Russell's paradoxes in higher order logic

Frédéric Ruyer *†

28 mars 2007

1 Introduction

The purpose of this paper is to present direct applications of Russell's paradox to some formal type systems. The idea is to use fixpoints of a sort operator allowing intuitively to build a one-to-one correspondance between a set s and 2^{2^s} (the set of parts of the set of parts of s). It is the same idea that is used by Coquand in [1], but here we work more directly with the behaviour of terms.

The inconsistency of polymorphic higher order logic (or system U^-) is known since the work of Coquand [2], and follows the inconsistency of system U discovered by Girard [3] and analyzed in [1]. The proof presented here is self-contained and does not refer neither to any notion of initial algebra, nor to Reynold's result about the non existence of set theoretical models of polymorphism [5].

Moreover, in [1], Coquand thought it was not possible to derive such a paradox in U.

We show the inconsistency of U^-, and of another type system where sorts are built with $\exists$ and $\rightarrow$, using a fixpoint which is the dual of the precedent one. We also make a generalization of these results with a sort system that has a fixpoint operator μ.

*email: frederic.ruyer@univ-savoie.fr

†address: Laboratoire de Mathématique, Université de Savoie, 73376 Le Bourget-du-Lac cedex (FRANCE)

2 Polymorphic higher-order logic (pHOL)

The grammars are :

– Sorts :

$$\mathcal{S} = o \mid \mathcal{V} \mid \mathcal{S} \to \mathcal{S} \mid \forall \mathcal{V}\mathcal{S}$$

with $\mathcal{V}$ a denumerable set of sort variables.

– Terms :

$$\Lambda = \Lambda_{const} \mid \Lambda_{var} \mid (\Lambda)\Lambda \mid \lambda\Lambda_{var}.\Lambda$$

with Λ_{var} a denumerable set of sorted term variables, and Λ_{const} a set of sorted constants,that is term variables and constants decorated with a sort.

We suppose given these constants : $\Rightarrow^{o\to(o\to o)}$ and for each sort s, $\forall^{(s\to o)\to o}$. $(\forall)\lambda x.F$ will be written $\forall x.F$.

The rules of construction are :

– Sort rules : (A context C is a multi-set of sort declarations $x : s$)

axiom_var $\dfrac{}{C \vdash' x : s}$ If $x : s \in C$	axiom_const $\dfrac{}{C \vdash' c : s(c)}$ for c a sorted constant of sort s
$\to_i \dfrac{C \vdash' t : s}{C - \{x : s'\} \vdash' \lambda x.t : s' \to s}$	$\to_e \dfrac{C \vdash' u : s' \to s \quad C \vdash' v : s'}{C \vdash' (u)v : s}$
$\forall_i \dfrac{C \vdash' t : s}{C \vdash' t : \forall as}$ a not free in s for $x : s \in C$	$\forall_e \dfrac{C \vdash' t : \forall as(a)}{C- \vdash' t : s(b)}$

– Proof rules :(a context Γ is a multi-set of terms of sort o)

axiom $\dfrac{}{\Gamma \vdash A}$ with $A \in \Gamma$	
$\Rightarrow_i \dfrac{\Gamma \vdash B}{\Gamma - \{A\} \vdash A \Rightarrow B}$	$\Rightarrow_e \dfrac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B}$
$\forall_i \dfrac{\Gamma \vdash A}{\Gamma \vdash \forall^s x A}$ with x not free in Γ	$\forall_e \dfrac{\Gamma \vdash \forall^s x A \quad \Delta \vdash' v : s}{\Gamma \vdash A(v)}$ with $\Delta =$ free variables of Γ
$\beta \dfrac{\Gamma \vdash A}{\Gamma \vdash B}$ with $A =_\beta B$	

3 Exhibiting Russel's paradox

We consider the sort s defined by : $s = \forall a((f(a) \to a) \to a)$ with $f(a) = (a \to o) \to o$.

