Равенство в Теории Типов

Шабалин Александр

2015 г.

Propositions as Types

$$\begin{array}{rcl} \text{Proposition} & \leftrightarrow & \text{Type} \\ \text{Proof} & \leftrightarrow & \text{Program} \\ \text{Proof simplification} & \leftrightarrow & \text{Computation} \end{array}$$

- Syntactic $=$. $\dfrac{A \qquad B}{A \wedge B}$
- $=$ on propositions.
- $=$ as proposition.

Intentional Type Theory

Martin-Löf 1975

Definitional equality $=_{\text{def}}$

Definition, αβ-equivalence, $\dfrac{a =_{\text{def}} b}{x[a] =_{\text{def}} x[b]}$, $\dfrac{a : A \qquad A =_{\text{def}} B}{a : B}$.

Propositional equality (a.k.a. Identity) $I(A, a, b)$

$$r(a) : I(A, a, a)$$

$$\frac{c[x] : C[x, x, r(x)]}{f(x, x, r(x)) =_{\text{def}} c[x]}$$

$$\begin{aligned} J : (&(x : A) \to C(x, x, r(x))) \\ &\to (xy : A)(p : I(A, x, y)) \\ &\to C(x, y, p) \end{aligned}$$

Extentional Type Theory

Martin-Löf 1983

Definitional equality

Definition, Equivalence, Substituiting equals for equals, α.

Judgemental equality $A = B,\ a = b : A$

Defined for every type, $\beta\eta$,

$$\xi\ \frac{\begin{array}{c}[x : A] \\ f(x) = g(x) : B(x)\end{array}}{\lambda x.\ f(x) = \lambda x.\ g(x) : \Pi(x : A)B(x)}\ ,$$

$$\frac{a \in A \qquad A = B}{a \in B}\ ,\ \frac{a = b \in A \qquad A = B}{a = b \in B}\ .$$

Propositional equality

$$\mathrm{ER}\ \frac{c : I(A, a, b)}{a = b : A}\ ,\ \mathrm{UIP}\ \frac{c : I(A, a, b)}{c = r : I(A, a, b)}\ .$$

Observational Type Theory

Definitional equality: αβη.

Propositional equality is heterogeneous equality:

$(x : A) = (y : B)$, $\quad$ $\mathit{refl} : (x : A) \to (x : A) = (x : A)$. Coercion:

$$\frac{p : S_0 = S_1 \qquad s_0 : S_0}{s_0[p\rangle : S_1}$$

Coherence:

$$\frac{p : S_0 = S_1 \qquad s_0 : S_0}{s_0[|p|\rangle : (s_0 : S_0) = (s_0[p\rangle : S_1)}$$

Coercion, coherence rules are defined for each and every type.

Result: propositional equality is truly extensional.

Unified Type Theory

ITT + impredicative Prop. Equality is Leibniz:
$\forall P : A \to Prop.\ P(a) \implies P(b).$

Computational Type Theory

Based on ETT. Define the meaning of $x = y : A$ for the types you want. $x : A$ is then a shortcut for $x = x : A$. And propositional equality is truly identified with judgemental equality.
Set types: $\{x : A | p(x)\}$ Quotient types: $(x, y) : A//E$. Change an equivalence relation to E.

Homotopy Type Theory

Definitionally like ITT but with η.
Proof relevance for equality. Think βreduction chain. $a = b : A$ when there is a path from a to b. As such no UIP (and K). OTOH have function extensionality.

ZOMBIE

Forget automatic β. Get equality relations from the context and use them.