(Co)ends and (Co)structure

HoTTEST Seminar – 01 December 2022

Jacob Neumann (jacob.neumann@nottingham.ac.uk)

Where to find these slides

jacobneu.github.io/research

0 Impredicative Encodings

Impredicative Encodings of Inductive Types in HoTT

In System F, we can obtain encodings of inductive types using the impredicative $\forall$ operator, e.g. $\mathbb{N}$ can be encoded as $\forall\alpha.(\alpha \to \alpha) \times \alpha \to \alpha$.
Awodey, Frey, and Speight (2018) studies how to do something similar in HoTT.

Example $\mathbb{N}$

- Unrefined encoding

$$\mathbb{N}^* :\equiv \prod_{C:\mathsf{Set}} (C \to C) \times C \to C$$

✓ Can define 0 and succ, prove (judgmental) β laws

✗ Can't rule out nonstandard elements: no η law

Analogous work in HoTT

- Unrefined encoding

$$\mathbb{N}^* :\equiv \prod_{C:\mathsf{Set}} (C \to C) \times C \to C$$

- Refined encoding

$$\mathbb{N}^+ :\equiv \sum_{\phi:\prod_{C:\mathsf{Set}}(C\to C)\times C\to C} \mathsf{isNat}\ \phi$$

where

$$\mathsf{isNat}\ \phi :\equiv \prod_{f:C\to D} (f\ c_0 = d_0) \to (f \circ \gamma = \delta \circ f) \to f(\phi_C(\gamma, c_0)) = \phi_D(\delta, d_0)$$

✓ Can define 0 and succ, prove (judgmental) β laws

✓ Can prove (propositional) η law, principle of induction

General Case

Defn If $T : \mathsf{Set} \to \mathsf{Set}$ is a functor, then define the category of T-**algebras** by

$$|T\text{-Alg}| :\equiv \sum_{C:\mathsf{Set}} T(C) \to C \qquad (C,\gamma) \to (D,\delta) :\equiv \sum_{f:C\to D} f\circ\gamma = \delta\circ T(f)$$

Thm The underlying set of the initial T-algebra is given by

$$\mu_T :\equiv \sum_{\phi:\prod_{C:\mathsf{Set}}(T(C)\to C)\to C} \mathsf{isNat}\ \phi$$

where

$$\mathsf{isNat}\ \phi :\equiv \prod_{(C,\gamma),(D,\delta):T\text{-Alg}} \ \prod_{f:(C,\gamma)\to(D,\delta)} f(\phi_C\ \gamma) = \phi_D\ \delta$$

Ingredients for an encoding:

- Polymorphic operation
- Naturality condition

1 A Structure Calculus

Category-Theoretic Gadget for operation+naturality: Ends

Defn Given a category $\mathbb{C}$ and a profunctor $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathrm{Set}$, the **end** of F is defined as

$$\int_{C:\mathbb{C}} F(C,C) :\equiv \sum_{(\phi : \prod_{C:\mathbb{C}} F(C,C))} \mathrm{isNat}\ \phi$$

where $\mathrm{isNat}(\phi)$: Prop is defined as

$$\mathrm{isNat}\ \phi :\equiv \prod_{C,D:\mathbb{C}} \prod_{f:\mathrm{Hom}_{\mathbb{C}}(C,D)} F(C,f)\ \phi_C = F(f,D)\ \phi_D$$

If F, G are both covariant functors (or both contravariant functors), then

$$\int_{C:\mathbb{C}} F(C) \to G(C)$$

is the type of natural transformations from F to G.

Lemma (Yoneda) For any $K : \mathbb{C}^{\mathrm{op}} \to \mathrm{Set}$ and $C_0 : \mathbb{C}$,

$$K(C_0) \simeq \int_{C:\mathbb{C}} \mathbf{y}\ C_0\ C \to K(C)$$

Idea: Use ends for impredicative encodings

Does it work to define

$$\mu_T \equiv \int_{C:\mathrm{Set}} (T(C) \to C) \to C?$$

- ✓ Polymorphic operation, β laws
- X Naturality condition: not right
  isNat ϕ: for all $f : C \to D$ and all $\theta : T(D) \to C$,
  $$f(\phi_C(\theta \circ T(f))) = \phi_D(f \circ \theta)$$

Structure Integrals

Defn For a profunctor $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathsf{Set}$, define the category F-Struct as

$$|F\text{-Struct}| :\equiv \sum_{C:\mathbb{C}} F(C,C)$$

$$(C,\gamma) \to (D,\delta) :\equiv \sum_{f:\mathrm{Hom}_{\mathbb{C}}(C,D)} F(C,f)\,\gamma = F(f,D)\,\delta$$

Note: If $F(C^-,C^+)$ is $T(C^-) \to C^+$, then F-Struct $\equiv$ T-Alg

Defn Given $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathsf{Set}$, define

$$\int_{C:\mathbb{C}} F(C,C)\,\mathbf{d}G(C,C) :\equiv \sum_{\phi:\prod_{(C,\gamma):F\text{-Struct}} G(C,C)} \mathrm{isNat}\ \phi$$

