On the axiom of extensionality

May 17, 2010

Introduction

The goal of this note is to extend Gandy's interpretability result [2] of extensional type theory in intensional type theory for simple type theory, to a dependent type system with one universe. For simple type theory, Gandy observed that any definable term is extensional. As formulated in [2], this reflects the fact that "the mathematician believes that the complex quantities which he builds up from the primitives by the usual logical operations will also be extensional". The goal of this note is to analyse this result in the context of dependent type theory, showing how to extend type theory with an extensional equality on each type. An important methodological remark about this approach is the belief that

equality should not be thought of as inductively defined, with reflexivity as the introduction rule, but should be instead defined by recursion on the type

For dependent types, equality at an universe is defined to be isomorphism. (A crucial point is to precise what isomorphic means; the equality will be extensional equality.) So, in any statement replace a type by an isomorphic one. V. Voevodsky [6] emphasizes the difference between type theory and set theory in this respect. In set theory, we cannot replace a set with two elements $X = \{a, b\}$ by another one $Y = \{0, 1\}$ in an arbitrary statement. For instance, in the statement $a \in X$, we cannot replace X by Y. However, X and Y are isomorphic. Here would be a corresponding statement for type theory, with no mention of equality. We consider a type system with a type of natural numbers N, a type of boolean N_2, unit type N_1 and sum type $A + B$. Then, if we have a definable functional $F : U_1 \to U_0$ we have $F\ (N_1 + N) \leftrightarrow F\ N$ and $F\ N_2 \leftrightarrow F\ (N_1 + N_1)$ and $F\ (N + N) \leftrightarrow F\ N$. Notice that, on the other hand it is possible to define F such that $F(N_1)$ is true and $F(N_2)$ false, by taking for instance

$$F = \lambda X.\forall f : X \to N_2.\forall x\ y : X.\ f\ x = f\ y$$

The goal of this note is to present a method that should produce a proof of this statement, and furthermore, an algorithm that will compute from F an isomorphism between $F\ (N_1 + N_1)$ and $F\ N_k$. We present it first for simple type lambda calculus. We explain then how this can be extended to dependent types. Beside giving this algorithm for computing isomorphism, this method gives a way to interpret extensional equality, subset and quotient of (small) types.

1 Gandy's extensionality proof

We consider simple type theory with the following types

$$A\ ::=\ o \mid A \to A$$

We interpret Gandy's extensional proof as a model construction. Dependent types are used in a crucial way for building this model. For each simple type A, we define a type $[A]$ and an equivalence relation $eq\ A$ on this type $[A]$. We define $[o]$ to be the type o (thought of as one universe in some dependent type theory) and $eq\ o$ to be equivalence. For higher types we take

$$[A \to B] = (\Sigma f : [A] \to [B])\ \forall x_1\ x_2 : [A].\ eq\ A\ x_1\ x_2 \to eq\ B\ (f\ x_1)\ (f\ x_2)$$

and

$$eq\ (A \to B)\ f\ g = \forall x : [A].\ eq\ B\ (f.1\ x)\ (g.1\ x)$$

Notice that the interpretation of function types is non standard: an element f of type $[A \to B]$ is a pair, the first component $f.1$ of which is a function of type $[A] \to [B]$ and the second component $f.2$ is a proof that this function sends equal elements to equal images.

If Γ is a context $x_1 : A_1, \ldots, x_n : A_n$ we define $\sigma : [\Gamma]$ to mean that σ is an environment such that $\sigma(x_i) : [A_i]$ for $i = 1, \ldots, n$. We define

$$eq\ (\Gamma, x : A)\ (\sigma_1, x = a_1)\ (\sigma_2, x = a_2) = eq\ \Gamma\ \sigma_1\ \sigma_2 \times eq\ A\ a_1\ a_2$$

One can then interpret Gandy's extensionality proof in the following way. For each term $\Gamma \vdash t : A$ we define $t\sigma$ of type $[A]$ if σ is of type $[\Gamma]$ and we define a proof $tq : eq\ A\ (t\sigma_1)\ (t\sigma_2)$ whenever $q : eq\ \Gamma\ \sigma_1\ \sigma_2$.

