Type-Theoretic Signatures for Algebraic Theories and Inductive Types

András Kovács[1]

j.w.w. Thorsten Altenkirch, Paolo Capriotti, Ambrus Kaposi, Ambroise Lafont, Christian Sattler

[1]Eötvös Loránd University

21 May 2022, EuroProofNet WG6 meeting, Stockholm

Outline

Outline

Introduction

"Abstract" algebraic signatures:

- Finite product/limit categories, contextual cats, representable map cats.
- *Far from proof assistant implementations.*

Sketches:

- *Still far from implementations.*

"Syntactic" signatures:

- CIC signatures, GATs.
- *Formally tedious and poorly structured.*

Introduction

A **theory of signatures (ToS)** is a type theory where algebraic signatures can be defined.

The semantics of signatures is given by a model of a ToS.

Goals

1. Adequacy in implementation:
   - Exact computation of induction principles and β-rules.
   - Low encoding overheads.
   - Amenable to elaboration, perhaps also metaprogramming.
2. The theory of signatures is itself algebraic (perhaps even self-describing).
3. Semantics in categories of algebras.

Outline

Framework

We work in a type theory with **four universes**:

1. Set: universe of metatheoretic types (in the sense of 2LTT).
2. Sig: universe of signatures.
3. Sort: universe of "algebraic sorts".
4. $\mathbb{C}$: the category where semantic algebras live (internally).

Russell-style cumulative universes:

$$\mathsf{Sort} \subseteq \mathsf{Sig} \subseteq \mathsf{Set} \qquad \mathsf{Sort} : \mathsf{Sig} : \mathsf{Set}$$
$$\mathbb{C} \subseteq \mathsf{Set} \qquad \mathbb{C} : \mathsf{Set}$$

Restriction on elimination:

- From $\mathbb{C}$, only eliminate to $\mathbb{C}$.
- From Sig and Sort, only eliminate to Sig.

Framework - type formers

- $\mathsf{Sort} \subseteq \mathsf{Sig} \subseteq \mathsf{Set}$
- $\mathbb{C} \subseteq \mathsf{Set}$
- From $\mathbb{C}$, only eliminate to $\mathbb{C}$.
- From Sig and Sort, only eliminate to Sig.

General Assumptions

- Set is closed under ETT type formers.
- Sig is closed under $\top$ and Σ.

By varying type formers in Sig and Sort, we can describe numerous classes of inductive signatures.

We look at several of these in the following.

Closed inductive-inductive signatures

- Sort $\subseteq$ Sig $\subseteq$ Set
- $\mathbb{C} \subseteq$ Set
- From $\mathbb{C}$, only eliminate to $\mathbb{C}$.
- From Sig and Sort, only eliminate to Sig.

Close Sig under dependent functions with Sort domains:

$$\frac{A : \mathsf{Sort} \qquad B : A \to \mathsf{Sig}}{(a : A) \to B\,a : \mathsf{Sig}}$$

($+\ \lambda$, application)
Remark: $A \to \mathsf{Sig}$ above is a metatheoretic function type in Set

$$\begin{aligned}
&\mathsf{ConTySig} : \mathsf{Sig}\\
&\mathsf{ConTySig} := \quad (\mathsf{Con} : \mathsf{Sort}) \times (\mathsf{Ty} : \mathsf{Con} \to \mathsf{Sort})\\
&\qquad\qquad\qquad \times (- \triangleright - : (\Gamma : \mathsf{Con}) \to \mathsf{Ty}\,\Gamma \to \mathsf{Con}) \times \ldots
\end{aligned}$$

Open inductive-inductive signatures

Close Sig under dependent functions with $\mathbb{C}$ domains:

$$\frac{A : \mathbb{C} \qquad B : A \to \mathsf{Sig}}{(a : A) \to B\,a : \mathsf{Sig}}$$

$(+\ \lambda$, application)

$$\begin{aligned}&\mathsf{ListSig} : \mathbb{C} \to \mathsf{Sig}\\&\mathsf{ListSig}\,A\ :\equiv (\mathsf{List} : \mathsf{Sort}) \times (\mathsf{nil} : \mathsf{List}) \times (\mathsf{cons} : A \to \mathsf{List} \to \mathsf{List})\end{aligned}$$

Possible simple ListSig semantics:

A function sending each object A of a finite product category $\mathbb{C}$ to the category of A-list algebras that are internal to $\mathbb{C}$.