Lemma 1 *s is a fixpoint of f in the following sense : there exists t_1 and t_2 so that $\vdash' t_1 : f(s) \to s$ and $\vdash' t_2 : s \to f(s)$.*

proof We take :
- $\nearrow = \lambda u.\lambda v.\lambda w.(v)\lambda x.(w)(u)x$
- $t_1 = \lambda a.\lambda b.(b)((\nearrow)\lambda s.(s)b)a$
- $t_2 = \lambda s.(s)(\nearrow)t_1$

They come from the usual proof of Tarski's theorem, that is we have : $\vdash' \nearrow : \forall a \forall b((a \to b) \to (f(a) \to f(b)))$, and t_1 and t_2 have the desired property. ■

Unfortunately, those two terms do not enjoy the property that $t_1 o t_2 = \lambda x.x$. Nevertheless, the following lemma states what can be seen as their "caracteristic property" in the sense that they have a good behavior on the application of predicates of predicates on predicates :

Lemma 2 (Caracteristic property) *We have*

$$\vdash \forall G^{(s\to o)\to o}\forall P^{s\to o}(((t_2)(t_1)G)P \Leftrightarrow (G)\lambda z.(P)(t_1)(t_2)z)$$

<u>proof</u> It is only a β-reduction :

$$\begin{array}{lcl} ((t_2)\ (t_1)G)\ P & = & ((\lambda s.(s)\ (\nearrow)t_1)\ (t_1)G)\ P \\ & \succ_\beta & (((t_1)\ G)\ (\nearrow)t_1)\ P \\ & = & (((\lambda a.\lambda b.(b)\ ((\nearrow)\lambda s.(s)b)a)\ G)\ (\nearrow)t_1)\ P \\ & \succ_\beta & (((\nearrow)t_1)\ ((\nearrow)\lambda s.(s)(\nearrow)t_1)G)\ P \\ & = & (((\lambda u.\lambda v.\lambda w.(v)\lambda x.(w)(u)x)\ t_1)\ ((\nearrow)\lambda s.(s)(\nearrow)t_1)G)\ P \\ & \succ_\beta & (v)\ \lambda x.(w)(u)x \end{array}$$

■

3.1 Inconsistency of FHOL+EXT+PROP

The proofs of this section have been checked on the PhoX [4] proof assistant [...], using the axiom of the characteristic property of the terms t_1 and t_2, EXT and PROP.

We define the formulas :

– EXT=$\forall f \forall g(\forall x((f)x = (g)x) \Rightarrow (f = g))$.

– PROP=$\forall A \forall B((A \Leftrightarrow B) \Rightarrow (A = B))$

The predicate $\in$ and the comprehension schema are defined as follows :

Definition 1 *$x \in y$ is $((t_2)y)\{x\}$ and SCP is $(t_1)\lambda X.\exists v(\{v\} \subset X) \wedge Pv)$.*

With $\{x\}$ is $\lambda z.(z = x)$ and $A \subset B$ is $\forall z(Az \Rightarrow Bz)$.

We prove the caracterisation of SC :

Lemma 3 $\vdash \forall x\, \forall P\, ((x \in \mathrm{SC}\, P) \Leftrightarrow \exists v\, (\forall z{=}v\; t_1\, (t_2\, z) = x \wedge\; P\, v))$

We define the predicate Set as the smaller predicate P such that if $X \subset P$ then $P(SCX)$.

Definition 2 Set $x = \forall P\, (\forall X\, (\forall u{:}X\; P\, u \Rightarrow\; P\, (\mathrm{SC}\, X)) \;\Rightarrow\; P\, x)$.