For defining these objects, we need first to define $p; q : eq\ A\ a_1\ a_3$ for $p : eq\ A\ a_1\ a_2$ and $q : eq\ A\ a_2\ a_3$. This corresponds to the proof that $eq\ A$ is a transitive relation. This is direct by induction on A, given that the relation $eq\ o$, which is logical equivalence, is transitive. We define also for each element $a : [A]$ an object $1 : eq\ A\ a\ a$ and for each environment $\sigma : [\Gamma]$ an object $1 : eq\ \Gamma\ \sigma\ \sigma$. This corresponds to the proof that $eq\ A$ is a reflexive relation. (Interestingly, the proof of symmetry is not needed for defining $t\sigma$ and tq.)

We get then the following computation rules

$$\begin{array}{lcl} x\sigma & = & \sigma(x) \\ (\lambda x.t)\sigma.1\ a & = & t(\sigma, x = a) \\ (\lambda x.t)\sigma.2\ a_1\ a_2\ p & = & t(1, p) \\ (t\ u)\sigma & = & t\sigma.1\ (u\sigma) \end{array}$$

and, if q is a proof of $eq\ \Gamma\ \sigma_1\ \sigma_2$

$$\begin{array}{lcl} xq & = & q(x) \\ (\lambda x.t)q\ a & = & t(q, 1) \\ (t\ u)q & = & (t\sigma_1.2\ (u\sigma_1)\ (u\sigma_2)\ (uq)); (tq\ (u\sigma_2)) \end{array}$$

There is some choice in the last rule, since we could take instead

$$(t\ u)q = (tq\ (u\sigma_1)); (t\sigma_2.2\ (u\sigma_1)\ (u\sigma_2)\ (uq))$$

and this choice is not understood yet: an interpretation should not be ambiguous. In any case, both choices give an explanation of extensional equality.

2 A symbolic view of this proof

We forget the distinction between A and $\langle A \rangle$ (which is a priori dangerous to do, but this works) and we define

$$\mathsf{Eq}_o\ X\ Y = (X \to Y) \times (Y \times X)$$

and

$$\mathsf{Eq}_{A\to B}\ f\ g = \forall a : A.\ \mathsf{Eq}_B\ (f\ a)\ (g\ a)$$

What is now quite weird is that, a priori, this kind of recursion on the types is too simple, but we can still follow the structure Gandy's proof, and produce sensible computations. We define also $\langle t\rangle l$ of type $\mathsf{Eq}_A\ (t\sigma_1)\ (t\sigma_2)$ if l is of type $\mathsf{Eq}_\Gamma\ \sigma_1\ \sigma_2$ and $c|p$ of type $\mathsf{Eq}_B\ (c\ a_1)\ (c\ a_2)$ if $c : A \to B$ and $p : \mathsf{Eq}_A\ a_1\ a_2$. The computation rules are

$$\begin{array}{lcl} x\sigma & = & \sigma(x) \\ (\lambda x.t)\sigma\ a & = & t(\sigma, x = a) \\ (\lambda x.t)\sigma|p & = & t(1, p) \\ (t\ u)\sigma & = & t\sigma\ (u\sigma) \end{array}$$

and, if l is a proof of $\mathsf{Eq}_\Gamma\ \sigma_1\ \sigma_2$

$$\begin{array}{lcl} xl & = & l(x) \\ \langle t\ u\rangle l & = & (t\sigma_1|\ \langle u\rangle l); (\langle t\rangle l\ (u\sigma_2)) \\ \langle \lambda x.t\rangle l\ a & = & \langle t\rangle (l, 1) \end{array}$$

The composition $p; q : \mathsf{Eq}_A\ a_1\ a_3$ for $p : \mathsf{Eq}_A\ a_1\ a_2$ and $q : \mathsf{Eq}_A\ a_2\ a_3$ is defined by induction on the type A. We have for $A = o$

$$p; q = (q.1 \circ p.1, p.2 \circ q.2)$$

and, at higher type

$$(p; q)\ a = (p\ a); (q\ a)$$

Similary the reflexivity proof $1 : \mathsf{Eq}_A\ c\ c$ is defined by $1 = (\lambda x.x, \lambda x.x)$ for $A = o$ and $1\ a = 1$ at higher type.

