A mode theory for a type theory of cubical and simplicial types

Ulrik Buchholtz
jww Jonathan Weinberger

TU Darmstadt/CAS Oslo

EUTypes, October, 2018

Outline

Introduction

Recall from Jonathan's talk that we're looking for a modal dependent type theory in which we can define and reason about universes of ∞-categories and reason about them in much the same way as we informally reason about 1-categories.

In particular, we would like to give $\hom$ the more precise type $\hom : (A :: \mathrm{Cat}) \to A^{\mathrm{op}} \to A \to \mathrm{Space}$. Therefore, we need the op-types.

(Compare in Riehl-Shulman: $\hom : (A : \mathsf{U}) \to A \to A \to \mathsf{U}$)

Introduction

Recall from Jonathan's talk that we're looking for a modal dependent type theory in which we can define and reason about universes of ∞-categories and reason about them in much the same way as we informally reason about 1-categories.

In particular, we would like to give $\hom$ the more precise type $\hom : (A :: \mathrm{Cat}) \to A^{\mathrm{op}} \to A \to \mathrm{Space}$. Therefore, we need the op-types.

(Compare in Riehl-Shulman: $\hom : (A : \mathsf{U}) \to A \to A \to \mathsf{U}$)

There is a now proliferation of modal calculi: for necessity (comonad), for possibility (monad), monads with strength. Other calculi also put restrictions on how derivations are formed, but in other ways, as in linear/affine/relevant logic, or bunched implications, or . . .

(Differential cohesive type theory involves six interacting modalities.)

Perhaps it would be interesting to have a general modal calculus for opposite types?!

Mode theories, 1

A mode theory (Licata-Shulman-Riley ’17) is a syntactic presentation of a simple kind of 2-(multi)category. (Actually, poset-enriched multi-category.) It consists of finitely many modes p (representing objects), typed function constants: $c : p_1, \ldots, p_n \to q$, equations between mode morphisms $\alpha \equiv \alpha' : \psi \to p$ and structural transformations of mode morphisms $\alpha \Rightarrow \alpha'$ (where $\psi \vdash \alpha, \alpha' : p$).

The equational axioms specify reversible object language structural properties (such as associativity, commutativity, unit laws).

The transformational axioms specify other object language structural properties (such as weakening and contraction).

All further syntactic structures will be indexed by mode morphisms up to $\equiv$.

Mode theory rules

Signatures: $\dfrac{}{\cdot\,\mathrm{sig}}$ $\quad \dfrac{\Sigma\,\mathrm{sig}}{(\Sigma, p\,\mathrm{mode})\,\mathrm{sig}}$ $\quad \dfrac{\Sigma\,\mathrm{sig} \quad (p_1\,\mathrm{mode}, \ldots, p_n\,\mathrm{mode}, q\,\mathrm{mode}) \in \Sigma}{(\Sigma, c : p_1, \ldots, p_n \to q)\,\mathrm{sig}}$

$$\frac{\Sigma\,\mathrm{sig} \quad \psi\,\mathrm{ctx}_\Sigma \quad p\,\mathrm{mode} \in \Sigma \quad \psi \vdash_\Sigma \alpha : p \quad \psi \vdash_\Sigma \alpha' : p}{(\Sigma, (\alpha \equiv \alpha' : \psi \to p))\,\mathrm{sig}}$$

(and same with $\Rightarrow$)

F and U Connectives

- F types are left adjoints (opfibration in fibrational framework): products, tensor, flat, op
- U types are right adjoints (opfibration in fibrational framework): implications, sharp, op

Rules

Propositions ($A\,\mathrm{type}_p$):

$$\frac{}{P\,\mathrm{type}_p} \qquad \frac{\psi \vdash \alpha : q \quad \Delta\,\mathrm{ctx}_\psi}{\mathsf{F}_\alpha(\Delta)\,\mathrm{type}_q} \qquad \frac{\psi, x : q \vdash \alpha : p \quad \Delta\,\mathrm{ctx}_\psi \quad A\,\mathrm{type}_p}{\mathsf{U}_{x.\alpha}(\Delta \mid A)\,\mathrm{type}_q}$$