We firstly show an introduction rule for the predicate Set :

Lemma 4 $\vdash \forall P \, \text{Set}\,(\text{SC}\,\lambda x(P\,x \wedge \text{Set}\,x))$

proof Trivial. ■
We show that $t_1 \circ t_2$ is the identity on Set :

Lemma 5 $\vdash \forall x{:}\text{Set}\; t_1\,(t_2\,x) = x$

proof We assume that Set x and $\forall u{:}X\; t_1\,(t_2\,u) = u$, and we want $t_1\,(t_2\,(\text{SC}\,X)) = \text{SC}\,X$. It is sufficient to prove that :

$$t_2\,(t_1\,\lambda X_0(\exists v\,(\forall z{=}v\; X_0\,z \wedge\; X\,v))) = \lambda X_0 \exists v\,(\forall z{=}v\; X_0\,z \wedge\; X\,v)$$

Using EXT, we have to show for all P :

$$t_2\,(t_1\,\lambda X_0(\exists v\,(\forall z{=}v\; X_0\,z \wedge\; X\,v)))\;P = \exists v\,(\forall z{=}v\; P\,z \wedge\; X\,v)$$

With the Caracteristic Property, the goal becomes :

$$(\exists v\,(\forall z{=}v\; P\,(t_1\,(t_2\,z)) \;\wedge\; X\,v)) = \exists v\,(\forall z{=}v\; P\,z \wedge\; X\,v)$$

Using PROP, we have to show

$$\exists v\,(\forall z{=}v\; P\,(t_1\,(t_2\,z)) \;\wedge\; X\,v) \;\Leftrightarrow\; \exists v\,(\forall z{=}v\; P\,z \wedge\; X\,v)$$

This is trivial by the induction hypothesis $\forall u{:}X\; t_1\,(t_2\,u) = u$. ■
Let's show the theorem that will allow us to conclude

Theorema 1 $\vdash \forall x{:}\text{Set}\; \forall P\; (\forall y{:}P\; \text{Set}\; y \Rightarrow\; P\,x \Leftrightarrow\; x \in \text{SC}\,P)$

proof It is an application of Lemma 3 and Lemma 5. ■
Now the proof of the contradiction :

Theorema 2 $\vdash \bot$

proof We use Lemma 4 with $P = \lambda x(\neg\,(x \in x))$, and then Theorem 1 with $\text{SC}\,\lambda x(P\,x \wedge \text{Set}\,x)$. The end of the proof is as usual. ■

3.2 Inconsistency of FHOL

We show now that the rules PROP and EXT are not necessary to get the result. For this, we make a translation of the derivation by defining new constants and teir properties.

3.2.1 Equalities and extensionnalities

Let E be a finite, non empty, set of sorts. We define the predicates $=_s^E$ and ext_s^E by induction on the sort s.

- $P =_o^E Q$ is $P \Leftrightarrow Q$.
- $x =_s^E y$ is not necessiraly defined if s is atomic.
- $ext_s^E x$ is $x =_s^E x$.
- $f =_{s \to s'}^E g$ is $\forall y, z(ext_s^E y \Rightarrow ext_s^E z \Rightarrow y =_s^E z \Rightarrow fy =_{s'}^E gz)$.
- $x =_{\forall a s(a)}^E y$ is

$$(\bigwedge_{e \in E} \forall X(x(=_{s(a)} [=_a := X])y))$$

Remark : note that the rule for $\forall$ is designed to avoid the intuitively obvious one that would be allowed in system U, and restrict ourselves to the eliminations of $\forall$ used in the sort derivations of the proof!

We define the translation $*E$ of a term by :

- $t^{*E} = t$ if t is a variable or a constant $\neq \forall$.
- $\forall^{*E} = \lambda F.(\forall)\lambda x.(ext^E x \Rightarrow Fx)$.
- $((u)v)^{*E} = (u^{*E})v^{*E}$.
- $(\lambda x.t)^{*E} = \lambda x.t^{*E}$.

