Free Algebraic Theories as Higher Inductive Types

Henning Basold, Niels van der Weide and Niccolò Veltri

LIACS — Leiden University
Radboud University Nijmegen
IT University of Copenhagen

TYPES 2019, Oslo, Norway

2019 June 10

Introduction

Interest

- Universal (higher) algebra in constructive type theory
  There are two more talks on this!
  - Andreas Lynge and Bas Spitters: *Universal Algebra in HoTT*
  - Stefano Piceghello: *Coherence for symmetric monoidal groupoids in HoTT/UF*
- Extension of type systems:
  - Well-behaved class of HITs
  - Coinductive types over HIT-generated functors

Interest

- Universal (higher) algebra in constructive type theory
  There are two more talks on this!
  - Andreas Lynge and Bas Spitters: *Universal Algebra in HoTT*
  - Stefano Piceghello: *Coherence for symmetric monoidal groupoids in HoTT/UF*
- Extension of type systems:
  - Well-behaved class of HITs
  - Coinductive types over HIT-generated functors

Extending Type Systems With HITs

- We use set-truncated HITs (QITs) to generate free algebras
  - Free monoids, groups, ...
  - Finite sets as free join-semilattices
  - Generally adjunctions between **Set** and $\mathrm{Alg}(T)$ for some algebraic theory T
- Instead we could introduce QITs as free algebras
- This covers most (all?) interesting examples of QITs
- How to go higher? Free $(\infty, 1)$-algebras!
- For a first step, go to Stefano's talk

More Coinductive Types

- QITs from algebraic theories give many functors: accessible functor
- Interesting for the theory of coalgebras
- Represent finite powerset $\mathcal{P}_f$ as HIT and introduce

$$\begin{aligned} \mathrm{Tr}_f &\colon \mathbf{Set} \to \mathbf{Set} \\ \mathrm{step} &\colon \mathrm{Tr}_f\, A \to \mathcal{P}_f(A \times \mathrm{Tr}_f\, A) \end{aligned}$$

$$\frac{c \colon X \to \mathcal{P}_f(A \times X)}{\mathsf{corec}\, c \colon X \to \mathrm{Tr}_\mathrm{f}\, A}$$

- Semantics of finitely branching labelled transition systems

Algebraic Theories and Their Algebras

Signatures and Terms

- Signature with symbols of arbitrary arity

$$\begin{aligned}&\textsf{record Signature} : \mathcal{U}_1 \textsf{ where}\\&\quad \textsf{sym} : \mathcal{U}_0\\&\quad \textsf{ar} \;\; : \textsf{sym} \longrightarrow \mathcal{U}_0\end{aligned}$$

- Given a signature Σ, we write $|\Sigma|$ for $\textsf{sym}\ \Sigma$
- Polynomial functor associated to signature

$$\begin{aligned}&[\![\_]\!] : (\Sigma : \textsf{Signature}) \longrightarrow \mathcal{U}_0 \longrightarrow \mathcal{U}_0\\&[\![\, \Sigma \,]\!]\ X = \bigsqcup\ |\Sigma|\ \lambda\ s \longrightarrow \textsf{ar}\ \Sigma\ s \longrightarrow X\end{aligned}$$

- Terms over a signature (free monad)

$$\begin{aligned}&\textsf{data Term}\ (\Sigma : \textsf{Signature})\ (V : \mathcal{U}_0) : \mathcal{U}_0 \textsf{ where}\\&\quad \textsf{leaf} \;\; : V \longrightarrow \textsf{Term}\ \Sigma\ V\\&\quad \textsf{node} : [\![\, \Sigma \,]\!]\ (\textsf{Term}\ \Sigma\ V) \longrightarrow \textsf{Term}\ \Sigma\ V\end{aligned}$$

Algebraic Theories

- Algebraic theories are given by a signature and equations

```
record AlgTheory : 𝒰₁ where
  sig  : Signature
  eqs : ∀ {X : 𝒰₀} → Rel (Term sig X)
```

Algebras for a Theory

- Pre-algebras do not have to respect the equations

```
record PreAlgebra : 𝒰₁ where
    carrier : 𝒰₀
    algebra : ⟦ Σ ⟧ carrier → carrier

  algebra* : Term Σ carrier → carrier
  algebra* = Term-rec (λ x → x) algebra
```

