A syntax for cubical type theory

Ambrus Kaposi
(joint work with Thorsten Altenkirch)

University of Nottingham

Agda Ideas’ Meeting XIX
May 23, 2014
Introduction

Problem

- Goal: a type theory with the property:
  if two objects are indistinguishable by observation, they are equal
- A candidate: homotopy type theory
  - Equality is defined by an inductive type with the J eliminator
  - Addition of the univalence axiom ("isomorphic types are equal")
  - We don’t know how to run programs involving this axiom
Homotopy type theory teaches us that equality can be described individually for each type former, eg.:

- **pairs:** $((a, b) =_{A \times B} (a', b')) \simeq (a =_A a' \times b =_B b')$
- **functions:** $(f =_{A \to B} g) \simeq (\prod(x : A). f x =_B g x)$
- **natural numbers:**
  - $(\text{zero} =_{\mathbb{N}} \text{zero}) \simeq 1$
  - $(\text{zero} =_{\mathbb{N}} \text{suc} m) \simeq 0$
  - $(\text{suc} m =_{\mathbb{N}} \text{zero}) \simeq 0$
  - $(\text{suc} m =_{\mathbb{N}} \text{suc} n) \simeq (m =_{\mathbb{N}} n)$

Let's define equality separately for each type former, as above!
Inspiration and structure of talk

This work is based on the following papers:

- Bernardy, Moulin: A computational interpretation of parametricity, 2012
- Bezem, Coquand, Huber: A cubical set model of type theory, 2013

Table of contents:

Introduction

Internal parametricity

Connection

Kan Cubical sets
Internal parametricity

Basic setup

- Type theory with explicit substitutions, without the identity type
- Judgement types:

\[\Gamma \vdash a : A\]
\[\Delta \vdash a[\rho] : A[\rho]\]

- Applying substitutions:

\[\Gamma \vdash A : U\]
\[\rho : \Delta \Rightarrow \Gamma\]
\[\Delta \vdash t : A[\rho]\]

\[(\rho, x \mapsto t) : \Delta \Rightarrow \Gamma, x : A\]
Heterogeneous equality (i)

For elements of a $\Sigma$-type, the second equality depends on the first:

\[
((a, b) =_{\Sigma(x:A).B x (a', b')}) \simeq (\Sigma(r : a =_{A} a').\text{transport}_B r b =_{B} a' b')
\]

To model this, we will a heterogeneous equality: a binary logical relation.
Heterogeneous equality (ii)

- The heterogeneous equality relation:

\[
\frac{\Gamma \vdash A : U}{\Gamma^= \vdash \sim_A : A[0\Gamma] \to A[1\Gamma] \to U}
\]

- \(\Gamma^=\) is the context containing two copies of the context \(\Gamma\) and proofs that they are related.

\[
\emptyset^= \equiv \emptyset
\]

\[(\Gamma, x : A)^= \equiv \Gamma^=, x_0 : A[0\Gamma], x_1 : A[1\Gamma], \bar{x} : x_0 \sim_A x_1\]

- 0, 1 project out the corresponding components.

\[
i_{\emptyset} \equiv () : \emptyset \Rightarrow \emptyset
\]

\[
i_{\Gamma,A} \equiv (i_{\Gamma}, x \mapsto x_i) : (\Gamma, x : A)^= \Rightarrow \Gamma, x : A
\]
\(\sim\) on different type formers

Given \(\Gamma \vdash A, \Gamma, x : A \vdash B\), we previously had:

\[
\begin{align*}
\Gamma & \vdash (a, b) : \Sigma(x : A).B \\
\Gamma & \vdash (a', b') : \Sigma(x : A).B \\
\Gamma & \vdash ((a, b) =_{\Sigma(x : A).B} (a', b')) \sim (\Sigma(r : a =_A a').\text{transport}_{\lambda x. B[x]} r b =_B a'[a] b')
\end{align*}
\]

Now we have:

\[
\begin{align*}
\Gamma & \vdash A \vdash B : U \\
\Gamma & \vdash (a, b) : (\Sigma A B)[0] \\
\Gamma & \vdash (a', b') : (\Sigma A B)[1] \\
\Gamma & \vdash (a, b) \sim_{\Sigma(x : A).B} (a', b') \equiv \Sigma(r : a \sim_A a'). \\
& \quad b \sim_B [x_0 \mapsto a, x_1 \mapsto a', \bar{x} \mapsto r] b' : U
\end{align*}
\]

