A variation of Reynolds-Hurkens Paradox

Thierry Coquand, University of Gothenburg, Sweden

Introduction

We present a variation of Hurkens paradox [8], itself being a variation of Reynolds "paradox" [10], as used in [4]. We first explain a related paradox in higher order logic, which can be seen as a variation of Russell's paradox. We then show how this paradox can be formulated in system $\lambda \mathsf{U}^-$. We finally argue that an analysis of the computational behavior of this paradox requires to extend existing type systems with a first class notion of definitions and head linear reductions, as advocated by N.G. de Bruijn [6].

1 Some paradoxes in minimal Higher-Order logic

We first present some paradoxes in some extensions of the system $\lambda\mathsf{HOL}$, minimal Higher-Order logic, described in [7]. This system can be seen as a minimal logic version of higher-order logic introduced by A. Church [1]. With the notation of [7], it has sorts $*, \Box, \Delta$ with $* : \Box$ and $\Box : \Delta$ and the rules

$$(*, *),\ \ (\Box, \Box),\ \ (\Box, *)$$

We denote by $X, Y, \dots$ types of this system.

We can define $\mathsf{Pow} : \Box \to \Box$ by $\mathsf{Pow}\ X = X \to *$ and $T : \Box \to \Box$ by $T\ X = \mathsf{Pow}\ (\mathsf{Pow}\ X)$.

Note that T defines a *judgmental* functor: if $f : X \to Y$ we can define $T\ f : T\ X \to T\ Y$ by

$$T\ f\ F\ q = F\ (\lambda_{x:X} q\ (f\ x))$$

and we also have if furthermore $g : Y \to Z$ the judgemental equality (here β-conversion [7]) $T\ (g \circ f) = (T\ g) \circ (T\ f)$ defining $g \circ f$ as $\lambda_{x:X} g\ (f\ x)$.

We assume in this section to have a type $A : \Box$ together with two maps $\mathsf{intro} : T\ A \to A$ and $\mathsf{match} : A \to T\ A$.

We explain now how to derive simple paradoxes assuming some convertibility properties of these maps.

1.1 A variation of Russell's paradox

The first version is obtained by assuming that we have $\mathsf{match}\ (\mathsf{intro}\ u)$ convertible to u, i.e. $T\ A$ is a judgemental retract of A.

Intuitively, we expect $\mathsf{Pow}\ A$ to be a retract of $T\ A$, and this would imply that $\mathsf{Pow}\ A$ is a retract of A and we should be able to deduce a contradition by Russell's paradox. One issue with this argument is that it holds only using some form of *extensional* equalities, and we work in an intensional setting. One way to solve this issue is to work with Partial Equivalence Relations; this is what was done in [4]. The work [8], suggests that there should be a more direct way to express this idea, and this is what we present here.

The contradiction is obtained as follows. We first define a relation $C : \mathsf{Pow}\ A \to \mathsf{Pow}\ A$

$$C\ p\ x\ \ =\ \ p\ x \to \neg(\mathsf{match}\ x\ p)$$

where, as usual, we define $\bot : *$ by $\bot = \forall_{p:*} p$ and $\neg : * \to *$ by $\neg\ p = p \to \bot$. We can then define $p_0 : \mathsf{Pow}\ A$

$$p_0\ x\ \ =\ \ \forall_{p:\mathsf{Pow}\ A} C\ p\ x$$

We can also define $X_0 : T\ A$

$$X_0\ p\ =\ \forall_{x:A} C\ p\ x$$

and $x_0 : A$ as $x_0 = \mathsf{intro}\ X_0$. We can then build $l_1 : X_0\ p_0 = \mathsf{match}\ x_0\ p_0$

$$l_1\ x\ h\ =\ h\ p_0\ h$$

and $l_2 : p_0\ x_0$ by

$$l_2\ p\ h\ h_1\ =\ h_1\ x_0\ h\ h_1$$

But this is a contradiction since $\mathsf{match}\ x_0 = \mathsf{match}\ (\mathsf{intro}\ X_0) = X_0$ by hypothesis, and hence $l_2\ p_0\ l_2\ l_1$ is of type $\perp$.

We can summarize this discussion as follows.

