A Cubical Type Theory

Simon Huber
(j.w.w. Cyril Cohen, Thierry Coquand, Anders Mörtberg)

University of Gothenburg

HoTT/UF 2015
Warsaw, June 29, 2015

Cubical Type Theory: Overview

- Type theory where we can directly argue about *n*-dimensional cubes (points, lines, squares, cubes, ...).
- Based on a *constructive* model of type theory in cubical sets with connections and diagonals.
- Id, Π, Σ, data types, U
- The Univalence Axiom and function extensionality are provable.
- But: usual definitional equality for J only propositional! Problem in our previous approach recently pointed out by Dan Licata.

- Not having definitional equalities for J does not seem to be a problem (N.A. Danielsson)
- Other definitional equalities, e.g.,

$$\mathsf{ap} : (f : A \to B) \to \mathsf{Id}\, A\, a\, b \to \mathsf{Id}\, B\, (f\ a)\, (f\ b)$$
$$\mathsf{ap}\, f\, (\mathsf{refl}\, A\, a) = \mathsf{refl}\, B\, (f\ a)$$
$$\mathsf{ap}(g \circ f)\, p = \mathsf{ap}\, g\, (\mathsf{ap}\, f\ p)$$
$$\mathsf{ap}\, \mathrm{id}\, p = p$$

- Some higher inductive types with "good" definitional equalities

Implementation: Cubicaltt

Prototype proof-assistant implemented in Haskell.

Based on: "A simple type-theoretic language: Mini-TT",
T. Coquand, Y. Kinoshita, B. Nordström, M. Takeya (2008).

Mini-TT is a variant of Martin-Löf type theory with data types.
Cubicaltt extends Mini-TT with:

- name abstraction and application
- identity types
- composition
- equivalences can be transformed into equalities (glueing)
- some higher inductive types (experimental)

Try it: `https://github.com/mortberg/cubicaltt`

Basic Idea

Expressions may depend on *names* $i, j, k, \ldots$. E.g.,

$$x : A, i : \mathbb{I}, y : B(i, x) \vdash u(x, i) : C(x, i, y)$$

is a line connecting the two points

$$x : A, y : B(0, x) \vdash u(x, 0) : C(x, 0, y)$$
$$x : A, y : B(1, x) \vdash u(x, 1) : C(x, 1, y)$$

Each line $i : \mathbb{I} \vdash t(i) : A$ gives an equality

$$\vdash \langle i \rangle\, t(i) : \mathsf{Id}\, A\, t(0)\, t(1)$$

The Interval $\mathbb{I}$

- Given by $\varphi, \psi ::= 0 \mid 1 \mid i \mid 1 - i \mid \varphi \wedge \psi \mid \varphi \vee \psi$ (formulas)
- i ranges over *names* or *symbols*
- Intuition: i an element of $[0, 1]$, $\wedge$ is min, and $\vee$ is max.
- Equality is the equality in the free bounded distributive lattice with generators i, $1 - i$.
- De Morgan algebra via

$$
\begin{array}{ll}
1 - 0 = 1 & 1 - (\varphi \wedge \psi) = (1 - \varphi) \vee (1 - \psi) \\
1 - 1 = 0 & 1 - (\varphi \vee \psi) = (1 - \varphi) \wedge (1 - \psi) \\
 & 1 - (1 - i) = i
\end{array}
$$

NB: $i \wedge (1 - i) \neq 0$ and $i \vee (1 - i) \neq 1$!

Overview of the Syntax (w/o Universe)

$A, B, P, t, u, v ::=$	x	variables
	$\mid (x : A) \to B \mid \lambda x : A.\, t \mid t\; u$	Π-types
	$\mid (x : A) \times B \mid (t, u) \mid t.1 \mid t.2$	Σ-types
	$\mid \mathsf{ID}\; A\; B \mid \mathsf{IdP}\; P\; a\; b$	identity types
	$\mid \langle i \rangle t$	name abstraction
	$\mid t\; \varphi$	formula application
	$\mid \mathsf{comp}\; P\; u\; \vec{u}$	composition
	$\mid \mathsf{glue}\; A\; \vec{u} \mid (a, \vec{t}) \mid \mathsf{unGlue}\; A\; \vec{u}\; v$	glueing
	$\mid \ldots$	data types

Contexts and Substitutions

Contexts

$$\frac{}{() \vdash} \qquad \frac{\Gamma \vdash A}{\Gamma, x : A \vdash} \qquad \frac{\Gamma \vdash}{\Gamma, i : \mathbb{I} \vdash}$$

Substitutions are as usual but we also allow to assign a formula to a name:

$$\frac{\sigma : \Delta \to \Gamma \qquad \Delta \vdash \varphi : \mathbb{I}}{(\sigma, i = \varphi) : \Delta \to \Gamma, i : \mathbb{I}}$$

Face Operations

Certain substitutions correspond to face operations. E.g.:

$$(x = x, i = 0, y = y) \colon (x : A, y : B(i = 0)) \to (x : A, i : \mathbb{I}, y : B)$$

In general a face operation are $\alpha \colon \Gamma\alpha \to \Gamma$ setting some names to 0 or 1 and otherwise the identity.