Finitary quotient inductive-inductive signatures

Close Sig under extensional equality:

$$\frac{A : \mathrm{Sig} \qquad x : A \qquad y : A}{x = y : \mathrm{Sig}}$$

(+ refl, equality reflection)

$\mathrm{QuotientSig} : (A : \mathbb{C}) \to (R : A \to A \to \mathbb{C}) \to \mathrm{Sig}$
$\mathrm{QuotientSig}\, A\, R := (A/R : \mathrm{Sort}) \times (|{-}| : A \to A/R) \times (\mathrm{quot} : R\, x\, y \to |x| = |y|)$

Infinitary quotient inductive-inductive signatures

Drop extensional equality from Sig, but add it to Sort instead.[1]

Also close Sort under dependent functions with $\mathbb{C}$ domains:

$$\frac{A : \mathbb{C} \qquad B : A \to \mathsf{Sort}}{(x : A) \to B : \mathsf{Sort}}$$

($+\ \lambda$, application)

$$\mathsf{WSig} : (A : \mathbb{C}) \to (B : A \to \mathbb{C}) \to \mathsf{Sig}$$
$$\mathsf{WSig}\, A\, B \ := (\mathsf{W} : \mathsf{Sort}) \times (\mathsf{sup} : (a : A) \to (B\, a \to \mathsf{W}) \to \mathsf{W})$$

At this point, we can specify every QII type from the HoTT book.

E.g. Cauchy reals, surreals, the cumulative hierarchy of sets.

[1] There's a semantic issue in mixing extensional Sig equality with infinitary branching.

Higher inductive-inductive signatures

We close Sig and Sort under intensional identity.

$$
\begin{aligned}
&\mathsf{TorusSig} : \mathsf{Sig} \\
&\mathsf{TorusSig} := \quad (\mathsf{T}^2 : \mathsf{Sort}) \times (\mathsf{b} : \mathsf{T}^2) \times (\mathsf{p} : \mathsf{b} = \mathsf{b}) \times (\mathsf{q} : \mathsf{b} = \mathsf{b}) \\
&\qquad\qquad\quad \times (\mathsf{t} : \mathsf{p} \cdot \mathsf{q} = \mathsf{q} \cdot \mathsf{p})
\end{aligned}
$$

Path composition $- \cdot -$ is definable from J.

Preliminary semantics

closed $A : \mathrm{Sig}$	$\Longrightarrow$	a finitely complete category of algebras
closed $f : A \to B$ with $A, B : \mathrm{Sig}$	$\Longrightarrow$	finitely continuous functor

We have a simple directed type theory.

We can do more than just write signatures:

- The *erasure map* NatSig $\to$ ListSig which forgets list elements is an ornament (see McBride, Dagand).
- Various model constructions of type theories can be defined as Sig functions. Most *syntactic models* can be rephrased in this way.
- Sig equivalences yield isomorphisms or equivalences of categories (depending on the exact semantics).

Outline

Setup & overview

The high-level syntax is a **2LTT whose inner level is a theory of signatures**.

We compile values in Sig and Sort to syntax in a formal ToS, using the "standard" presheaf model.

