formers in the theory of signatures feel a bit ad hoc. For example, type formers such as equality types in U seem to allow for more definable substitutions, even if they do not necessarily increase the categories of algebras that can be encoded.
- Contexts in a cwf can be seen as a linearization of (the opposite of) finite direct categories. Finite direct categories can be generalized in two orthogonal directions:
- Allow for an external or internal type of objects at every degree. This is covered by types being closed under external and internal dependent function type.
- Allow for direct categories with transfinite height. This would mean infinite contexts, with variables indexed by an ordinal, for example ω. This can be reflected into types via ω^op-Reedy limits (record types with components indexed by ω).

One could imagine extending the theory of signatures in this way. The first one is already covered by infinitary signatures, but not the second one. For example, this would be needed to define semisimplicial types. Can your framework be extended in this direction?

- Categories of algebras of signatures fit a similar role as locally presentable categories in classical category theory. This relates to some discussion in Section 4.7 (around Gabriel-Ulmer duality). Assuming a classical setting, one should prove:
  - the category of algebras of a signature is a locally presentable category,
  - the functor of categories of algebras induced by a morphism of signatures is an accessible functor (and right adjoint, but you already proved that).

  In the reverse direction, one should examine which every locally presentable category can be realized as the category of algebras of a signature. Presumably, in the non-finitary case, one would have to generalize signatures to transfinite signatures as outlined in a comment above.

  I do not expect every accessible right adjoint between categories of algebras of two signatures to lift to a morphism of signatures. For that, the language of signatures seems too ad hoc. However, it may still be possible to find a class of signature morphisms that should be

regarded as invertible (because they induce equivalences of categories of algebras). Then one could hope that the bicategorical localization of the category of signatures at those "weak equivalences" is the (2, 1)-category of locally presentable categories.

In any case, one should start with the finitary case. That one can presumably be made constructive (there should be no choice issues).

- It would be interesting to look at a "dual" version of this framework that lets you build signatures to describe categories of "coalgebras". Coinductive types would then be terminal objects in those categories. I wonder whether there is a dual version of the term construction.