Theorem 1.1 *In* λHOL*, we cannot have a type A such that* $\mathsf{Pow}\ (\mathsf{Pow}\ A)$ *is a judgemental retract of A.*

This can be seen as a variation of Russell/Cantor's paradox, which states that $\mathsf{Pow}\ A$ cannot be a retract of A. Here we state that $T\ A$ cannot be a retract of A.

1.2 A refinement

We define $\delta : A \to A$ by $\delta = \mathsf{intro} \circ \mathsf{match}$ and assume the judgemental equality

$$\mathsf{match} \circ \mathsf{intro} = T\ \delta \tag{1}$$

which implies $\mathsf{match}\ (\delta\ x)\ p = \mathsf{match}\ x\ (p \circ \delta)$.

We now (re)define $p_0 : \mathsf{Pow}\ A$

$$p_0\ x\ =\ \forall_{p:\mathsf{Pow}\ A}\ p\ (\delta\ x) \to \neg(\mathsf{match}\ x\ p)$$

and $X_0 : T\ A$ as before

$$X_0\ p\ =\ \forall_{x:A}\ p\ x \to \neg(\mathsf{match}\ x\ p)$$

and $x_0 : A$ as $x_0 = \mathsf{intro}\ X_0$. Using the judgemental equality (1), it is possible to build

$$s_1 : \forall_x\ p_0\ x \to p_0\ (\delta\ x) \qquad\qquad s_2 : \forall_p\ X_0\ p \to X_0\ (p \circ \delta)$$

by $s_1\ x\ h\ p\ =\ h\ (p \circ \delta)$ and $s_2\ p\ h\ x\ =\ h\ (\delta\ x)$.

We can now define and $l_0 : \forall_{p:\mathsf{Pow}\ A}\ p\ x_0 \to \neg(X_0\ p)$ by

$$l_0\ p\ h\ h_0\ =\ h_0\ x_0\ h\ (s_2\ p\ h_0)$$

using (1) and $l_1 : X_0\ p_0$ by

$$l_1\ x\ h\ =\ h\ p_0\ (s_1\ x\ h)$$

and $l_2 : p_0\ x_0$ by $l_2\ p = l_0\ (p \circ \delta)$.

For this, we use the judgemental equality $\mathsf{match}\ (\delta\ x)\ p = \mathsf{match}\ x\ (p \circ \delta)$, consequence of (1).

We can then form the term $l_0\ p_0\ l_2\ l_1$ which is of type $\perp$.

We thus get the following result, using $T\ X = \mathsf{Pow}\ (\mathsf{Pow}\ X)$.

Theorem 1.2 *In* λHOL*, we cannot have a type A with two maps* $\mathsf{intro} : T\ A \to A$ *and* $\mathsf{match} : A \to T\ A$ *with* $\mathsf{match} \circ \mathsf{intro}$ *convertible to* $T\ (\mathsf{intro} \circ \mathsf{match})$.

2 An encoding in $\lambda\mathsf{U}^-$

2.1 Weak representation of data type

Using the notations of [7] the system $\lambda\mathsf{U}^-$ has also sorts $*, \Box, \Delta$ with $* : \Box$ and $\Box : \Delta$ and the rules

$$(*, *),\ (\Box, \Box),\ (\Box, *),\ (\Delta, \Box)$$

We explain in this section why the refined paradox has a direct encoding in the system $\lambda\mathsf{U}^-$.

As before, T defines a judgemental functor: if $f : X \to Y$ we can define $T\ f : T\ X \to T\ Y$ by

$$T\ f\ F\ q = F\ (\lambda_{x:X} q\ (f\ x))$$

and we also have if furthermore $g : Y \to Z$ the judgemental equality $T\ (g \circ f) = (T\ g) \circ (T\ f)$ defining $g \circ f$ as $\lambda_{x:X} g\ (f\ x)$.

A T-algebra is a type $X : \Box$ together with a map $f : T\ X \to X$.

Following Reynolds [10, 11], we represent $A : \Box$ by

$$A = \Pi_{X:\Box}(T\ X \to X) \to X$$

It can be seen as a weak representation of a data type. If we have $X : \Box$ and $f : T\ X \to X$ we can define $\iota\ f : A \to X$ by $\iota\ f\ a = a\ X\ f$. We can then define $\mathsf{intro} : T\ A \to A$ by $\mathsf{intro}\ u\ X\ f = f\ (T\ (\iota\ f)\ u)$, and we have the conversion

$$(\iota\ f) \circ \mathsf{intro} = f \circ (T\ (\iota\ f)) \tag{2}$$

This expresses that the following diagram commutes strictly

$$\begin{array}{ccc} T\ A & \xrightarrow{T\ (\iota\ f)} & T\ X \\ \downarrow{\scriptstyle \mathsf{intro}} & & \downarrow{\scriptstyle f} \\ A & \xrightarrow{(\iota\ f)} & X \end{array}$$

So A, intro represents a *weak* initial T-algebra.