The ToS syntax is an initial structured cwf:

- Types as $\mathsf{Ty}\,\Gamma$, terms as $\mathsf{Tm}\,\Gamma\,A$.
- Tarski-style universe $\mathsf{Sort} : \mathsf{Ty}\,\Gamma$ with $\mathsf{El} : \mathsf{Tm}\,\Gamma\,\mathsf{Sort} \to \mathsf{Ty}\,\Gamma$.
- $A : \mathsf{Sig}$ is compiled to a type.
- $A : \mathsf{Sort}$ is compiled to a term with type Sort.
- Ty and Sort are closed under previous Sig and Sort type formers.

Setup & overview

The ToS syntax lives in **yet another 2LTT**, where $\mathbb{C}$ is the inner level.

We have Tarski-style $\mathbb{C} : \mathsf{Set}$ and $\mathsf{El}_{\mathbb{C}} : \mathbb{C} \to \mathsf{Set}$.

ToS type formers may refer to this $\mathbb{C}$, e.g.:

$$\Pi_{\mathbb{C}\,\mathsf{Ty}} : (A : \mathbb{C}) \to (\mathsf{El}_{\mathbb{C}}\,A \to \mathsf{Ty}\,\Gamma) \to \mathsf{Ty}\,\Gamma$$

$$\Pi_{\mathbb{C}\,\mathsf{Sort}} : (A : \mathbb{C}) \to (\mathsf{El}_{\mathbb{C}}\,A \to \mathsf{Tm}\,\Gamma\,\mathsf{Sort}) \to \mathsf{Tm}\,\Gamma\,\mathsf{Sort}$$

We consider three ToS-es and their semantics.

ToS	**semantics of types**
finitary QII	displayed cwf
infinitary QII	cwf isofibration
higher inductive-inductive	complete inner Reedy fibration[2]

[2] TYPES 2020, Capriotti & Sattler: *Higher categories of algebras for higher inductive definitions.*

Finitary QII semantics

Theory of signatures

- Ty is closed under Σ, $\top$, extensional $- = -$, $\mathbb{C}$-small products, Sort-small products
- Sort is closed under **no type formers**

Design choice: semantic contexts are *cwfs* + extra structure (not categories!)

The notion of **induction** can be directly defined in a cwf $\mathbb{C}$:

$$\mathsf{Inductive} : \mathsf{Obj}_{\mathbb{C}} \to \mathsf{Set}$$
$$\mathsf{Inductive}\,\Gamma := (A : \mathsf{Ty}_{\mathbb{C}}\,\Gamma) \to \mathsf{Tm}_{\mathbb{C}}\,\Gamma\,A$$

"An algebra Γ is inductive if every displayed algebra over it has a section."

Finitary QII semantics - finite limit cwfs

Definition

Finite limit cwf (flcwf): cwf + Σ + extensional identity + constant families ("democracy")

Clairambault & Dybjer: flcwfs are (bi)equivalent to finitely complete categories.

We model ToS contexts as flcwfs.

Theorem

In any flcwf, induction is equivalent to initiality.

Finitary QII semantics - summary

We assume that $\mathbb{C}$ is closed under $\top$, Σ and extensional identity.

(We can model $\mathbb{C}$ using any finitely complete category)

contexts:	flcwfs
types:	displayed flcwfs
substitutions:	strictly structure-preserving flcwf morphisms
terms:	strictly structure-preserving flcwf sections
Sort:	the flcwf of types in $\mathbb{C}$
El:	discrete displayed flcwf formation
$- = -$:	pointwise equality of strict flcwf sections
$\Pi_{\mathbb{C}\,\mathsf{Ty}}$	$\mathbb{C}$-small products
$\Pi_{\mathsf{Sort}\,\mathsf{Ty}}$	products with discrete index domains

Infinitary QII semantics

Theory of signatures

- Ty is closed under Σ, $\top$, $\mathbb{C}$-small products, Sort-small products.
- Sort is closed under Σ, $\top$, $\mathbb{C}$-small products, extensional $- = -$.

The previous semantics doesn't work!

The Sort type formers (e.g. $\top : \mathsf{Tm}\,\Gamma\,\mathsf{Sort}$) don't preserve limits strictly, only up to isos.