For \(\Pi\) types:

\[
\begin{align*}
\Gamma & \vdash A \vdash B : U \\
\Gamma & \vdash f_0 : (\Pi A B)[0] \\
\Gamma & \vdash f_1 : (\Pi A B)[1] \\
\Gamma & \vdash f_0 \sim_{\Pi A B} f_1 \equiv \Pi(x_0 : A[0], x_1 : A[1], \bar{x} : x_0 \sim_A x_1).f_0 x_0 \sim_B f_1 x_1 : U
\end{align*}
\]

For the universe (we will replace this later):

\[
A \sim_U B \equiv A \to B \to U
\]
Every term is a congruence

We validate the rule

\[
\frac{\Gamma \vdash u : A}{\Gamma = \vdash u^\sim : u[0\Gamma] \sim_A u[1\Gamma]}
\]

for each term former. This corresponds to showing that every term is parametric, eg.:

\[
\frac{\Gamma \vdash x : A \vdash b : B}{\Gamma = \vdash (\lambda x. b)^\sim \equiv \lambda x_0, x_1, \bar{x}. b^\sim}
\]

\[
\frac{\Gamma \vdash f : \Pi A B \quad \Gamma \vdash u : A}{\Gamma = \vdash (f \ u)^\sim \equiv f^\sim u[0\Gamma] u[1\Gamma] u^\sim}
\]

For types, we choose:

\[
\frac{\Gamma \vdash A : U}{\Gamma = \vdash A^\sim \equiv \sim_A}
\]
Homogeneous equality

To internalise the logical relation, i.e. to have an equality in the same context, we define the substitution $R$ and the term $\text{refl}$ mutually:

\[
\begin{align*}
\Gamma & \vdash R_{\Gamma} : \Gamma \Rightarrow \Gamma = \\
\Gamma & \vdash \text{refl} \equiv (a \sim) [R_{\Gamma}] : a \sim_A [R_{\Gamma}] a
\end{align*}
\]

\[
\begin{align*}
\emptyset & \vdash R_{\emptyset} \equiv () : \emptyset \\
\Gamma.x : A & \vdash R_{\Gamma.A} \equiv (R_{\Gamma}, x, x, \text{refl} x) : (\Gamma.A) =
\end{align*}
\]

We introduce the abbreviation:

\[
a =_A b \equiv a \sim_A [R] b
\]

We also need an $S_{\Gamma} : (\Gamma =) = \Rightarrow (\Gamma =) =$, with a family of similar operations to $\text{refl}$. 
The functor $\equiv$ (i)

We can extend $\equiv$ to act not only on contexts, but also terms, and substitutions:

- $\Gamma \mapsto \Gamma^\equiv$
- $\Gamma \vdash t : A \mapsto \Gamma^\equiv \vdash t^\equiv \equiv (t[0], t[1], t\sim) : A^\equiv$
- $(\rho, x \mapsto t) \mapsto (\rho^\equiv, t^\equiv)$
Higher dimensions (i)

By iterating $\equiv$, we get higher dimensional cubes:

\[
\Gamma \vdash x : A
\]

\[
\Gamma^= \vdash (x : A)^1 \equiv x_0 : A[0\Gamma].x_1 : A[1\Gamma].\bar{x} : x_0 \sim_A x_1
\]

\[
\Gamma^2 \vdash (x : A)^2 \equiv
\begin{align*}
x_{00} & : A[0\Gamma]0\Gamma= \\
x_{01} & : A[0\Gamma1\Gamma=] \\
x_{10} & : A[1\Gamma0\Gamma=. A[0]\Gamma] \\
x_{11} & : A[1\Gamma1\Gamma=. A[0]\Gamma]
\end{align*}
\]

\[
\bar{x}_0 : x_{00} \sim_A x_{10}[0\Gamma=. A[0]\Gamma].A[1]\Gamma].(\bar{x})_1 : (x_0 \sim_A x_1)[1\Gamma=. A[0]\Gamma].A[1]\Gamma].\bar{x} : (\bar{x})_0 \sim x_0 \sim_A x_1 (\bar{x})_1
\]