- Algebras also respect the equations

```
record Algebra : 𝒰₁ where
    pre-algebra : PreAlgebra
  open PreAlgebra pre-algebra public
    carrier-set  : is-set carrier
    resp-eq      : ∀ {t u : Term Σ carrier} →
                   eqs t u → algebra* t == algebra* u
```

Basics for Induction

- We denote a (pre-)algebra by

$$\mathcal{A} = (\mathrm{A}, \mathrm{a})$$

- Given a signature Σ, we lift $[\![\, \Sigma \,]\!]$ to predicates:

$$\langle\!\langle\, \Sigma \,\rangle\!\rangle : (A : \mathcal{U}_0)\,(P : A \longrightarrow \mathcal{U}_0) \longrightarrow [\![\, \Sigma \,]\!]\, A \longrightarrow \mathcal{U}_0$$

- We also lift terms to predicates:

$$\mathrm{TermP}\ \Sigma : \{X : \mathcal{U}_0\}\,(P : X \longrightarrow \mathcal{U}_0) \longrightarrow \mathrm{Term}\ \Sigma\, X \longrightarrow \mathcal{U}_0$$

Induction for Pre-Algebras

```
record PreInductive (𝒜 : PreAlgebra) : 𝒰₁ where
    predicate : A → 𝒰₀
    ind : ⟦ Σ ⟧ A → ⟪ Σ ⟫ A predicate x → predicate (a x)

  predicate* : Term Σ A → 𝒰₁
  predicate* = predicate ∘ algebra*

  ind* : ∀ {t : Term Σ A} → TermP Σ predicate t → predicate* t
  ind* = TermP-rec ...
```

Induction for Algebras

```
record InductiveProp (𝒜 : Algebra) : 𝒰₁ where
    pre-inductive : PreInductive (pre-algebra 𝒜)
  open PreInductive pre-inductive public
    ind-resp-eq : ∀{t u} (r : eqs t u)
      → (pt : TermP Σ predicate t) (pu : TermP Σ predicate u)
      → ind* pt == ind* pu [ predicate ↓ resp-eq r ]
```

The Free Algebra

Free algebra FreeAlgebra T for a theory T

- has exactly the constructors of the signature Σ
- fulfils exactly the equations of the theory
- is a set
- for each inductive property $P : (\text{FreeAlgebra } T) \longrightarrow \mathcal{U}_0$, has

$$\text{elim} : (x : \text{FreeAlgebra } T) \longrightarrow P\,x$$

- elim fulfils the expected reduction rules

Allows us to prove that FreeAlgebra T is initial algebra.

The Free Algebra

Free algebra FreeAlgebra T for a theory T

- has exactly the constructors of the signature Σ
- fulfils exactly the equations of the theory
- is a set
- for each inductive property $P : (\text{FreeAlgebra } T) \longrightarrow \mathcal{U}_0$, has
$$\text{elim} : (x : \text{FreeAlgebra } T) \longrightarrow P\,x$$
- elim fulfils the expected reduction rules

Allows us to prove that FreeAlgebra T is initial algebra.

Adding Generators

- To get an adjunction

$$\begin{tikzcd} \mathrm{Alg}(T) \arrow[rr, bend right, "U"'] \arrow[rr, phantom, "\perp"] & & \mathbf{Set} \arrow[ll, bend right, "L"'] \end{tikzcd}$$

  we need to embed generators
- Add inclusion of generators X in (pre-)algebras

  $$\mathsf{inj} : X \longrightarrow \mathsf{A}$$

- Add base case to (pre-)inductive properties

Ok Papa, this is all very cool ...

...but I knew that already!

So, What's Next?

Universal Algebra

- Note that our HITs work even for non-finitary theories
- We would like to generate quotient algebras as HITs
- This requires generating the free theory + quotient over the given algebra
- Hope: No need for axiom of choice
- Constructive Birkhoff variety theorem!
- NB: Niels and Herman showed that the adjunction can be constructed from quotients for finitary theories (MFPS'19)

Universal Algebra

- Note that our HITs work even for non-finitary theories
- We would like to generate quotient algebras as HITs
- This requires generating the free theory + quotient over the given algebra
- Hope: No need for axiom of choice
- Constructive Birkhoff variety theorem!
- NB: Niels and Herman showed that the adjunction can be constructed from quotients for finitary theories (MFPS'19)

Universal Coalgebra

- Many functors of interest are finitary or accessible
- Allow coinductive types over the associated HITs
- This gives theory of finitely-branching LTS etc.

Thank you!