The paradox of trees in Type Theory

Thierry Coquand

INRIA, and University of Göteborg/Chalmers

Abstract.

We show how to represent a paradox similar to Russell's paradox in Type Theory with W-types and a type of all types, and how to use this in order to represent a fixed-point operator in such a theory. It is still open whether or not such a construction is possible without the W-type.

Introduction.

It is known that Martin-Löf's Type Theory with one universe is inconsistent if this universe contains a name of itself (cf. [5,6,7,8]). Though it is possible under this hypothesis to produce a paradox similar to the one of Russell if we have an extensional equality (i.e. the equality described in [1]; see for instance [7] for a description of this paradox), it is not known yet if such a paradox occurs with the more intensional equality of Type Theory (as described in [3][1]), if we assume only as type constructors the Π and the Σ type operators (see [5,6] for a discussion of this problem). This question can be precised by asking whether there exists a term of Type Theory with a type of all types that has the property of reducing to itself in a non-trivial way (as for the term $\mathbf{app}(\lambda x.\mathbf{app}(x,x), \lambda x.\mathbf{app}(x,x))$ in untyped λ-calculus).

In this note, we show how to build such a paradox with the "intensional" equality of Type Theory, if we assume also as type constructor the W type operator. We then explain how to use this paradox to build a fixed-point operator.

1. An informal presentation of the paradox of trees.

We assume that we have a basic intuition of the notion of a tree. If we have already a collection of trees, we collect them together under a common root to form a new tree. Each member of the collection will be said to be an immediate subtree of the new tree and a tree is said to be normal if it is not equal to any of its immediate subtree. We form then a new tree R with the collection of all trees that are normal.

If R is equal to one of its immediate subtrees (that are all normal), then R is normal and hence we have a contradiction. Hence, R is not equal to any of its immediate subtree. So R is normal, and, being normal, is an immediate subtree of R. We thus have a tree that is both normal and equal to one of its subtree, which is a contradiction (this "concrete" version of Russell's paradox was pointed out to the author by Pr. de Bruijn).

[0] This research was partly supported by ESPRIT Basic Research Action "Logical Frameworks".

[1] There are three kind of equalities: the equality of [1] will be called "extensional" equality, the equality of [3], i.e. intensional I-set, will be called "intensional" equality, and the conversion, i.e. judgemental equality, will be called "definitional" equality.

Typeset by $\mathcal{AMS}$-TeX

2. Representation of the paradox of trees in Type Theory.

Assume we have a universe U with one element $u : U$, and the definitional equality $T(u) = U$.

We have also, as usual in Type Theory, the operator $\pi(a, b) : U\ [a : U, b(x) : U[x : T(a)]]$, with the definitional equality

$$T(\pi(a, b)) = (\Pi x : T(a))T(b(x))\ [a : U, b(x) : U[x : T(a)]],$$

the operator $\sigma(a, b) : U\ [a : U, b(x) : U[x : T(a)]]$, with the definitional equality

$$T(\sigma(a, b)) = (\Sigma x : T(a))T(b(x))\ [a : U, b(x) : U[x : T(a)]],$$

and the operator $w(a, b) : U\ [a : U, b(x) : U[x : T(a)]]$, with the definitional equality

$$T(w(a, b)) = (Wx : T(a))T(b(x))\ [a : U, b(x) : U[x : T(a)]].$$

We assume also an operator $i(a, x, y) : U\ [a : U, x, y : T(a)]$ with the definitional equality

$$T(i(a, x, y)) = I(T(a), x, y)\ [a : U, x, y : T(a)].$$

We will write $\mathbf{ref}(x) : I(A, x, x)\ [x : A]$ the constructor of the equality type. The elimination rule is

$$\mathbf{J}(c, d) : C(a, b, c)\ [a, b : A, c : I(A, a, b), d(x) : C(x, x, \mathbf{ref}(x))\ [x : A]],$$

with the conversion rule

$$\mathbf{J}(\mathbf{ref}(a), d) = d(a) : C(a, a, \mathbf{ref}(a))\ [a : A, d(x) : C(x, x, \mathbf{ref}(x))\ [x : A]].$$

Actually, it would be possible to use Leibniz equality instead, which is definable if we have a type of all types: $I(A, x, y)$ will be defined as $(\Pi P : A \to U)T(P(y)) \to T(P(x))$. However, it does not seem possible to use such impredicative codings for the definition of Σ or W in such a way that we get a fixed-point (and this seems related to the problem of the representation of the predecessor function with the impredicative coding of data types).