\[
\Gamma^3 \vdash (x : A)^3 \equiv
\begin{align*}
x_{000} & : A[000] \\
x_{001} & : A[001] \\
x_{010} & : A[010] \\
x_{011} & : A[011] \\
x_{100} & : A[100] \\
x_{101} & : A[101] \\
x_{110} & : A[110] \\
x_{111} & : A[111]
\end{align*}
\]

\[
\begin{align*}
\bar{x}_0 & : x_{000} \sim_A x_{110}[0\Gamma=. A[0]\Gamma].x_{001} \sim_A x_{111}[1\Gamma=] x_{011} \\
\bar{x}_0 & : (x_0 \sim_A x_1)[0\Gamma=. A[0]\Gamma].x_{010} \sim_A x_{110}[1\Gamma=. A[0]\Gamma].x_{000} \sim x_0 \sim_A x_1 (\bar{x})_1
\end{align*}
\]
Higher dimensions (ii)

Figure: Cubes of dimension 0-3.
The functor $\equiv$ (ii)

The iterated version of $\equiv$ makes any context into a presheaf over the base category of cubical sets.

So, a context $\Gamma$ is a presheaf $\mathcal{C} \to \text{Con}$ where

- $\mathcal{C}$ is the category of names and substitutions for the cubical set model,
- $\text{Con}$ is the category of contexts and substitutions in the term model.
Definition of $\sim_U$

Our previous definition:

$$A \sim_U B \equiv A \to B \to U$$

We replace this by:

$$\Gamma \vdash A \sim_U B \equiv \Sigma_{\sim\sim} : A \to B \to U$$

- $\text{coe}^0 : A \to B$
- $\text{coh}^0 : \Pi(x : A). x \sim \text{coe}^0 x$
- $\text{uni}^0 : \Pi(x : A, p p' : \Sigma(y : B). x \sim y). p = p'$
- $\text{coe}^1 : B \to A$
- $\text{coh}^1 : \Pi(y : B). \text{coe}^1 y \sim y$
- $\text{uni}^1 : \Pi(y : B, p p' : \Sigma(x : A). x \sim y). p = p'$
Kan conditions (i)

We are required to provide coe and coh now for each type former. For \( \Sigma \) we can define it as

\[
\Gamma \vdash \text{coe}^i_{\Sigma(x:A).B} \equiv \lambda (a, b). (\text{coe}^i_A a, \text{coe}^i_B [\bar{x} \mapsto \text{coh}^1_i a] b) \\
: (\Sigma(x:A).B)[i] \to (\Sigma(x:A).B)[1-i]
\]

\[
\Gamma \vdash \text{coh}^i_{\Sigma(x:A).B} \equiv \lambda (a, b). (\text{coh}^i_A a, \text{coh}^i_B [\bar{x} \mapsto \text{coh}^1_i a] b) \\
: \Pi (w : (\Sigma(x:A).B)[i]). w \sim^i_{\Sigma(x:A).B} \text{coe}^1_i w
\]
Kan conditions (ii)

coe and coh can be seen as first level Kan operations: given a point, they extend it to a line.

A higher level Kan operation completes an incomplete square, 3-dimensional cube, etc. Eg.:

\[
\begin{array}{cccc}
 & x_{01} & & x_{11} \\
\downarrow & & \uparrow & \\
\bar{x}_0 & & \bar{x}_1 & \\
\downarrow & (\bar{x})_0 & & \downarrow & \\
x_{00} & & x_{10} &
\end{array}
\]

To define the first level Kan operations for Π, we need the second level Kan operations. However,

\[
\Gamma \vdash .x_0 : A[0].x_1 : A[1] \vdash x_0 \sim_A x_1 : U,
\]

so

\[
(\Gamma \vdash .x_0 : A[0].x_1 : A[1]) \vdash (x_0 \sim_A x_1) \sim : (x_{00} \sim_A [0] x_{10}) \sim_U (x_{01} \sim_A [1] x_{11}).
\]
Kan for $\Pi$

