Type Theory based on Dependent Inductive and Coinductive Types

With an Eye Towards Cubical OTT

Henning Basold[1,2] Herman Geuvers[1,3]

1: Radboud University of Nijmegen, 2: CWI, Amsterdam and 3: TU Eindhoven

TYPES'16, Novi Sad
25 May 2016

Outline

1. The Calculus and Its Properties
2. The Good, The Bad and The Ugly
3. Towards a Fix

Familiar Examples I/II

Example (Vectors)

data $\mathrm{Vec}_A : \mathbb{N} \to$ **Set where**

$\mathsf{nil} \;:\; \top \to \mathrm{Vec}\ 0$

$\mathsf{cons} \;:\; (\mathsf{k} \;:\; \mathbb{N}) \to A \times \mathrm{Vec}\ k \to \mathrm{Vec}\ (k+1)$

In our type system

$$\mathrm{Vec}_A := \mu(X : (n : \mathbb{N}) \multimap * ; (\sigma_1, \sigma_2); (\mathbf{1}, A \times X \mathbin{@} k))$$

$\sigma_1 = (0) : \emptyset \to (n : \mathbb{N})$ and $\sigma_2 = (k+1) : (k : \mathbb{N}) \to (n : \mathbb{N})$

$X : (n : \mathbb{N}) \multimap * \mid \emptyset \vdash \mathbf{1} : *$

$X : (n : \mathbb{N}) \multimap * \mid k : \mathbb{N} \vdash A \times X \mathbin{@} k : *$

Familiar Examples II/II

- (Dependent) Function space is a coinductive type
- Existential quantification is its dual

Example (Propositional Equality)

data $\mathsf{Eq}_A : A \to A \to \mathbf{Set}$ **where**
  $\mathsf{refl} : (x : A) \to \top \to \mathsf{Eq}_A\ x\ x$

Represented by

$$\mathsf{Eq}_A(t, s) := \mu(X : (x : A, y : A) \rightarrowtriangle * ; \delta; \top)\ @\ t\ @\ s,$$

where $\delta : (x : A) \to (x : A, y : A)$ is the diagonal $\delta = (x, x)$.

- ...

Forming terms

Inspired by initial and final dialgebras (in fibrations):

	Introduction	Elimination
Inductive Types	Constructors	Recursion
Coinductive Types	Corecursion	Destructors

For example: Coinductive types

$$\frac{\vdash \nu(X : \Gamma \to\!\!\triangleright * ; \vec{\sigma}; \vec{A}) : \Gamma \to\!\!\triangleright * \qquad 1 \leq k \leq n}{\vdash \xi_k : (\Gamma_k, y : \nu \mathbin{@} \sigma_k) \to\!\!\triangleright A_k[\nu/X]} \textbf{(Coind-E)}$$

$$\frac{\vdash C : \Gamma \to\!\!\triangleright * \qquad \Delta, \Gamma_k, y_k : (C \mathbin{@} \sigma_k) \vdash g_k : A_k[C/X] \qquad \forall k = 1, \ldots, n}{\Delta \vdash \text{corec } \overrightarrow{(\Gamma_k, y_k).\, g_k} : (\Gamma, y : C \mathbin{@} \mathrm{id}_\Gamma) \to\!\!\triangleright \nu \mathbin{@} \mathrm{id}_\Gamma}$$

Computations

Reduction

- Closure of contraction relation that follows homomorphism diagrams
- Contraction for coinductive types:

$$\xi_k \mathbin{@} \tau \mathbin{@} (\mathrm{corec}\ \overrightarrow{(\Gamma_k, y_k).\, g_k} \mathbin{@} (\sigma_k \bullet \tau) \mathbin{@} u)$$
$$\succ \widehat{A_k} \left(\mathrm{corec}\ \overrightarrow{(\Gamma_k, y_k).\, g_k} \mathbin{@} \mathrm{id}_\Gamma \mathbin{@} x \right) [g_k/x][\tau, u]$$

- $\widehat{A_k}(t)$ – action of type A_k on term t (think functor)
- Example: Gives the usual β-reduction for functions
- Sanity check: Definitions are implemented in Agda

Theorem

The reduction relation preserves types.

Strong Normalisation

Theorem

The reduction relation is strongly normalising on well-typed terms.

Proof Outline

- Define saturated set model for types
- Set X of terms is saturated if
  - $X \subseteq \mathbf{SN}$
  - contains neutral terms
  - backwards closed under eliminators (destructors and recursion)
- Dependent types are interpreted families of saturated sets
- Show soundness: If $\Gamma \vdash t : A$, then $\forall \rho \in [\![\Gamma]\!].t \in [\![A]\!]_\rho$.

For details see

'Type Theory based on Dependent Inductive and Coinductive Types' – H.B. and Herman Geuvers, LICS 2016, arXiv: CS.LO/1605.02206.

The Good

- Inductive and coinductive types exactly dual
- No special type constructors for function space etc. necessary

The Bad

- No induction principle – though this can be easily added
- But what about coinduction then?

The Ugly

- Silent introduction of propositional equality
- Thus no good way of introducing coinduction

Blending OTT, Cubical TT and Dependent Inductive-Coinductive Types

Thank you for the inspiration Conor![1]

[1] McBride, that is. *A Cubical Crossroads* @ Agda Implementors' Meeting XXIII.