We think that this is the kind of calculus D. Turner had in mind in the work [5]. Already for simple type calculus, it does not seem so easy to design such a calculus without the help of the model following Gandy's extensionality proof.

We can use these laws to effectively compute equality proofs. For instance, if N_1 is of type o with inhabitant $0 : N_1$ we have an element

$$\lambda X.(\lambda x.\lambda y.x, \lambda f.f\ 0) : \forall X : o.\mathsf{Eq}_o\ X\ (N_1 \to X)$$

and hence

$$\lambda X.(\lambda x.\lambda y.x, \lambda z.z\ 0) : \mathsf{Eq}_{o\to o}\ f\ g$$

where $f = \lambda X.X$ and $g = \lambda X.N_1 \to X$. On the other hand, we have $\Phi = \lambda h.(h\ N_1) \to (h\ N_1)$ which is of type $(o \to o) \to o$. We deduce that we have

$$\Phi()|\lambda X.(\lambda x.\lambda y.x, \lambda z.z\ 0) : \mathsf{Eq}_o\ (\Phi\ f)\ (\Phi\ g)$$

This means that we get a proof of this equivalence of $\Phi\ f$ and $\Phi\ g$. We can in this way transform any proof of $\Phi\ f$ to a proof of $\Phi\ g$. For instance if we apply this to $\lambda x.x$ which is a proof of $\Phi\ f = N_1 \to N_1$, we get a proof of $\Phi\ g = (N_1 \to N_1) \to (N_1 \to N_1)$ which is $\lambda z.\lambda x.z\ 0$.

3 Simple type theory

We give the rules of type formation

$$\frac{\Gamma \vdash}{\Gamma \vdash N : U} \qquad \frac{\Gamma \vdash}{\Gamma \vdash N_2 : U} \qquad \frac{\Gamma \vdash}{\Gamma \vdash o : U}$$

$$\frac{\Gamma \vdash X_1 : U \quad \Gamma \vdash X_2 : U}{\Gamma \vdash X_1 \to X_2 : U}$$

and for forming propositions

$$\frac{\Gamma \vdash \varphi_1 : o \quad \Gamma \vdash \varphi_2 : o}{\Gamma \vdash \varphi_1 \to \varphi_2 : o} \qquad \frac{\Gamma \vdash \varphi_1 : o \quad \Gamma \vdash \varphi_2 : o}{\Gamma \vdash \varphi_1 \wedge \varphi_2 : o}$$

$$\frac{\Gamma, x : X \vdash \varphi : o}{\Gamma \vdash \forall x : X.\varphi : o} \qquad \frac{\Gamma, x : X \vdash \varphi : o}{\Gamma \vdash \exists x : X.\varphi : o}$$

We have special propositions $\bot\colon o$ and $\top : o$ with $0 : \top$.

We have the rules of natural deductions:

$$\frac{\Gamma \vdash p : \varphi_1 \to \varphi_2 \quad \Gamma \vdash q : \varphi_1}{\Gamma \vdash \mathsf{app}(p, q) : \varphi_2} \qquad \frac{\Gamma \vdash p : \forall x : X.\varphi \quad \Gamma \vdash t : X}{\Gamma \vdash \mathsf{app}(p, t) : \varphi(x/t)}$$

$$\frac{\Gamma, x : X \vdash p : \varphi}{\Gamma \vdash \lambda x.p : \forall x : X.\varphi} \qquad \frac{\Gamma, x : \varphi_1 \vdash p : \varphi_2}{\Gamma \vdash \lambda x.p : \varphi_1 \to \varphi_2}$$

$$\frac{\Gamma \vdash p_i : \varphi_i}{\Gamma \vdash (p_1, p_2) : \varphi_1 \wedge \varphi_2} \qquad \frac{\Gamma \vdash r : \varphi_1 \wedge \varphi_2}{\Gamma \vdash \mathsf{p}\ r : \varphi_1} \qquad \frac{\Gamma \vdash r : \varphi_1 \wedge \varphi_2}{\Gamma \vdash \mathsf{q}\ r : \varphi_2}$$

$$\frac{\Gamma \vdash t : X \quad \Gamma \vdash p : \varphi(x/t)}{\Gamma \vdash (t, p) : \exists x : X.\varphi} \qquad \frac{\Gamma \vdash r : \exists x : X.\varphi \quad \Gamma \vdash p : \forall x : X.\varphi \to \psi}{\Gamma \vdash \mathsf{Elim}(r, p) : \psi}$$

with the computation rule $\mathsf{Elim}((t, p), q) = \mathsf{app}(\mathsf{app}(q, t), p)$.

