
Paradoxes in Type Theory

— *PLClub — 4 April 2025* —

What is a paradox?

in set theory + logic:

Russell's paradox:

$$R := \{x \mid x \notin x\}$$

$$R \in R \Leftrightarrow R \notin R$$

contradiction!

in type theory:

closed proof of false:

$$\cdot \vdash b : \perp$$

all closed types are inhabited:

$$\forall A. \exists a. \cdot \vdash a : A$$

$$\Leftrightarrow \forall A. \cdot \vdash \text{absurd}(b) : A$$

Topics:

1. **Which** type-theoretic features together cause a paradox?
2. **How** do you construct a paradox from these features?

① Ingredients for a paradox

dependent functions, plus one of:

- type-in-type
- two impredicative universes
- strong impredicative pairs
- strong elimination + large inductives
- negative inductive types
- typecase + parametrized inductive types
- proof irrelevance + strong elimination
- substitution + dependent elimination + observable effects
- nonstrictly-positive inductive types + impredicativity

This list is [incomplete](#); you can help by [expanding it](#).

Type-in-type

$\Gamma \vdash A : \text{Type}$

$\Gamma, x: A \vdash B : \text{Type}$

$\Gamma \vdash \Pi x: A. B : \text{Type}$

$\Gamma, x: A \vdash b : B$

$\Gamma \vdash \lambda x. b : \Pi x: A. B$

$\Gamma \vdash b : \Pi x: A. B$

$\Gamma \vdash a : A$

$\Gamma \vdash b a : B\{a/x\}$

 $\Gamma \vdash \text{Type} : \text{Type}$

A Theory of Types (Martin-Löf 1971)

2.7.1

$V \in V$

This is the basic axiom with which every derivation begins. There are no other axioms.

2.7.2 If we have a (possibly open) derivation of $A \in V$, then we may introduce a new variable x and use

$x \in A$

as assumption. The type symbol A is to be uniquely associated with x , that is, having made the assumption $x \in A$, we are not allowed to use $x \in B$ as assumption unless B is identical with A .

2.7.3

$$\prod\text{rule} \quad \frac{A \in V \quad \begin{array}{c} [x \in A] \\ B(x) \in V \end{array}}{(\prod x \in A) B(x) \in V}$$

This rule of inference is subjected to the restriction that the variable x must not occur in any assumption of the derivation of $B(x) \in V$ other than the indicated $x \in A$. Note that, when passing

from $B(x) \in V$ to $(\prod x \in A) B(x) \in V$, all assumptions of the form $x \in A$ are cancelled, but, instead, the conclusion becomes dependent on the assumptions of the derivation of $A \in V$ which allowed us to use $x \in A$ as assumption.

2.7.4

$$\lambda\text{rule} \quad \frac{A \in V \quad \begin{array}{c} [x \in A] \\ b(x) \in B(x) \end{array}}{(\lambda x \in A) b(x) \in (\prod x \in A) B(x)}$$

Similar remarks as for the previous rule of inference.

2.7.5

$$\text{application rule} \quad \frac{b \in (\prod x \in A) B(x) \quad a \in A}{ba \in B(a)}$$

Here, of course, $B(a)$ denotes the result of substituting a for all free occurrences of x in $B(x)$.

2.7.6

$$\text{equality rule} \quad \frac{a \in A \quad B \in V}{a \in B} \quad \text{if } A = B$$

An application of the equality rule with A and B identical is redundant and can be removed. Also, whenever convenient, we can assume that there are no two successive applications of the equality rule, because, clearly, two such applications can be made into one application of the same rule.

System U (Girard 1972, Coquand 1986, Hurkens 1995)

$\Gamma \vdash A : * / \square / \triangleleft$

$\Gamma, x : A \vdash B : *$

 $\Gamma \vdash * : \square$

 $\Gamma \vdash \Pi x : A. B : *$

$\Gamma \vdash A : * / \square / \triangleleft$

$\Gamma, x : A \vdash B : \square$

 $\Gamma \vdash \square : \triangle$

 $\Gamma \vdash \Pi x : A. B : \square$

impredicativity:
this universe
may be bigger than
this universe

System U (Girard 1972, Coquand 1986, Hurkens 1995)

$$\Gamma \vdash A : \Delta$$

$$\Gamma, x: A \vdash B : \square$$

 $\Gamma \vdash \square : \Delta$

 $\Gamma \vdash \Pi x: A. B : \square$

to show the paradox,

this universe must match *this* universe
 (I will not explain)

$$\vdash \underbrace{\Pi x: \square}_{\Delta}. \underbrace{(((X \rightarrow *) \rightarrow *) \rightarrow X) \rightarrow ((X \rightarrow *) \rightarrow *)}_{\square} : \square$$

Strong impredicative pairs

$$\Gamma \vdash p : \Sigma x: A. B$$

$$\Gamma \vdash \text{fst } p : A$$

strong
(has projections)

$$\Gamma \vdash A : */\square/\triangle$$
$$\Gamma, x: A \vdash B : \square$$

$$\Gamma \vdash \Sigma x: A. B : \square$$

impredicative
(has larger component)

$$\Gamma \vdash a : A$$
$$\Gamma \vdash b : B\{a/x\}$$

$$\Gamma \vdash \langle a, b \rangle : \Sigma x: A. B$$

pair
(pair)