Structure Integral is the type of strong dinatural transforms

$$\phi : \prod_{(C,\gamma):F\text{-Struct}} G(C,C) \prod_{C:\mathbb{C}} F(C,C) \to G(C,C)$$

$$\text{isNat}\ \phi :\equiv \prod_{(C,\gamma),(D,\delta):F\text{-Struct}} \prod_{f:(C,\gamma)\to(D,\delta)} G(C,f)\ (\phi_C\ \gamma) = G(f,D)\ (\phi_D\ \delta)$$

$$\begin{tikzcd}
 & F(C,C) \arrow[rr, "\phi_C"] \arrow[dr, "F(C,f)"] & & G(C,C) \arrow[dr, "G(C,f)"] & \\
1 \arrow[ur, "\gamma"] \arrow[dr, "\delta"'] & & F(C,D) & & G(C,D) \\
 & F(D,D) \arrow[rr, "\phi_D"'] \arrow[ur, "F(f,D)"'] & & G(D,D) \arrow[ur, "G(f,D)"'] &
\end{tikzcd}$$

Structure Integrals and Initial Algebras

Thm For any functor $T : \mathrm{Set} \to \mathrm{Set}$, the set

$$\mu_T :\equiv \int_{C:\mathrm{Set}} T(C) \to C \,\mathbf{d}C$$

equipped with

$$\mathrm{in}_T :\equiv \lambda x.\big(\lambda(C,\gamma).\gamma(T(\phi_C\,\gamma)\;x),\;\ldots\big) \quad : T(\mu_T) \to \mu_T$$

is an initial T-algebra.

We also get the more general Yoneda-style lemma (due to Uustalu):

Lemma For any $K : \mathrm{Set} \to \mathrm{Set}$, and for any T with initial algebra (μ_T, in_T),

$$K(\mu_T) \simeq \int_{C:\mathrm{Set}} T(C) \to C \,\mathbf{d}K(C)$$

Curried encoding of $\mathbb{N}$

With this framework, we can also obtain the curried encoding of $\mathbb{N}$: if

$$\phi\colon \int_{C:\mathrm{Set}} (C \to C)\, \mathbf{d}(C \to C)$$

then this means

$$\phi : \prod_{C:\mathrm{Set}} (C \to C) \to (C \to C)$$

such that

$$\prod_{\gamma:C\to C} \quad \prod_{\delta:D\to D} \quad \prod_{f:C\to D} (f \circ \gamma = \delta \circ f) \to f \circ (\phi_C\, \gamma) = (\phi_D\, \delta) \circ f$$

Theorems for Free!

We can also use it to calculate free theorems. For instance, the type

$$\forall\alpha.(\alpha \to \alpha \to \mathsf{bool}) \to \mathsf{List}(\alpha) \to \mathsf{List}(\alpha)$$

If we take the structure integral

$$\int_{C:\mathsf{Set}} (C \to C \to \mathsf{bool})\, \mathbf{d}(\mathsf{List}\, C \to \mathsf{List}\, C)$$

then the naturality condition comes out as:

If $f : (C, \prec_C) \to (D, \prec_D)$ is monotone (in the sense that $(c \prec_C c') = (f\ c \prec_D f\ c')$ for all $c, c' : C$), then

$$(\mathsf{map}\ f) \circ (\phi_C\ \ \prec_C) = (\phi_D\ \ \prec_D) \circ (\mathsf{map}\ f)$$

2 A Costructure Calculus

Defn For $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathrm{Set}$, the **costructure integral** is defined as

$$\int^{C:\mathbb{C}} F(C,C) \,\mathbf{p}\, G(C,C) :\equiv \left(\sum_{(C,\gamma):F\text{-Struct}} G(C,C) \right) \Big/ \mathrm{Sim}_{F,G}$$

where $\mathrm{Sim}_{F,G}$ is the least equivalence relation such that

$$\prod_{(C,\gamma),(D,\delta):F\text{-Struct}} \; \prod_{f:(C,\gamma)\to(D,\delta)} \; \prod_{\psi:G(D,C)} \mathrm{Sim}_{F,G}\,(C,\gamma,G(f,C)\,\psi)\,(D,\delta,G(D,f)\,\psi)$$

Costructural Yoneda Lemma

Lemma For any T with terminal coalgebra (ν_T, out_T) and any $K : \mathrm{Set} \to \mathrm{Set}$,

$$K(\nu_T) \simeq \int^{C:\mathrm{Set}} C \to T(C) \,\mathbf{p}\, K(C)$$

This allows us to give an impredicative encoding of **coinductive** types, e.g. the type $\mathrm{Stream}(A)$ can be encoded as the costructure integral

$$\mathrm{Stream}(A) :\equiv \int^{C:\mathrm{Set}} C \to A \times C \,\mathbf{p}\, C$$

Cut for time:
Bisimulations and coinduction

Existential Types: Queues

Consider

$$\mathrm{Q}(X^-, X^+) :\equiv (A \times X^- \to X^+) \times (X^- \to \mathrm{Maybe}(A \times X^+)) \times X^+$$

Then a Q-Struct is an implementation of *queues*.

Then the costructure integral

$$\int^{C:\mathrm{Set}} \mathrm{Q}(C, C)\ \mathbf{p}1$$

"glues together" those implementations of queues which are bisimilar (representation independence).

3 Future Work

Avenues to explore

- Improvements to Impredicative Encodings
  - Higher Inductive Types?
  - Eliminate into arbitrary types (á la Shulman)
- Parametricity and Strong Dinaturality
- Develop the calculus more
- Semantics
- In Directed Type Theory

Thank you!