Faces are determined by all the assignments $i = b$, $b \in \{0, 1\}$; write

$$\alpha = (i_1 b_1) \ldots (i_n b_n)$$

(Special case: $\alpha = \mathrm{id}$)

Basic Typing Rules

$$\frac{\Gamma \vdash}{\Gamma \vdash x : A} \quad (x : A \text{ in } \Gamma) \qquad \frac{\Gamma \vdash}{\Gamma \vdash i : \mathbb{I}} \quad (i : \mathbb{I} \text{ in } \Gamma)$$

$$\frac{\Gamma, x : A \vdash B}{\Gamma \vdash (x : A) \to B} \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A.\, t : (x : A) \to B}$$

$$\frac{\Gamma \vdash t : (x : A) \to B \qquad \Gamma \vdash u : A}{\Gamma \vdash t\, u : B(x = u)}$$

Also: Sigma types and data types ...

Equality between types

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash \mathsf{ID}\, A\, B} \qquad \frac{\Gamma, i : \mathbb{I} \vdash A}{\Gamma \vdash \langle i \rangle A : \mathsf{ID}\, A(i0)\, A(i1)}$$

$$\frac{\Gamma \vdash P : \mathsf{ID}\, A\, B \qquad \Gamma \vdash \varphi : \mathbb{I}}{\Gamma \vdash P\varphi}$$

$$(\langle i \rangle A)\varphi = A(i\varphi)$$
$$\langle i \rangle\, Pi = P$$

$$\frac{\Gamma \vdash P : \mathsf{ID}\, A\, B}{\Gamma \vdash P0 = A} \qquad \frac{\Gamma \vdash P : \mathsf{ID}\, A\, B}{\Gamma \vdash P1 = B}$$

Heterogeneous Identity Types

$$\frac{\Gamma \vdash P : \mathsf{ID}\, A\, B \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B}{\Gamma \vdash \mathsf{IdP}\, P\, a\, b}$$

$$\frac{\Gamma, i : \mathbb{I} \vdash t : A}{\Gamma \vdash \langle i \rangle t : \mathsf{IdP}(\langle i \rangle A)\ \ t(i0)\ \ t(i1)}$$

$$\frac{\Gamma \vdash e : \mathsf{IdP}\, P\, a\, b \qquad \Gamma \vdash \varphi : \mathbb{I}}{\Gamma \vdash e\varphi : P\varphi}$$

$$(\langle i \rangle t)\varphi = t(i\varphi)$$
$$\langle i \rangle\, ei = e$$

$$\frac{\Gamma \vdash e : \mathsf{IdP}\, P\, a\, b}{\Gamma \vdash e0 = a : P0}$$

$$\frac{\Gamma \vdash e : \mathsf{IdP}\, P\, a\, b}{\Gamma \vdash e1 = b : P1}$$

Identity Types

We set

$$\mathsf{Id}\, A\, a\, b := \mathsf{IdP}\, (\langle i \rangle A)\, a\, b$$

This is enough to justify reflexivity, symmetry, function extensionality, and that singletons are contractible!

In the implementation:

- universe U with U : U
- ID $A\, B$ is Id U $A\, B$

Demo!

Kan Operations

Given $i : \mathbb{I} \vdash A$ we want an equivalence between $A(i0)$ and $A(i1)$.

Require additional composition operations.

Refinement of Kan's extension condition (1955)

"Any open box can be filled"

Systems

A system

$$\vec{u} = [\alpha \mapsto u_\alpha]$$

for $\Gamma \vdash A$ is given by a family of compatible terms

$$\Gamma\alpha \vdash u_\alpha : A\alpha$$

(α ranging over a set of faces L, L downwards closed)

Systems

For a system $\vec{u}$

$$\Gamma\alpha \vdash u_\alpha : A\alpha \quad (\alpha \in L)$$

and substitution $\sigma\colon \Delta \to \Gamma$ we get a system

$$\Delta\beta \vdash (\vec{u}\sigma)_\beta : A\sigma\beta \quad (\beta \in L\sigma)$$