We add

$$\frac{\Gamma \vdash p : \bot \quad \Gamma \vdash X : U}{\Gamma \vdash \mathsf{Exit}\ p : X} \qquad \frac{\Gamma \vdash p : \bot \quad \Gamma \vdash \varphi : o}{\Gamma \vdash \mathsf{Exit}\ p : \varphi}$$

We define the extensional equality $\mathsf{Eq}_X\ a_1\ a_2 : o$ by induction on X. We take $\mathsf{Eq}_o\ p_1\ p_2$ to be $(p_1 \to p_2) \wedge (p_2 \to p_1)$. The equality on N_2 is defined by cases

$$\mathsf{Eq}_{N_2}\ 0\ 0 = \mathsf{Eq}_{N_2}\ 1\ 1 = \top \qquad \mathsf{Eq}_{N_2}\ 0\ 1 = \mathsf{Eq}_{N_2}\ 1\ 0 = \bot$$

Finally we define $\mathsf{Eq}_{X \to Y}\ f_1\ f_2 = \forall x : X.\mathsf{Eq}_Y\ \mathsf{app}(f_1, x)\ \mathsf{app}(f_2, x)$.

It is possible to define by induction on X

$$\frac{\Gamma \vdash t : X}{\Gamma \vdash \mathsf{ref}\ t : \mathsf{Eq}_X\ t\ t} \qquad \frac{\Gamma \vdash \alpha : \mathsf{Eq}_X\ t_1\ t_2}{\Gamma \vdash \alpha^{-1} : \mathsf{Eq}_X\ t_2\ t_1} \qquad \frac{\Gamma \vdash \alpha : \mathsf{Eq}_X\ t_1\ t_2 \quad \Gamma \vdash \beta : \mathsf{Eq}_X\ t_2\ t_3}{\Gamma \vdash \alpha; \beta : \mathsf{Eq}_X\ t_1\ t_3}$$

with the computation rules

$$\mathsf{ref}\ 0 = \mathsf{ref}\ 1 = 0 \qquad \mathsf{app}(\alpha^{-1}, t) = \mathsf{app}(\alpha, t)^{-1} \quad \mathsf{app}(\alpha; \beta, t) = \mathsf{app}(\alpha, t); \mathsf{app}(\beta, t) \quad 0^{-1} = 0; 0 = 0$$

4 Addition of subset types

We add the type formation

$$\frac{\Gamma \vdash X : U \quad \Gamma \vdash \varphi : X \to o}{\Gamma \vdash \{X \mid \varphi\} : o}$$

with the rules

$$\frac{\Gamma \vdash t : X \quad \Gamma \vdash p : \mathsf{app}(\varphi, t)}{\Gamma \vdash (t, p) : \{X \mid \varphi\}} \qquad \frac{\Gamma \vdash u :: \{X \mid \varphi\}}{\Gamma \vdash \mathsf{p}\ u : X} \qquad \frac{\Gamma \vdash u :: \{X \mid \varphi\}}{\Gamma \vdash \mathsf{q}\ u : \mathsf{app}(\varphi, \mathsf{p}\ u)}$$

5 The Axiom of Description

6 Addition of large structures

References

[1] T. Altenkirch. Extensional Equality in Intensional Type Theory. LICS 1999.

[2] R. Gandy. On The Axiom of Extensionality -Part I. The Journal of Symbolic Logic, Vol. 21, 1956.

[3] M. Hofmann. *Extensional concepts in intensional type theory.* Ph.D. thesis, University of Edinburgh, 1995.

[4] M. Hofmann and Th. Streicher. The groupoid interpretation of type theory. in *25 years of Type Theory*, 1996.

[5] D. Turner. Extensional Type Theory. Talk, recorder in proceeding of Båstad, 1989.

[6] V. Voevodsky. Formalization of Mathematics and Homotopy Theory. Lecture at the IAS, Princeton, 2006.