The goal is to show :

Theorema 3 *For all term t of sort s, there exists a set E so that for all finite set of sorts E', if $E \subset E'$ then $\Delta_{E'} \vdash ext_s^{E'} t^{*E'}$, where $\Delta_{E'}$ is $\{ext_{s_1}^{E'} x_1, ..., ext_{s_n}^{E'} x_n\}$ with $\{x_1, .., x_n\}$ the set of free variables of t.*

We firstly show a lemma :

Lemma 6 *For all sort s, for all sort variable a, a is free in s iff the symbol $=_a$ appears in $=_s$.*

proof By induction on s. ■

Lemma 7 *For all $x_1, ...x_n$:*
If $x_1 : s_1, \ldots, x_n : s_n \vdash' t : s$ then there exists a finite sort of sorts E so that for every $y_1, ...y_n$, $z_1, ...z_n$:

$$ext_{s_i}^E y_i, ext_{s_i}^E z_i, y_i =_{s_i}^E z_i \vdash t^{*E}[x := y] =_s^E t^{*E}[x := z]$$

and for all finite set of sorts E', if $E \subset E'$ then :

$$ext_{s_i}^{E'} y_i, ext_{s_i}^{E'} z_i, y_i =_{s_i}^{E'} z_i \vdash t^{*E'}[x := y] =_s^{E'} t^{*E'}[x := z]$$

proof By induction on the derivation of $x_1 : s_1, \dots, x_n : s_n \vdash' t : s$.

- For the rules axiom, one has only to check $ext_{(s\to o)\to o} \forall^*$ and $ext_{o\to o\to o} \Rightarrow$.
- For $\to_{intro}$, and $\to_{elim}$ it is easy.
- For the rule $\forall_{intro}$, we use Lemma 6.
- For the rule $\forall_{elim}$, we use the induction hypothesis with $E' = E \cup \{b\}$, and then make enough eliminations of $\wedge$ and replace X by $=_b$.

■

Now we can prove the theorem :

proof It is a particular case of Lemma 7. ■

We will also need in the next section a lemma about β conversion :

Lemma 8 *For every t, t' : if $t \to_\beta t'$ then $t^{*E} \to_\beta t'^{*E}$.*

proof By induction on t. The only fact to check is that for all u, v, $(u[x := v])^{*E} = u^{*E}[x := v^{*E}]$ which is shown by induction on u. ■

3.2.2 Removing EXT and PROP

The goal is to prove the result :

Theorema 4 *If EXT,PROP,$\Gamma \vdash A$ then there exists a set E so that $\Delta_E, \Gamma^{*E} \vdash A^{*E}$, where Δ_E is $\{ext^E_{s_1} x_1, ..., ext^E_{s_n} x_n\}$ with $\{x_1, .., x_n\}$ the set of free variables of $\Gamma \cup \{A\}$.*

We must verify the following result :

Lemma 9 *For evey non empty set of sorts E, $\vdash EXT^{*E}$ and $\vdash PROP^{*E}$.*

proof It is just an application and the definition for PROP, and a little reasonning gives the result for EXT. ■

Like in the previous section, we firstly show a lemma :

Lemma 10 *If EXT,PROP,$\Gamma \vdash A$ then there exists a set E so that for all finite set of sorts E', if $E \subset E'$ then :*

$\Delta_{E'}, \Gamma^{*E'} \vdash A^{*E'}$, where $\Delta_{E'}$ is $\{ext^{E'}_{s_1} x_1, ..., ext^{E'}_{s_n} x_n\}$ with $\{x_1, .., x_n\}$ the set of free variables of $\Gamma \cup \{A\}$.

proof By induction on the derivation of EXT,PROP,$\Gamma \vdash A$:

- If the rule is an axiom, it is trivial if A is in Γ, and follows Lemma 9 else.
- If the rule is $\Rightarrow_{intro}$ or $\Rightarrow_{elim}$, it follows the fact that $(A \Rightarrow B)^{*E} = A^{*E} \Rightarrow B^{*E}$.
- If the rule is β, it comes from Lemma 8.
- If the rule is $\forall_{intro}$, it comes from the induction hypothesis and the definition of $\forall^{*E}$.
- The special case is $\forall_{elim}$. By Theorema 3, there is a non empty set F verifying that for all finite set of sorts F', if $F \subset F'$ then $\Phi_{F'} \vdash ext_s^{F'} v^{*F'}$, where $\Phi_{F'}$ is $\{ext_{s_1}^{F'} y_1, ..., ext_{s_n}^{F'} y_n\}$ with $\{y_1, .., y_n\}$ the set of free variables of v.
  Let $G = E \cup F$, and G' so that $G \subset G'$. We have $\Phi_{G'} \vdash ext_s^{G'} v^{*G'}$, and since $\{y_1, .., y_n\} \subset \{x_1, .., x_n\}$, we also have $\Delta_{G'}, \Gamma^{*G'} \vdash ext_s^{G'} v^{*G'}$. By induction hypothesis, we also have that $\Delta_{G'}, \Gamma^{*G'} \vdash ext_s^{G'} \forall^{*G} A^{*G}$. So we get the result.

■

Now, if we apply the theorema to the proof of falsity given before, we have that there is a finite set E so that $\vdash (\forall XX)^{*E}$, and the definitions allow us to get $\vdash (\forall XX)$.

This proof has been checked with the PhoX [4] proof assistant, using for only axioms that t_1 and t_2 are extensionnal(in the sense defined before), and the characteristic property of thes terms. (in the sense defined before).

4 A new type system

4.1 Rules

The grammars are :
- Sorts : $\mathcal{S} = o|\mathcal{V}|\mathcal{S} \rightarrow \mathcal{S}|\exists\mathcal{V}\mathcal{S}$. With $\mathcal{V}$ a denumerable set of sort variables.
- Terms : $\Lambda = \Lambda_{const}|\Lambda_{var}|(\Lambda)\Lambda|\lambda\Lambda_{var}.\Lambda$ .With Λ_{var} a denumerable set of sorted term variables, and Λ_{const} a set of sorted constants,that is term variables and constants decorated with a sort.

We suppose given these constants : $\Rightarrow^{o\rightarrow(o\rightarrow o)}$ and for each sort s, $\forall^{(s\rightarrow o)\rightarrow o}$. The meaning of $(\forall)\lambda x.F$ is $\forall x.F$.

The rules of construction are :
- Sort rules : (A context C is a multi-set of sort declarations $x : s$)

axiom_var $\dfrac{}{C \vdash' x : s}$ If $x : s \in C$	axiom_const $\dfrac{}{C \vdash' c : s(c)}$ for c a sorted constant of sort s
$\rightarrow_i \dfrac{C \vdash' t : s}{C - \{x : s'\} \vdash' \lambda x.t : s' \rightarrow s}$	$\rightarrow_e \dfrac{C \vdash' u : s' \rightarrow s \quad C \vdash' v : s'}{C \vdash' (u)v : s}$
$\exists_i \dfrac{C \vdash' t : s}{C \vdash' t : \exists a s}$	$\exists_e \dfrac{C, x : s(a) \vdash' t : s'}{C, x : \exists a s(a) \vdash' t : s'}$ with a not free in C and s'

axiom_var $\rightarrow_i$ $\rightarrow_e$ $\exists_i$ $\exists_e$

– Proof rules : they are the same as in the preceeding system.

4.2 Adequacy and normalization

We show in this section that the β rule is semantically correct.

4.2.1 Adequacy

We firstly define the interpretation of a sort. It is very similar to the same notion in system F. Let $U \subset \mathcal{P}(\Lambda)$.