We will assume $x_0 : U$ and define $X_0 = T(x_0)$. This type X_0 may be thought of as the empty type, but actually, in the following reasoning, we don't have to assume anything special about x_0. Relatively to x_0, we can define a negation $\neg x = \pi(x, (y)x_0)\ [x : U]$, such that $T(\neg x) = (\Pi y : T(x))X_0$.

We define then the type of trees by $\mathbf{Tree} = W(U, T)$. Notice that the type $\mathbf{Tree}$ has a name $\mathbf{tree}$ in U, namely $\mathbf{tree} = w(u, (y)y)$, which is such that $T(\mathbf{tree}) = \mathbf{Tree}$. The type $\mathbf{Tree}$ can also be seen as an inductively defined data type with one constructor

$$\mathbf{sup}(x, f) : \mathbf{Tree}\ [x : U, f(y) : \mathbf{Tree}[y : T(x)]].$$

In particular, we can introduce a constant $\mathbf{normal}(x) : U\ [x : \mathbf{Tree}]$ with the definitional equality

$$\mathbf{normal}(\mathbf{sup}(x, f)) = \pi(x, (y)\neg i(\mathbf{tree}, f(y), \mathbf{sup}(x, f))) : U.$$

We have then

$$T(\mathbf{normal}(\mathbf{sup}(x, f))) = (\Pi y : T(x))(\Pi z : I(\mathbf{Tree}, f(y), \mathbf{sup}(x, f)))X_0.$$

That is, $\mathbf{normal}(x) : U\ [x : \mathbf{Tree}]$ expresses intuitively "that x is not equal to any of its immediate subtree", and corresponds to the property of being "normal" presented in the first section.

We can now define the type of normal trees as $\mathbf{nt} = \sigma(\mathbf{tree}, \mathbf{normal}) : U$. The first projection defines an application $\mathbf{p}(y) : \mathbf{Tree}\ [y : T(\mathbf{nt})]$, and the second projection an application $\mathbf{q}(y) : \mathbf{normal}(\mathbf{p}(y))\ [y : T(\mathbf{nt})]$. We define then the tree of normal trees as $\mathbf{r} = \mathbf{sup}(\mathbf{nt}, \mathbf{p}) : \mathbf{Tree}$. It is straigthforward to build a proof **lemma** of $T(\mathbf{normal}(\mathbf{r}))$. Then the reader can check that it is possible to take

$$\mathbf{lemma} = (\lambda y)(\lambda z)\mathbf{app}(\mathbf{app}(\mathbf{app}(\mathbf{J}(z, (\lambda u)(\lambda v)v), \mathbf{q}(y)), y), z) : T(\mathbf{normal}(\mathbf{r})).$$

From this a proof of the proposition X_0 is easily built:

$$\mathbf{russell} = \mathbf{lemma}((\mathbf{r}, \mathbf{lemma}), \mathbf{ref}(\mathbf{r})) : X_0.$$

We can then check that **russell** reduces to itself in a non-trivial way.

Since X_0 was an arbitrary type which has a name in U, we have that, if U has a name of itself, then any type named by an element of U is inhabited.

This derivation can be seen as a derivation of Russell paradox in Type Theory via the coding of sets as trees. Such a coding is used in [4], but the equality and the membership relations are defined by induction over a W type. It is enough in order to derive Russell paradox to use as equality the intensional equality, and as membership relation the relation of being an immediate subtree.

3. A fixed-point operator.

We can use the trick described in [5,6] to build a fixed-point operator on types that have a name in U. We take $g(x) : X_0\ [x : X_0]$ to be an arbitrary function and we show how to build a fixed-point of g (remember that, in the above derivation, we only use from X_0 the fact that it is a type with a name in U, that is X_0 denotes an arbitrary type with a name in U). Remark that we know that the type $(\Sigma x : X_0)I(X_0, x, g(x))$ is inhabited (since this type has a name in U), but this is not enough a priori to conclude, since the equality is only "intensional".