We switch to weak limit-preservation everywhere. This is technically more complicated.

Infinitary QII semantics - summary

We assume that $\mathbb{C}$ is closed under $\top$, Σ, extensional identity and Π.

(We can model $\mathbb{C}$ using any LCCC)

contexts:	flcwfs
types:	flcwfs isofibrations
substitutions:	weak cwf morphisms
terms:	weak cfw sections
Sort:	the flcwf of types in $\mathbb{C}$
El:	discrete flcwf isofibration formation
$- = -$:	pointwise equality of weak sections
$\Pi_{\mathbb{C}\,\mathsf{Ty}}$	$\mathbb{C}$-small indexed products
$\Pi_{\mathsf{Sort}\,\mathsf{Ty}}$	products with discrete index domains
$\Pi_{\mathbb{C}\,\mathsf{Sort}}$	internal $\mathbb{C}$-small products

HII semantics (Capriotti & Sattler)

Theory of signatures

- Ty is closed under Σ, $\top$, $\mathbb{C}$-small products, Sort-small products, intensional $- = -$.
- Sort is closed under Σ, $\top$, $\mathbb{C}$-small products, intensional $- = -$.

We assume that $\mathbb{C}$ models HoTT (we work in the "original" 2LTT).

contexts:	marked semisimplicial types
types:	complete inner Reedy fibrations
Sort:	universe of left fibrations

- This also yields a **structure identity principle** for HII theories.
- In an extra step we can add finite limits to categories of algebras.
- In yet another step we can show equivalence of induction and initiality.

Outline

Term algebras

We'd like *sufficient conditions* on $\mathbb{C}$ to have initial algebras for each signature.

In other words: construct initial algebras from simple "type formers".

Idea:

1. If $\mathbb{C}$ has an initial algebra for a ToS, we can use terms and types to build initial algs.
2. We construct the initial ToS model from simpler type formers.

Currently this works only for some ToS-es & semantics.

Term algebras for (in)finitary QII signatures

Assumptions

- $\mathbb{C}$ is a model of ETT.
- $\mathbb{C}$ has an initial ToS model.
- We fix a syntactic ToS context Ω (as a signature).

Each inductive sort in Ω is modeled as a set of terms.

For example, if $\Omega = \mathsf{NatSig}$:

$$\mathsf{Nat} := \mathsf{Tm}\,(\bullet \triangleright (N : \mathsf{Sort}) \triangleright (z : \mathsf{El}\,N) \triangleright (s : N \to \mathsf{El}\,N))\,(\mathsf{El}\,N)$$

Term algebras for (in)finitary QII signatures

1. An **internal algebra of** Ω in a ToS model is a morphism from the empty context to Ω.
2. By induction on ToS we show that any internal algebra yields an Ω-algebra in $\mathbb{C}$ (the term algebra).
3. In the slice model ToS$/\Omega$ the identity morphism from Ω to Ω gets us an internal algebra, hence also a term algebra.
4. By another induction on ToS, we can directly show that the term algebra is initial.

Theorem

If a model of ETT supports syntax for (in)finitary QII signatures, it supports all (in)finitary QII types.

Reductions to simple type formers

The remaining job is construct ToS syntaxes from simple type formers.

This is the **initiality construction** popularized by Voevodsky.

Results so far:

- ToS for **finitary inductive-inductive signatures** is constructible from just **W-types**.
- ToS for **closed QII signatures** was almost[3] constructed by Brunerie and De Boer in Agda from **propositional extensionality, inductive types and simple quotients by relations**.

Open problems:

- Fiore, Pitts, Steenkamp[4]: a class of infinitary QITs is constructible from the WISC axiom. Can we extend this to infinitary QIITs?
- The case for HIITs is open.

[3] The constructed theory is not exactly the same, but it can be plausibly adjusted to our use case.

[4] arXiv:2101.02994: *Quotients, inductive types, and quotient inductive types*