Plan

- Remove possibility to introduce propositional equality
- Paths between types through abstraction over points in interval (Cubical)
- Coercion operator (OTT)
- Heterogeneous path type (OTT)
- Coinduction: Constructor for paths

Removing Equality

Easy: Replace ...

$$\frac{k = 1, \ldots, n \qquad \sigma_k : \Gamma_k \to \Gamma \qquad \Theta, X : \Gamma \rightarrowtriangle * \mid \Gamma_k \vdash A_k : *}{\Theta \mid \emptyset \vdash \rho(X : \Gamma \rightarrowtriangle * ; \vec{\sigma}; \vec{A}) : \Gamma \rightarrowtriangle *}$$

... by

$$\frac{k = 1, \ldots, n \qquad \Theta, X : \Gamma \rightarrowtriangle * \mid \Gamma, \Gamma_k \vdash A_k : *}{\Theta \mid \emptyset \vdash \rho(X : \Gamma \rightarrowtriangle * ; \vec{A}) : \Gamma \rightarrowtriangle *}$$

Interval, Paths between Types and Coercion

$$\frac{}{\Gamma \vdash 0 : \mathbb{I}} \qquad \frac{}{\Gamma \vdash 1 : \mathbb{I}} \qquad \frac{i \in \Delta}{\Gamma \mid \Delta \vdash i : \mathbb{I}} \qquad \frac{\vdash p, q, r : \mathbb{I}}{\vdash p(q - r) : \mathbb{I}}$$

$$\frac{\Gamma \vdash S : * \qquad \Gamma \vdash T : *}{\Gamma \vdash S \sim T : *} \qquad \frac{\vdash Q : S \sim T \qquad \vdash p : \mathbb{I}}{\vdash Q\, p : *}$$

$$\frac{\Gamma \mid \Delta, i : \mathbb{I} \vdash M : * \qquad M[0/i] \equiv S \qquad M[1/i] \equiv T}{\Gamma \mid \Delta \vdash \langle i \rangle .\, M : S \sim T}$$

$$\frac{\vdash Q : S \sim T \qquad \vdash p, q : \mathbb{I} \qquad \vdash s : Q\, p}{\vdash s\, [p \mid Q \mid q\rangle : Q\, q}$$

Paths between terms

$$\frac{\vdash s : S \qquad \vdash t : T}{\vdash s \sim t : *} \qquad \frac{\vdash s, t : T \qquad \vdash Q : s \sim t \qquad \vdash p : \mathbb{I}}{\Gamma \vdash Q\, p : T}$$

$$\frac{\vdash t : T}{\vdash \mathsf{refl}_t : t \sim t}$$

Coinduction (for non-dependent types):

$$R : (x : \nu, y : \nu) \rightarrowtriangle *$$

$$\frac{x_k : \nu, y_k : \nu, z_k : R \mathbin{@} x_k \mathbin{@} y_k \vdash g_k : \overline{A_k}[R/X][\xi_k \mathbin{@} x_k / x, \xi_k \mathbin{@} y_k / y]}{\vdash \mathsf{coind}\ \overrightarrow{(x_k, y_k, z_k).\, g_k} : (x : \nu, y : \nu, z : R \mathbin{@} x \mathbin{@} y) \rightarrowtriangle x \sim y}$$

Reductions

$$0(p - q) \succ p$$

$$1(p - q) \succ q$$

$$(\langle i \rangle .\, M)\, p \succ M[p/i]$$

$$\text{If } Q : s \sim t, \text{then } Q\,0 \succ s$$

$$\text{If } Q : s \sim t, \text{then } Q\,1 \succ t$$

$$s\,[0 \mid Q \mid 0\rangle \succ s$$

$$s\,[1 \mid Q \mid 1\rangle \succ s$$

$$(\alpha_k \mathbin{@} t)[0 \mid \langle i \rangle .\, \mu(X : * ; \overrightarrow{A[i]}) \mid 1\rangle \succ \alpha_k \mathbin{@} (t[0 \mid \langle i \rangle .\, A_k[i] \mid 1\rangle)$$

$$\left(\text{corec } \overrightarrow{(y_k).\, g_k} \mathbin{@} t\right)[0 \mid \langle i \rangle .\, \nu(X : * ; \overrightarrow{A[i]}) \mid 1\rangle \succ (\text{corec } \overrightarrow{(y_k).\, g'_k} \mathbin{@} t')$$

$$\xi_k \mathbin{@} \left((\text{coind } \overrightarrow{(x_k, y_k, Q_k).\, g_k} \mathbin{@} u \mathbin{@} v \mathbin{@} Q)\, p\right)$$

$$\succ \left(\widehat{\overrightarrow{A_k}}(\text{coind } \overrightarrow{(x_k, y_k, Q_k).\, g_k} \mathbin{@} x \mathbin{@} y \mathbin{@} z)[g_k/z][\xi_k\, u/x, \xi_k\, v/y, Q/z]\right)\, p$$

Example (Function Extensionality)

Define

$$R : (h : A \to B, k : A \to B) \rightarrowtriangle *$$
$$R := (h).\,(k).\,\Pi x : A.h\,x \sim k\,x$$

and

$$\frac{\dfrac{x : A, h, k : A \to B, z : R\,@\,h\,@\,k \vdash z : \Pi x : A.h\,x \sim k\,x}{x : A, h, k : A \to B, z : R\,@\,h\,@\,k \vdash z\,x : h\,x \sim k\,x}}{f, g, p : \Pi x : A.f\,x \sim g\,x \vdash \mathsf{coind}\,((x, h, k, z).\,z\,x)\,@\,f\,@\,g\,@\,p : f \sim g}$$

Thank you very much for your attention!