Satisfying: $(\vec{u}\alpha)_{\mathrm{id}} = u_\alpha$ for $\alpha \in L$

Composition

$$\frac{\Gamma \vdash P : \mathsf{ID}\, A\, B \qquad \Gamma \vdash a : A \qquad \Gamma\alpha \vdash p_\alpha : \mathsf{IdP}\, P\alpha\ a\alpha\ u_\alpha\ (\alpha \in L)}{\Gamma \vdash \mathsf{comp}\, P\, a\, \vec{p} : B}$$

$$\begin{aligned} (\mathsf{comp}\, P\, a\, \vec{p})\sigma &= \mathsf{comp}\, P\sigma\ a\sigma\ \vec{p}\sigma \\ \mathsf{comp}\, P\, a\, \vec{p} &= p_{\mathrm{id}}\, 1 \qquad \text{if } \mathrm{id} \in L \\ \text{So: } (\mathsf{comp}\, P\, a\, \vec{p})\alpha &= p_\alpha 1 \qquad \text{if } \alpha \in L \end{aligned}$$

Kan Filling

$$\frac{\Gamma \vdash P : \mathsf{ID}\, A\, B \qquad \Gamma \vdash a : A \qquad \Gamma\alpha \vdash p_\alpha : \mathsf{IdP}\; P\alpha\;\; a\alpha\;\; u_\alpha \;\; (\alpha \in L)}{\Gamma \vdash \mathsf{fill}\, P\, a\, \vec{p} : \mathsf{IdP}\; P\, a\, (\mathsf{comp}\, P\, a\, \vec{p})}$$

Can be reduced to composition using connections:

$$\mathsf{fill}\, P\, a\, \vec{p} = \langle i \rangle\;\, \mathsf{comp}\, (\langle j \rangle\; P(i \wedge j))\;\, a\;\, [\alpha \mapsto \langle j \rangle\; p_\alpha(i \wedge j), (i0) \mapsto \langle j \rangle\; a]$$

Special case: path lifting property ($\vec{p} = [\,]$)

Demo!

Composition

$\mathsf{comp}\,(\langle i\rangle A)\,a\,\vec{p}$ is defined by induction on the type A:

- Case $i : \mathbb{I} \vdash A = \mathsf{Id}\,B\,b_0\,b_1$.

$$\begin{aligned}&\mathsf{comp}\,(\langle i\rangle A)\,a\,\vec{p} = \\ &\qquad \langle i\rangle\ \mathsf{comp}\,(\langle i\rangle B)\,(a i)\,[\alpha \mapsto p_\alpha i, (i0) \mapsto b_0, (i1) \mapsto b_1]\end{aligned}$$

- Case $i : \mathbb{I} \vdash A = (x : B) \to C$. For $b_1 : B(i1)$

$$\mathsf{comp}\,(\langle i\rangle A)\,f\,\vec{g}\,b_1 = \mathsf{comp}\,(\langle i\rangle C(x = b))\,(f\ b_0)\,(\vec{g}\ b)$$

  with $b = \mathsf{fill}^-\,(\langle i\rangle B)\,b_1\,[\,]$ and $b_0 = b0 : B(i0)$.

Glue

Given a system of equivalences on a type we introduce a new type:

$$\frac{\Gamma \vdash A \qquad \Gamma\alpha \vdash f_\alpha : \mathsf{Equiv}\ T_\alpha\ A\alpha \quad (\alpha \in L)}{\Gamma \vdash \mathsf{glue}\ A\ \vec{f}}$$

$$\frac{\Gamma \vdash a : A \qquad \Gamma\alpha \vdash t_\alpha : T_\alpha \qquad \Gamma\alpha \vdash f_\alpha t_\alpha = a\alpha : A\alpha}{\Gamma \vdash (a, \vec{t}) : \mathsf{glue}\ A\ \vec{f}}$$

$$(\mathsf{glue}\ A\ \vec{f})\sigma = \mathsf{glue}\ A\sigma\ \vec{f}\sigma \qquad (a, \vec{t})\sigma = (a\sigma, \vec{t}\sigma)$$

$$\mathsf{glue}\ A\ \vec{f} = T_{\mathrm{id}} \qquad (a, \vec{t}) = t_{\mathrm{id}} \qquad \text{if } \mathrm{id} \in L$$

Composition in a Universe

We also can define composition for glue $A\,\vec{f}$.

If we have a universe U, we can reduce composition in U to glue.

Any path $P : \mathrm{Id}\,\mathrm{U}\,A\,B$ induces an equivalence $P^+ : \mathrm{Equiv}\,A\,B$ whose function part is given by:

$$a : A \vdash \mathrm{comp}\,P\,a\,[\,] : B$$

Univalence Axiom

Using glue we can also prove the Univalence Axiom!

Demo!

Further Work

- Formal correctness proof of model and implementation
- Proof of canonicity for the type system
- Definitional equality for J?
- Related work: Brunerie/Licata, Polonsky, Altenkirch/Kaposi, Bernardy/Coquand/Moulin

Thank you!