Coerce for $\Pi$:

$$\Gamma = \vdash \text{coe}^0_{\Pi(x:A)}.B \equiv \lambda f. \lambda x. \text{coe}^0_B[\bar{x} \mapsto \text{coh}^1_A x] (f (\text{coe}^1_A x))$$

$$\quad : (\Pi(x : A[0]).B[0, x]) \to (\Pi(x : A[1]).B[1, x])$$

The type of the coherence operation:

$$\Gamma = \vdash \text{coh}^0_{\Pi(x:A)}.B : \Pi (f : (\Pi(x : A).B)[0]. f \sim_{\Pi(x:A)}.B (\text{coe}^0_{\Pi(x:A)}.B f))$$

We get coherence by using higher level Kan:
Identity type (i)

Non-dependent eliminator:

\[
\Gamma \vdash P : A \to U \\
\Gamma \vdash r : x \sim_A [R \Gamma] y \\
\Gamma \vdash u : P x \\
\Gamma \vdash \text{transport}_P r u : P y
\]

We have that \( P \) is a congruence:

\[
\Gamma \vdash P : A \to U \\
\Gamma \vdash P^\sim[R \Gamma] : \Pi(x_0, x_1 : A, \bar{x} : x_0 \sim_A [R \Gamma] x_1).P x_0 \sim_U P x_1
\]

And we define transport by using \( P^\sim[R] \):

\[
\Gamma \vdash P : A \to U \\
\Gamma \vdash r : x =_A y \\
\Gamma \vdash u : P x \\
\Gamma \vdash \text{transport}_P r u \equiv (P^\sim[R \Gamma] x y r).\text{coe}^0 u : P y
\]
Identity type (ii)

The computation rule of transport says that $\text{transport}_P(\text{refl}\, x) \equiv \text{id}$. We have

$$
\text{transport}_P(\text{refl}\, x) \\
\equiv (P \sim [R\] x x x \sim [R\] ).\text{coe}^0 \\
\equiv (P x) \sim [R\].\text{coe}^0 \\
\equiv \text{id}
$$

The last step is justified by adding the following rule:

$$\Gamma \vdash A : U
\frac{}{\Gamma \vdash A \sim [R\] \equiv (\sim_A [R\] \sim, \text{id}, \text{refl}, \text{id}, \text{refl}) : A \sim U A}$$
Identity type (iii)

We also show that singletons are contractible i.e. we show how to construct the terms $s$ and $t$ of the following type:

$$
\begin{align*}
\Gamma & \vdash a, b : A \\
\Gamma & \vdash r : a =_A b \\
\Gamma & \vdash (s, t) : (a, \text{refl } a) = \Sigma(x : A).a =_A x (b, r) \\
& \equiv \Sigma(s : a \sim_A [R\Gamma] b).\text{refl } a \sim_{a \sim_A [R\Gamma] x [R\Gamma, a, b, s]} r
\end{align*}
$$

$s$ is constructed by filling the following incomplete square from bottom to top:

\begin{center}
\begin{tikzpicture}
    \node (a) at (0,0) {$a$};
    \node (b) at (2,0) {$b$};
    \node (c) at (0,-2) {$a$};
    \node (d) at (2,-2) {$a$};
    \draw[->] (a) -- node[above] {$s$} (b);
    \draw[->] (c) -- node[above] {refl $a$} (d);
    \draw[->] (a) -- node[right] {refl $a$} (c);
    \draw[->] (b) -- node[right] {$r$} (d);
\end{tikzpicture}
\end{center}
Conclusion

- A different presentation of Bernardy and Moulin’s work on internal parametricity: equality defined as a logical relation.
- Using equivalence for elements of the universe instead of any relation.
- This forces us to define the first level Kan operations for type formers.
- Higher Kan operations can be computed from the first level Kan operations.
- Unfinished work:
  - Relation to the uniformity condition in the Bezem, Coquand, Huber cubical set model
  - Prove decidability, canonicity
  - Examples of higher inductive types
  - Implement it in Agda
  - Create a proof assistant based on this theory :)

Ambrus Kaposi  (University of Nottingham)  A syntax for cubical type theory  May 23, 2014  22 / 22