Definition 3 *An environment E is a function $\mathcal{V} \rightarrow U$. Let E be an environment $P \in U$, and $x \in \mathcal{V}$. We note $E' = E[x := P]$ the environment E' defined by $E'(y) = E(y)$ if $y \neq x$, and $E'(x) = P$.*

Let P and Q be to parts of Λ. We note $P \rightarrow Q = \{u; \forall x \in P : (u)x \in Q\}$

Definition 4 *We define simultaneously for every environment E the interpretation $\|s\|_E$ of a sort s in the environment E, by :*

- *$\|s\|_E = E(s)$ if $s \in \mathcal{V}$.*
- *$\|a \rightarrow b\|_E$ is $\|a\|_E \rightarrow \|b\|_E$.*
- *$\|\exists a s(a)\|_E$ is*

$$\bigcup_{P \in U} \|s(a)\|_{E[a:=P]}$$

Definition 5 *Let $U \subset \mathcal{P}(\Lambda)$. U is well adapted if :*

- *U is closed by $\rightarrow$.*
- *U is closed by union.*

– *for every terms* $u, t_1, ..., t_n$*, for every* $P, P' \in U$ *: for every* $v \in P$*, if* $(...(u[x := v])t_1...)t_n \in P'$ *then* $(...((\lambda x.u)v)t_1...)t_n \in P'$ *(we say that* U *is saturated).*

Lemma 11 *If* U *is well adapted, for every environment* E*, for every context* $x_1 : s_1, ...x_n : s_n$*, and for every* $t, s, u_1, .., , u_n$ *:*

$$\left.\begin{array}{r} x_1 : s_1, ...x_n : s_n \vdash' t : s \\ u_1 \in \|s_1\|_E, ..., u_n \in \|s_n\|_E \end{array}\right\} \Rightarrow t[x_1 := u_1, ..., x_n := u_n] \in \|s\|_E$$

proof The facct that U is well adapted ensures that the interpretation of a sort will be in U. We make the proof by induction on the derivation.

– rule axiom_var : it is immediate.
– rule $\rightarrow_i$: $t = \lambda y.u$ with $y \notin \{x_1; ....x_n\}$ and $s = a \rightarrow b$. Take $v \in \|a\|_E$. By induction hypothesis we have that $u[y := v, x_1 := u_1, ..., x_n := u_n] \in \|b\|_E$; since U is saturated, $(\lambda y.u[x_1 := u_1, ..., x_n := u_n])v \in \|b\|_E$. So, $t[x_1 := u_1, ..., x_n := u_n] \in \|a \rightarrow b\|_E$.
– rule $\rightarrow_e$: $t = (u)v$, with $u[x_1 := u_1, ..., x_n := u_n] \in \|a \rightarrow b\|_E$ and $v[x_1 := u_1, ..., x_n := u_n] \in \|a\|_E$. by definition, $t[x_1 := u_1, ..., x_n := u_n] \in \|b\|_E$.
– rule $\exists_i$: by induction hypothesis, we have $t[x_1 := u_1, ..., x_n := u_n] \in \|s(a)\|_E$ for a sort a. So $t[x_1 := u_1, ..., x_n := u_n] \in \|s(x)\|_{E[x:=\|s(a)\|_E]} \subset \|\exists a s(a)\|_E$.
– rule $\exists_e$: we suppose that this left rule is applicated on the first hypothesis, that is $x_1 : \exists a s(a)$. Take $u_1 \in \|\exists a s(a)\|_E$. There exists $P \in U$ so that $u_1 \in \|s(a)\|_{E[a:=P]}$. Because a is not free neither in s_i for $i > 1$, nor in s, we have $\|s_i\|_{E[a:=P]} = \|s_i\|_E$ and $\|s\|_{E[a:=P]} = \|s\|_E$. By induction hypothesis with the environment $E' = E[a := P]$, we have then that $t[x_1 := u_1, ..., x_n := u_n] \in \|s\|_E$.

■

Lemma 12 *Let* N *be the set of strongly normalisable terms, and* $N_0 = \{(...(x)t_1...)t_n; x \in \Lambda_{var}, t_i \in N\}$*. Take* U *the set of parts* P *such that :*
– $N_0 \subset P \subset N$*.*
– *for every* $u, t_1, ..., t_n$*, for every* $v \in N$ *if* $(...(u[x := v])t_1...)t_n \in P$ *then* $(...((\lambda x.u)v)t_1...)t_n \in P$*.*

U is well adapted.

proof

– U is closed by $\to$ because $N_0 \to N \subset N$, and $N_0 \subset N \to N_0$, and because of the usual properties of co- and contra- variance of $\to$.
– It is obvious that U is closed by union.
– U is saturated : it comes from it's definition.