Here is a short description of the construction of a fixed-point of g: the proof of $T(\mathbf{normal}(\mathbf{r}))$ has the form $(\lambda y)(\lambda z)t[y,z] : T(\mathbf{normal}(\mathbf{r}))$ with

$$t[y,z] : X_0\ [y : T(\mathbf{nt}), z : I(\mathbf{Tree}, \mathbf{p}(y), \mathbf{r})].$$

We change this proof to the following term

$$\mathbf{lemma}_g = (\lambda y)(\lambda z)g(t[y,z]) : T(\mathbf{normal}(\mathbf{r})).$$

The new proof of X_0 becomes

$$\mathbf{russell}_g = \mathbf{lemma}_g((\mathbf{r}, \mathbf{lemma}_g), \mathbf{ref}(\mathbf{r})).$$

Then we can check that we have

$$\mathbf{russell}_g = g(\mathbf{russell}_g) : X_0.$$

That is, we just have built a fixed-point of $g(x) : X_0\ [x : X_0]$.

If N has a name in U then we have $\omega : N$ with $\omega = S(\omega) : N$. It is then possible to build a proof of $I(N, 0, S(0))$. We first consider the function f defined by primitive recursion by $f(0) = S(0) : N$ and $f(S(x)) = 0 : N\ [x : N]$. By recursion, we can build an element of $(p : N)I(N, f^{2p}(0), 0)$ and of $(p : N)I(N, f^{2p+1}(0), S(0))$. This shows that $(n : N)I(N, n, S(n)) \to I(N, 0, S(0))$ is provable. If $\omega = S(\omega) : N$, we have a proof of $I(N, \omega, S(\omega))$ and hence of $I(N, 0, S(0))$. From this, we deduce that the type $I(A, x, y)$ is inhabited for all type A, and $x, y : A$, by considering the function $g : N \to A$ such that $g(0) = x : A$ and $g(S(n)) = y : A\ [n : N]$. This shows that the intensional equality becomes trivial (but, as shown for instance in [3], the definitional equality is non trivial).

If N_0 has a name in U, then any type is inhabited. We get then a general fixed point for any function $f(x) : A\ [x : A]$: it is enough to iterate the given function "ω times" on any element of A.

4. Russell and Burali-Forti paradox.

It turns out that in this formalisation of the tree paradox inside Intuitionistic Type Theory, it is possible to shorten the proof and then get a paradox that can be seen as the expression of Burali-Forti paradox.

The remark is that, according to the semantics of Type Theory, the interpretation of objects of type **Tree** are well-founded trees, and hence are all normal, i.e. not equal to any of their immediate subtrees. This remark can be formalised completely, that is, one can build an object of type $(\Pi x : \mathbf{Tree})\mathbf{normal}(x)$. In particular, if we consider the tree $\mathbf{b} = \mathbf{sup}(\mathbf{tree}, (\lambda x)x) : \mathbf{Tree}$, we have an object of type $\mathbf{normal}(\mathbf{b})$. But this is clearly contradictory, since $\mathbf{b}$ is an immediate subtree of itself. $\mathbf{b}$ can be thought of as the tree of all well-founded trees. Another intuition is that, if we represent sets by trees, then the foundation axiom is satisfied because all trees are well-founded, Hence, we cannot have a set of all sets.

It is possible to check that this paradox reduces also to itself. It would be interesting to try to translate this paradox in a pure type system with only dependent product and a type of all types. Notice that, in the present version, this paradox uses only W types and dependent products.

Acknowledgments.

P. Martin-Löf noticed that the result of the existence of a fixed-point operator can be seen as the converse of the construction of [2] that builds a universe with a name of itself in partial type theory. I want also to thank him for helpful comments.

References

1. Martin-Löf, P., *Intuitionistic Type Theory*, Bibliopolis, 1984.
2. Palmgren, E., *A construction of Type:Type in Martin-Löf's partial type theory with one universe*, To appear in Journal of Symbolic Logic.
3. Palmgren, E. and Stoltenberg-Hansen, V., *Domain Interpretations of Intuitionistic Type Theory*, Uppsala University.
4. Aczel, P., *The type theoretic interpretation of constructive set theory*, Logic Colloquium '77 (1978), 55–66, A.Macintyre et al., editors.
5. Meyer, A. and Reinhold, M.B., *"Type" is not a type*, Principles of Programming language, ACM.
6. Howe, D., *The Computational Behaviour of Girard's Paradox*, Symposium on Logic in Computer Science, Ithaca, New York.
7. Hayashi, S. and Nakano, H., Communication in the TYPES electronic forum (August 7, 1987), types@theory.lcs.mit.edu.
8. Girard, J.Y., *Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieure*, Thèse d'Etat, Paris VII, 1972.