Derivations ($\Gamma \vdash_\alpha A$ where $\Gamma :\ \mathrm{ctx}_\psi$, $A\,\mathrm{type}_q$, and $\psi \vdash \alpha : q$):

$$\frac{x : P \in \Gamma \quad \beta \Rightarrow x}{\Gamma \vdash_\beta P}\,\mathsf{v} \qquad \frac{\Gamma, \Gamma', \Delta \vdash_{\beta[\alpha/x]} C}{\Gamma, x : \mathsf{F}_\alpha(\Delta), \Gamma' \vdash_\beta C}\,\mathsf{FL} \qquad \frac{\beta \Rightarrow \alpha[\gamma] \quad \Gamma \vdash_\gamma \Delta}{\Gamma \vdash_\beta \mathsf{F}_\alpha(\Delta)}\,\mathsf{FR}$$

$$\frac{x : \mathsf{U}_{x.\alpha}(\Delta \mid A) \in \Gamma \quad \beta \Rightarrow \beta'[\alpha[\gamma]/z] \quad \Gamma \vdash_\gamma \Delta \quad \Gamma, z : A \vdash_{\beta'} C}{\Gamma \vdash_\beta C}\,\mathsf{UL} \qquad \frac{\Gamma, \Delta \vdash_{\alpha[\beta/x]} A}{\Gamma \vdash_\beta \mathsf{U}_{x.\alpha}(\Delta \mid A)}\,\mathsf{UR}$$

Contexts:

$$\frac{}{\cdot\,\mathrm{ctx}_.} \qquad \frac{\Gamma\,\mathrm{ctx}_\psi \quad A\,\mathrm{type}_p}{\Gamma, x : A\,\mathrm{ctx}_{\psi, x:p}}$$

Substitutions:

$$\frac{}{\Gamma \vdash_. \cdot} \qquad \frac{\Gamma \vdash_\gamma \Delta \quad \Gamma \vdash_\alpha A}{\Gamma \vdash_{\gamma, \alpha/x} \Delta, x : A}$$

Admissible Rules

$$\frac{\alpha \Rightarrow \beta \quad \Gamma \vdash_\beta A}{\Gamma \vdash_\alpha A} \qquad \frac{}{\Gamma, x : A \vdash_x A} \qquad \frac{\Gamma, x : A \vdash_\beta B \quad \Gamma \vdash_\alpha A}{\Gamma \vdash_{\beta[\alpha/x]} B}$$

$$\frac{\Gamma \vdash_\alpha C}{\Gamma, y : A \vdash_\alpha C} \qquad \frac{\Gamma, x : A, y : B \vdash_\alpha C}{\Gamma, y : B, x : A \vdash_\alpha C} \qquad \frac{\Gamma, x : A, y : A \vdash_\alpha C}{\Gamma, x : A \vdash_{\alpha[x/y]} C}$$

Example 1: Linear products and implication

Mode theory: m with $\otimes$ (associative, commutative).

Write $A \otimes B :\equiv \mathsf{F}_{x \otimes y}(x : A, y : B)$:

$$\frac{\Gamma, x : A, y : B, \Gamma' \vdash_{\beta[x \otimes y/z]} C}{\Gamma, z : A \otimes B, \Gamma' \vdash_\beta C} \otimes\mathsf{L} \qquad \frac{\beta = \alpha \otimes \alpha' \quad \Gamma \vdash_\alpha A \quad \Gamma \vdash_{\alpha'} B}{\Gamma \vdash_\beta A \otimes B} \otimes\mathsf{R}$$

Both ways of writing implication as a U-type gives the same type, the linear implication, $A \multimap B :\equiv \mathsf{U}_{x.x \otimes y}(y : A \mid B) \equiv \mathsf{U}_{x.y \otimes x}(y : A \mid B)$:

$$\frac{\Delta \vdash_\alpha A \quad \Gamma, x : B, \Gamma' \vdash_{\beta \otimes x \otimes \beta'} C}{\Gamma, z : A \multimap B, \Delta, \Gamma' \vdash_{\beta \otimes z \otimes \alpha \otimes \beta'} C} (A \multimap B)\mathsf{L} \qquad \frac{\Gamma, y : A \vdash_{\beta \otimes y} B}{\Gamma \vdash_\beta A \multimap B} (A \multimap B)\mathsf{R}$$

To get affine logic, add structural transformation $x \Rightarrow 1$. To get relevant logic instead, add structural transformation $x \Rightarrow x \otimes x$. To get ordinary cartesian logic, add both.

Example 2: Spatial logic

The spatial type theory for cohesion due to Shulman can be modeled with a mode theory with one mode c (cohesive) with a cartesian monoid and a context descriptor $x : c \vdash f(x) : c$ with $f(f(x)) = f(x)$ and $f(x) \Rightarrow x$. Then we define $\flat A :\equiv F_f A$ and $\sharp A :\equiv U_f(A)$. The fact that $\flat$ preserves products is enforced with the axiom $f(x \times y) = f(x) \times f(y)$.

$$\frac{A \in \Delta \quad \Delta \mid \Gamma, A \vdash C}{\Delta \mid \Gamma \vdash C} \qquad \frac{\Delta \mid \cdot \vdash A}{\Delta \mid \Gamma \vdash \flat A} \qquad \frac{\Delta, A \mid \Gamma \vdash C}{\Delta \mid \Gamma, \flat A \vdash C}$$

$$\frac{\Delta, \Gamma \mid \cdot \vdash C}{\Delta \mid \Gamma \vdash \sharp C} \qquad \frac{\sharp A \in \Delta \quad \Delta \mid \Gamma, A \vdash C}{\Delta \mid \Gamma \vdash C}$$

Outline

Cubical types with opposites

We capture the opposite categories using the framework, as this needs to be strict. (But do we need opposites over opposites?)