We define next $\mathsf{match} : A \to T\ A$ by $\mathsf{match} = \iota\ (T\ \mathsf{intro})$. Using the conversion (2), we have

$$\mathsf{match} \circ \mathsf{intro} = (T\ \mathsf{intro}) \circ (T\ \mathsf{match}) = T\ (\mathsf{intro} \circ \mathsf{match})$$

This is the required conversion (1) and we get in this way an encoding of Theorem 1.2.

2.2 Some variations

In [8], Hurkens uses instead

$$B = \Pi_{X:\Box}(T\ X \to X) \to T\ X \tag{3}$$

He then develops a short paradox using this type B, but with a different intuition, which comes from Burali-Forti paradox. The variation we present in this note starts instead from the remark that $T\ A$ cannot be a retract of A. In [4], we also use this idea, but with a more complex use of partial equivalence relations, in order to build a strong initial T-algebra from a weak initial T-algebra. This was following Reynolds' informal argument in [10],

The same argument from Theorem 1.2 can use the encoding (3) instead. We define then

$$\iota : \Pi_{X:\Box}(T\ X \to X) \to B \to X$$

by

$$\iota\ X\ f\ b = f\ (b\ X\ f)$$

and $\mathsf{intro} : T\ B \to B$ by

$$\mathsf{intro}\ v\ X\ f = T\ (\iota\ f)\ v$$

We then have the choice for defining $\mathsf{match} : B \to T\ B$. We can use

$$\mathsf{match} = \iota\ (T\ B)\ \mathsf{intro}$$

as before. Maybe surprisingly, we also can use

$$\mathsf{match}\ b = b\ B\ \mathsf{intro}$$

In both cases, we get the judgemental equality $\mathsf{match} \circ \mathsf{intro} = T\ (\mathsf{intro} \circ \mathsf{match})$ required for the use of Theorem 1.2.

3 Computational behavior

For the paradox corresponding to Theorem 1.1, we have the following looping behavior with a term reducing to itself (in two steps) by *head linear reduction*

$$\begin{array}{lll} l_2\ p_0\ l_2\ l_1 & \to & l_1\ x_0\ l_2\ l_1 \\ & \to & l_2\ p_0\ l_2\ l_1 \\ & \to & \ldots \end{array}$$

3.1 Family of looping combinators

The paradox corresponding to Theorem 1.2 does not produce a term that reduces to itself

$$\begin{array}{lll} l_0\ p_0\ l_2\ l_1 & \to & l_1\ x_0\ l_2\ (s_2\ p_0\ l_1) \\ & \to & l_2\ p_0\ (s_1\ x_0\ l_2)\ (s_2\ p_0\ l_1) \\ & \to & l_0\ (p_0 \circ \delta)\ (s_1\ x_0\ l_2)\ (s_2\ p_0\ l_1) \\ & \to & s_2\ p_0\ l_1\ x_0\ (s_1\ x_0\ l_2)\ (s_2\ (p_0 \circ \delta)\ (s_2\ p_0\ l_1)) \\ & \to & l_1\ (\delta\ x_0)\ (s_1\ x_0\ l_2)\ (s_2\ (p_0 \circ \delta)\ (s_2\ p_0\ l_1)) \\ & \to & \ldots \end{array}$$

Like for Hurkens' paradox however, we obtain a term that reduces to itself if we forget types in abstraction [8].

In [2], I analysed another paradox, closer to Girard's original formulation (as was found out later by H. Herbelin and A. Miquel, a slight variation of this paradox can be expressed in System $\lambda\mathsf{U}^-$.) At about the same time, A. Meyer and M. Reinholdt [9], suggested a clever use of Girard's paradox for expressing a fixed-point combinator. While implementating this paradox [2], it was possible to check that, contrary to what [9] was hinting, the term representing this paradox was not reducing to itself[1]. A. Meyer found out then that it was however possible to use this paradox and produce a family of looping combinators instead, i.e. a term which has the same Böhm tree as one of a fixed-point combinator. A corollary, following [9], is that type-checking is undecidable for $\mathsf{type} : \mathsf{type}$.