■

Theorema 5 *Every typable term is strongly normalizing.*

proof It is a consequence of the adequation lemma and the previous lemma. ■

4.3 Russell's paradox

It is far more easy to derive the paradox in this system.

4.3.1 And and projections

We define the "conjunction" of two sorts a and b, and then a property of "projection".

Definition 6 *We define :* $a \times b = (a \to b \to o) \to o$.

Lemma 13 *For every sort* a *and* b*, there exists a term* $<,>^{a,b}$*, a term* $\pi_1^{a,b}$*, and a term* $\pi_2^{a,b}$ *such that :*
– *For every* $x : a,\ y : b : ((<,>^{a,b})x)y : a \times b$.
– *For every* $c : a \times b : (\pi_1^{a,b})c : (a \to o) \to o$.
– *For every* $c : a \times b : (\pi_2^{a,b})c : (b \to o) \to o$.

proof Take :
– $<,>^{a,b}= \lambda x.\lambda y.\lambda f.((f)x)y$.
– $\pi_1^{a,b} = \lambda c.\lambda g.(c)\lambda x.\lambda y.(g)x$
– $\pi_2^{a,b} = \lambda c.\lambda g.(c)\lambda x.\lambda y.(g)y$

■

We will omit the sorts when the context is clear, and write $< x, y >$ for $((<,>)x)y$.

Lemma 14 *For every x,y,x',y' : if* $\vdash < x, y > = < x', y' >$ *then* $\vdash x = x'$ *and* $\vdash y = y'$.

proof From $\vdash < x, y > = < x', y' >$ we deduce that $\vdash \pi_1 < x, y > = \pi_1 < x', y' >$. When we reduce the terms, we get : $\vdash \lambda g.(g)x = \lambda g.(g)x'$, and then $\vdash (\lambda g.(g)x)\{x\} = (\lambda g.(g)x')\{x\}$. The conclusion comes from the usual properties of Leibnitz's equality.the proof is the same for y and y'. ■

4.3.2 Obtaining the paradox

Definition 7 *We define* $i = \exists e((e \to f(e) \times e)$

Lemma 15 *i is a fixpoint of f in the following sense : there exists t_1 and t_2 so that $\vdash' t_1 : i \to f(i)$ and $\vdash' t_2 : f(i) \to i$.*

proof $t_1 = \lambda i.\lambda j.(j)i$ is convenient. We take $\nearrow$ the same term as in the first section (not typed with "forall", but with the sorts i and $f(i)$), and we have $t_2 = \lambda x. < (\nearrow)t_1, x >$ is convenient. ■

Definition 8 *We define the following terms :*
- $P \in Q = (Q)(t_2)\{P\}$.
- $CS(\Phi) = \lambda x.\exists u((x = (t_2)\{u\}) \wedge \Phi u)$

Lemma 16 $\vdash \forall P, \Phi(P \in CS(\Phi) \Leftrightarrow \Phi P)$.

proof straightforward, using π_2. ■

Theorema 6 *The system R is inconsistent.*

proof As usual ■

5 Recursive types

5.1 Rules

We follow the presentation given in (amadio,cardelli)

The grammars are :
- Sorts : $\mathcal{S} = o|\mathcal{V}|\mathcal{S} \to \mathcal{S}|\mu\mathcal{V}\mathcal{S}$. With $\mathcal{V}$ a denumerable set of sort variables.

– Terms : $\Lambda = \Lambda_{const}|\Lambda_{var}|(\Lambda)\Lambda|\lambda\Lambda_{var}.\Lambda$ .With Λ_{var} a denumerable set of sorted term variables, and Λ_{const} a set of sorted constants,that is term variables and constants decorated with a sort.