We have a mode theory with c (cohesive/cubical) and $x : c \vdash f(x) : c$ (flat) as well as $x : c \vdash o(x) : c$ (representing the opposite cubical type/category) with equations

$$f(f(x)) = f(x), \quad o(f(x)) = f(x), \quad o(o(x)) = x, \quad f(o(x)) = f(x).$$

Products and opposites

Further, we have $f(x \times y) = f(x) \times f(y)$ and $o(x \times y) = o(x) \times o(y)$, as both flat and opposite preserves products.

WLOG, any term $x_1, \ldots x_n \vdash t : c$ has the form

$$f(x_1) \times \cdots \times f(x_i) \times o(x_{i+1}) \times \cdots \times o(x_j) \times x_{j+1} \times \cdots \times x_n.$$

Thus, we may model the simply-typed calculus with three-zone contexts.

Dependent types with opposites

Contexts: $\Delta \mid \Gamma$ with rules

$$\frac{\Delta \mid \cdot \vdash A\,\mathrm{type}}{\Delta, x :: A \mid \Gamma\,\mathrm{ctx}} \qquad \frac{\Delta \mid \Gamma \vdash A\,\mathrm{type}}{\Delta \mid \Gamma, x : A\,\mathrm{ctx}} \qquad \frac{\Delta \mid \Gamma^{\mathrm{op}} \vdash A\,\mathrm{type}}{\Delta \mid \Gamma, x :^{\mathrm{op}} A\,\mathrm{ctx}}$$

Substitution/cut rules:

$$\frac{\Delta, x :: A \mid \Gamma \vdash \mathcal{J} \quad \Delta \mid \cdot \vdash a : A}{\Delta \mid \Gamma \vdash \mathcal{J}[a/x]} \qquad \frac{\Delta \mid \Gamma, x : A \vdash \mathcal{J} \quad \Delta \mid \Gamma \vdash a : A}{\Delta \mid \Gamma \vdash \mathcal{J}[a/x]}$$

$$\frac{\Delta \mid \Gamma, x :^{\mathrm{op}} A \vdash \mathcal{J} \quad \Delta \mid \Gamma^{\mathrm{op}} \vdash a : A}{\Delta \mid \Gamma \vdash \mathcal{J}[a/x]}$$

Opposite types:

$$\frac{\Delta \mid \Gamma \vdash A\,\mathrm{type}}{\Delta \mid \Gamma^{\mathrm{op}} \vdash A^{\mathrm{op}}\,\mathrm{type}} \qquad \frac{\Delta \mid \Gamma \vdash a : A}{\Delta \mid \Gamma^{\mathrm{op}} \vdash a^{\mathrm{op}} : A^{\mathrm{op}}}$$

$$\frac{\Delta \mid \Gamma, x :^{\mathrm{op}} A \vdash C\,\mathrm{type} \quad \Delta \mid \Gamma \vdash M : A^{\mathrm{op}} \quad \Delta \mid \Gamma^{\mathrm{op}}, u : A \vdash N : C^{\mathrm{op}}[u^{\mathrm{op}}/x]}{\Delta \mid \Gamma \vdash (\textsf{let } u^{\mathrm{op}} := M \textsf{ in } N) : C[M/x]}$$

$$(\textsf{let } u^{\mathrm{op}} := M^{\mathrm{op}} \textsf{ in } N) \equiv N[M/u] : C[M^{\mathrm{op}}/x] \qquad (A^{\mathrm{op}})^{\mathrm{op}} \equiv A$$

Perspectives

We don’t really know how useful this is, yet.
— If it is useful, we can start hacking on Agda-flat-op!

We also don’t really know how to axiomatize the interactions with universes.

What are some other instances of the setup?

Will our rules follow from the general framework of LSR ’19?

Perspectives

We don’t really know how useful this is, yet.
— If it is useful, we can start hacking on Agda-flat-op!

We also don’t really know how to axiomatize the interactions with universes.

What are some other instances of the setup?

Will our rules follow from the general framework of LSR ’19?

Thank you!