3.2 Definitions and Head linear Reduction

As discussed in [8], using the notion of *definition* is essential, even for "small" terms, for representing these paradoxes in an undertandable way. As was discovered in Automath [6], in a type system with *dependent* types, one cannot reduce definitions to abstractions and applications like in simply typed lambda calculus. Indeed, the representation of

$$\mathsf{let}\ x : A = e_0\ \mathsf{in}\ e_1$$

by $(\lambda_{x:A} e_1)\ e_0$ can be incorrect, since the definition $x : A = e_0$ can be used in the type-checking of e_1.

Furthermore, in order to understand the computational behavior of the paradox, the use of *head linear reduction*, which plays an important role in [6], is convenient. This is what was done when presenting above the computational behavior of various paradoxes, with a periodic behavior for the first example and a non periodic behavior for the paradox in $\lambda\mathsf{U}^-$. This use may also be relevant for understanding large proofs.

[1] It would be interesting to go back to this paradox and check if it reduces to itself when removing types in abstractions.

Conclusion

In this note, we presented a variation of Hurkens' paradox [8] and a paradox inspired by Reynolds [4]. This paradox can be seen as a refinement of the simple paradox presented in Theorem 1.1. The problem is that in the encoding in $\lambda\mathsf{U}^-$, we don't get that $T\ A$ is a *judgmental* retract of A^2. It is possible however to still use a weaker judgemental equality and derive a relatively simple paradox[3].

References

[1] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[2] Th. Coquand. An Analysis of Girard's Paradox. In *Proceedings of the Symposium on Logic in Computer Science (LICS '86), Cambridge, Massachusetts, USA, June 16-18, 1986*, pages 227–236. IEEE Computer Society, 1986.

[3] Th. Coquand. The Paradox of Trees in Type Theory. *BIT*, 32(1):10–14, 1992.

[4] Th. Coquand. A new paradox in type theory. In *Logic, methodology and philosophy of science IX. Proceedings of the ninth international congress of logic, methodology and philosophy of science, Uppsala, Sweden, August 7-14, 1991*, pages 555–570. Amsterdam: North-Holland, 1994.

[5] Th. Coquand and Ch. Paulin. Inductively defined types. In Per Martin-Löf and Grigori Mints, editors, *COLOG-88, International Conference on Computer Logic, Tallinn, USSR, December 1988, Proceedings*, volume 417 of *Lecture Notes in Computer Science*, pages 50–66. Springer, 1988.

[6] N. G. de Bruijn. Generalizing Automath by means of a lambda-typed lambda calculus. Mathematical logic and theoretical computer science, Lect. Notes Pure Appl. Math. 106, 71-92 (1987)., 1987.

[7] Herman Geuvers. (In)consistency of extensions of higher order logic and type theory. In *Types for proofs and programs. International workshop, TYPES 2006, Nottingham, UK, April 18–21, 2006. Revised selected papers.*, pages 140–159. Berlin: Springer, 2007.

[8] A. J. C. Hurkens. A simplification of Girard's paradox. In Mariangiola Dezani-Ciancaglini and Gordon Plotkin, editors, *Typed Lambda Calculi and Applications*, pages 266–278, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[9] A. R. Meyer and M. B. Reinhold. "Type" Is Not A Type. In *Conference Record of the Thirteenth Annual ACM Symposium on Principles of Programming Languages, St. Petersburg Beach, Florida, USA, January 1986*, pages 287–295. ACM Press, 1986.

[10] John C. Reynolds. Polymorphism is not set-theoretic. In Gilles Kahn, David B. MacQueen, and Gordon D. Plotkin, editors, *Semantics of Data Types, International Symposium, Sophia-Antipolis, France, June 27-29, 1984, Proceedings*, volume 173 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 1984.

[11] John C. Reynolds and Gordon D. Plotkin. On functors expressible in the polymorphic typed lambda calculus. *Inf. Comput.*, 105(1):1–29, 1993.

[2] This problem was presented in [5] as one main motivation for the primitive introduction of inductive definitions.

[3] We were not able however to refine in a similar way the paradox of trees [3], to obtain a new paradox in $\lambda\mathsf{U}^-$.