Toward a New Formulation of Extensional Type Theory

Andrew Polonsky

VU Amsterdam

We report progress on the design of a type system with extensional equality that would admit decidable type checking by avoiding the propositional reflection rule.

Our approach is similar to Observational Type Theory of Altenkirch et al, in that extensional equality is generated from a logical relation defined by induction on type structure. Equality on Π-types is defined to be pointwise equality of functions.

Our contribution is to reflect the logical relation by a dedicated type constructor (which inhabitants represent *type equalities*) in such a way that extensionality witnesses exist for every term *in the same type universe*. This allows us to define a higher-dimensional substitution operation:

$$\frac{\Gamma, x : A \vdash t : T \qquad \Gamma \vdash a : A}{\Gamma \vdash t[a/x] : T[a/x]} \qquad\qquad \frac{\Gamma, x : A \vdash t : T \qquad \Gamma \vdash a^* : a \simeq_A a'}{\Gamma \vdash t[a^*//x] : t[a/x] \sim_{T[a^*//x]} t[a'/x]}$$

The fact that T and $t[a] \sim_{T[a^*]} t[a']$ belong to the same type hierarchy allows the operation to be iterated to arbitrary dimensions.

Every type has the structure of a globular set with reflexivities, which furthermore satisfies a certain kind of Kan filling condition. The condition is realized by new operators which formally witness the homotopy lifting property.

For simplicity, we present our system as an extension of $\lambda *$, with the understanding that the stratified version is conjectured to be strongly normalizing.

$$\begin{aligned} A, t, e, \gamma \quad ::= \quad & * \mid x \mid \Pi x{:}A.B \mid \Sigma x{:}A.B \mid A \simeq B \mid a \sim_e b \mid a \simeq_A a' \\ & \mid \lambda x{:}A.t \mid st \mid (s,t) \mid \pi_1 t \mid \pi_2 t \\ & \mid *^* \mid \Pi^*[x, x', x^*]{:}A^*.B^* \mid \Sigma^*[x, x', x^*]{:}A^*.B^* \mid \simeq^* A^* B^* \\ & \mid \mathsf{r}(t) \mid e(t) \mid \bar{e}(t) \mid t_e \mid t^e \mid \gamma \looparrowdown_{e^*} \gamma' \end{aligned}$$

This system has three kinds of equality relations.

The constructor $A \simeq B$ represents the type of equalities between types:

$$\frac{A : * \qquad B : *}{A \simeq B : *}$$

Any term $e : A \simeq B$ of this type induces a binary relation between the corresponding types:

$$\frac{e : A \simeq B \qquad a : A \qquad b : B}{a \sim_e b : *}$$

The term former $\sim_e : A \to B \to *$ can thus be seen as the eliminator for the type $A \simeq B$. The constructors for this type are the symbols $*^*, \Pi^*, \Sigma^*, \simeq^*$, which witness the fact that equality is a congruence with respect to all type constructors (including $\simeq$ itself).

The type $A \simeq B$ enjoys the following computation rules:

$$\begin{aligned} A \sim_{*^*} B \quad & \longrightarrow \quad A \simeq B \\ f \sim_{\Pi^*[x,x',x^*]:A^*B^*} f' \quad & \longrightarrow \quad \Pi a{:}A \Pi a'{:}A' \Pi a^* : a \sim_{A^*} a'.\ fx \sim_{B^*[a/x, a'/x', a^*/x^*]} f'x' \\ p \sim_{\Sigma^*[x,x',x^*]:A^*B^*} p' \quad & \longrightarrow \quad \Sigma a^* : \pi_1 p \sim_{A^*} \pi_1 p'.\ \pi_2 p \sim_{B^*[\pi_1 p/x, \pi_1 p'/x', a^*/x^*]} \pi_2 p' \\ e \sim_{\simeq^* A^* B^*} e' \quad & \longrightarrow \quad \Pi a{:}A \Pi a'{:}A' \Pi a^* : a \sim_{A^*} a' \\ & \qquad\;\; \Pi b{:}B \Pi b'{:}B'\, \Pi b^* : b \sim_{B^*} b'.\ (a \sim_e b) \simeq (a' \sim_{e'} b') \end{aligned}$$

The type $a \simeq_A a'$ is the extensional equality type on A:

$$\frac{A : * \qquad a : A \qquad a' : A}{a \simeq_A a' : *} \qquad\qquad \frac{a : A}{\mathsf{r}(a) : a \simeq_A a}$$

The following identities are valid:

$$\begin{aligned} t[a^*//x] &= \mathsf{r}(t) && \text{if } x \notin \mathsf{FV}(t) \\ a \simeq_A a' &= a \sim_{\mathsf{r}(A)} a' \\ A \simeq B &= A \simeq_* B \end{aligned}$$

In particular, $(A \simeq B) = (A \simeq_* B) = (A \sim_{\mathsf{r}(*)} B) = (A \sim_{*[a^*//x]} B)$, justifying the typing of the higher-dimensional substitution.

For example, one can define in this system the "mapOnPaths" operation

$$\frac{\Gamma \vdash t : \Pi x{:}A.B \qquad \Gamma \vdash \alpha : a_1 \simeq_A a_2}{\Gamma \vdash t.\alpha : ta_1 \sim_{B[\alpha//x]} ta_2}$$

by taking $t.\alpha \coloneqq \mathsf{r}(t)a_1a_2\alpha$. It computes as

$$(\lambda x{:}A.b).\alpha \quad\longrightarrow\quad b[\alpha//x]$$

So far, the equality type gives each type the structure of a globular set with reflexivities. For higher-dimensional compositions, this structure must also satisfy a certain filling condition. We obtain this by adding operations which allow one to transfer terms from one side of type equality to another.

$$\frac{e : A \simeq B \qquad a : A}{e(a) : B} \qquad\qquad \frac{e : A \simeq B \qquad b : B}{\overline{e}(b) : A}$$

$$a_e : a \sim_e e(a) \qquad\qquad b^e : \overline{e}(b) \sim_e b$$

These are operations with reduction rules. As they can be used in any context, they have the effect of witnessing the homotopy lifting property:

$$\frac{\Gamma, x : A \vdash B(x) : * \qquad \Gamma \vdash a^* : a \simeq_A a' \qquad \Gamma \vdash b : B(a)}{\Gamma \vdash b_{B(a^*)} : b \sim_{B(a^*)} B(a^*)(b)}$$

where $B(a^*) = B[a^*//x] : B(a) \simeq B(a')$.

Composition and symmetry are defined as follows. For $\alpha : a \simeq_A a'$, we have

$$\begin{aligned} \mathring{\alpha}(x) &: (x \simeq_A a) \simeq (x \simeq_A a') & \qquad \overline{\alpha} &: a' \simeq_A a \\ \mathring{\alpha}(x) &\coloneqq (x \simeq_A y)[\alpha//y] & \qquad \overline{\alpha} &\coloneqq \overline{\mathring{\alpha}(a')}(\mathsf{r}(a')) \end{aligned}$$

In particular, for $a^* : a_0 \simeq_A a$, we have $\mathring{\alpha}(a_0)(a^*) : a_0 \simeq_A a'$.

Higher-dimensional fillers can be constructed following this pattern.

The $\looparrowleft_{e^*}$ operation is an "exchange law" used when higher cells are substituted over higher cells, as, for example, in the clause for path substitution over reflexivity:

$$\mathsf{r}(B)[a^*//x] \quad = \quad \looparrowleft_{\mathsf{r}(B[a^*//x])}$$