We suppose given these constants : $\Rightarrow^{o\to(o\to o)}$, for each sort s, $\forall^{(s\to o)\to o}$, and for every sort s, $\mathtt{fold}^{s[t:=\mu t.s]\to\mu t.s}$,and $\mathtt{unfold}^{\mu t.s\to s[t:=\mu t.s]}$. The meaning of $(\forall)\lambda x.F$ is $\forall x.F$.

The rules of construction are :

– Sort rules : (A context C is a multi-set of sort declarations $x : s$)

axiom_var $\dfrac{}{C \vdash' x : s}$ If $x : s \in C$	axiom_const $\dfrac{}{C \vdash' c : s(c)}$ for c a sorted constant of sort s
$\to_i \dfrac{C \vdash' t : s}{C - \{x : s'\} \vdash' \lambda x.t : s' \to s}$	$\to_e \dfrac{C \vdash' u : s' \to s \quad C \vdash' v : s'}{C \vdash' (u)v : s}$

– Proof rules : they are the same as the previous ones, and we add two rules :

fold-unfold $\dfrac{\Gamma \vdash A(x) \quad \Delta \vdash' x : \mu t.s}{\Gamma \vdash A((\mathtt{fold})(\mathtt{unfold})x)}$ Δ free variables of Γ
unfold-fold $\dfrac{\Gamma \vdash A(x) \quad \Delta \vdash' x : s[t := \mu t.s]}{\Gamma \vdash A((\mathtt{unfold})(\mathtt{fold})x)}$ Δ free variables of Γ

5.2 Russel's paradox

Take $s = \mu t(t \to o \to o)$.

Definition 9 *We define :*

– $x \in y$ *is* $((\mathtt{unfold})y)\{x\}$
– $SC(P)$ *is* $(\mathtt{fold})\lambda X.\exists u(X = \{u\} \wedge Pu)$

Theorema 7 *the system is inconsistent.*

proof Just check that $\vdash \forall x, P(x \in SC(P) \Leftrightarrow Px)$, and then obtain the paradox. ■

6 Conclusion

This paper presents a new paradoxes in type theory, always based on the same idea. It would be interesting to study the behaviour of the terms that comes from these proofs. The link with the work of Coquand [2], excepted the sort we use, is unclear for the authors, and is left open.
We would like now to study an ML-like type system, using restricted rules on μ and $\exists$, that preserves consistency, and define more precisely the boarder that separates the correct systems from the incorrect ones.

[1]Thierry Coquand ."A new paradox in type theory".In 9th International Conference on Logic, Methodology, and Philosophy of Science, pages 555–570, 1994. [2]Thierry Coquand. "An analysis of Girard's paradox."Proceedings of the first Logic in Computer Science, Boston, 1986, pp.227-236. [3]Jean-Yves Girard . "Interprtation fonctionnelle et limination des coupures de l'arithmtique d'ordre suprieur." Thse d'Etat, Paris VII (1972).

Références

[1] Thierry Coquand. "An analysis of Girard's paradox."Proceedings of the first Logic in Computer Science, Boston, 1986, pp.227-236.

[2] Thierry Coquand ."A new paradox in type theory".In 9th International Conference on Logic, Methodology, and Philosophy of Science, pages 555–570, 1994.

[3] Jean-Yves Girard . "Interprtation fonctionnelle et limination des coupures de l'arithmtique d'ordre suprieur." Thse d'Etat, Paris VII (1972).

[4] C.Raffalli.The PhoX proof assistant.
http ://www.lama.univ-savoie.fr/sitelama/Membres/pages_web/RAFFALLI/af2.html

[5] Reynolds J.C. "Polymorphism is not set-theoretic."Semantics of Data Types, edited by G.Kahn,D.B.MacQueen, and G.D. Plotkin, LNCS 173, Springer-Verlag, 1984, pp.145-146.