Strong impredicative pairs boxes

$$\Gamma \vdash b : \text{Box } A$$

$$\Gamma \vdash \text{unbox } b : A$$

strong
(has projection)

$$\Gamma \vdash A : * / \square / \triangle$$

$$\Gamma \vdash \text{Box } A : \square$$

impredicative
(has larger component)

$$\Gamma \vdash a : A$$

$$\Gamma \vdash \langle a \rangle : \text{Box } A$$

box
(box)

Strong impredicative pairs boxes

$$\Gamma \vdash b : \text{Box } A$$
$$\Gamma \vdash A : * / \square / \triangle$$
$$\Gamma \vdash a : A$$
$$\text{-----}$$
$$\Gamma \vdash \text{unbox } b : A$$
$$\text{-----}$$
$$\Gamma \vdash \text{Box } A : \square$$
$$\text{-----}$$
$$\Gamma \vdash \langle a \rangle : \text{Box } A$$

a larger universe can “hide inside” a smaller universe

$$\vdash \text{Box } (\underbrace{\Pi X : \square. (((X \rightarrow *) \rightarrow *) \rightarrow X) \rightarrow ((X \rightarrow *) \rightarrow *)}_{\triangle}) : \square$$

Strong elimination + large inductive

data Box (A : \triangle) : \square where
box : A \rightarrow Box A

unbox : Box A \rightarrow A
unbox (box a) = a

large inductive
(type of A bigger than type of Box A)

strong elimination
(univ. of return type bigger than
univ. of type of argument)

\vdash Box ($\prod X$: \square . $((X \rightarrow *) \rightarrow *) \rightarrow X$) \rightarrow $((X \rightarrow *) \rightarrow *)$) : \square

$\underbrace{\hspace{15em}}$
 \triangle

② Recipe for a paradox

1. Define a “paradoxical” type U
2. Define a notion of well-foundedness on U
3. Show that U is well founded
4. Show that U is not well founded
5. Contradiction!

Hurkens' (simplification of Girard's) paradox (1995)

We consider the following universe:

$$\mathcal{U} \equiv \Pi \mathcal{X} : \square. ((\wp \wp \mathcal{X} \rightarrow \mathcal{X}) \rightarrow \wp \wp \mathcal{X})$$

$$\vdash \Pi X : \square. (((X \rightarrow *) \rightarrow *) \rightarrow X) \rightarrow ((X \rightarrow *) \rightarrow *) : \square$$

For each term s of type \mathcal{U} , we define a term of type $\wp \wp \mathcal{U}$:

$$\sigma s \equiv (\{s \ \mathcal{U}\} \ \lambda t : \wp \wp \mathcal{U}. \ \tau t)$$

(So we do not consider σ and τ as *terms*.)

We define normal terms of type $\wp \mathcal{U}$ and \mathcal{U} , respectively:

$$\Delta \equiv \lambda y : \mathcal{U}. \ \neg \forall p : \wp \mathcal{U}. \ [(\sigma y \ p) \Rightarrow (p \ \tau \sigma y)]$$

$$\Omega \equiv \text{the normal form of } \tau \ \lambda p : \wp \mathcal{U}. \ \forall x : \mathcal{U}. \ [(\sigma x \ p) \Rightarrow (p \ x)]$$

In other words, $\Omega \equiv \Lambda \mathcal{X} : \square. \ \lambda f : (\wp \wp \mathcal{X} \rightarrow \mathcal{X}). \ \lambda p : \wp \mathcal{X}. \ \forall x : \mathcal{U}. \ [(\sigma x \ \lambda y : \mathcal{U}. \ (p \ (f \ (\{y \ \mathcal{X}\} \ f)))) \Rightarrow (p \ (f \ (\{x \ \mathcal{X}\} \ f)))]$.

We claim that the following is a term of type \perp in $\lambda \mathcal{U}^-$:

$$\begin{aligned} & [\text{suppose } 0 : \forall p : \wp \mathcal{U}. \ [\forall x : \mathcal{U}. \ [(\sigma x \ p) \Rightarrow (p \ x)]] \Rightarrow (p \ \Omega)]. \\ & [[\langle 0 \ \Delta \rangle \text{ let } x : \mathcal{U}. \ \text{suppose } 2 : (\sigma x \ \Delta). \ \text{suppose } 3 : \forall p : \wp \mathcal{U}. \ [(\sigma x \ p) \Rightarrow (p \ \tau \sigma x)]. \\ & \quad [[\langle 3 \ \Delta \rangle \ 2] \text{ let } p : \wp \mathcal{U}. \ \langle 3 \ \lambda y : \mathcal{U}. \ (p \ \tau \sigma y) \rangle] \text{ let } p : \wp \mathcal{U}. \ \langle 0 \ \lambda y : \mathcal{U}. \ (p \ \tau \sigma y) \rangle] \\ & \quad \text{let } p : \wp \mathcal{U}. \ \text{suppose } 1 : \forall x : \mathcal{U}. \ [(\sigma x \ p) \Rightarrow (p \ x)]. \ [\langle 1 \ \Omega \rangle \text{ let } x : \mathcal{U}. \ \langle 1 \ \tau \sigma x \rangle] \end{aligned}$$

Hurkens' (simplification of Girard's) paradox (1995)

We consider the following universe:

$$\mathcal{U} \equiv \Pi \mathcal{X} : \square. ((\rho \rho \mathcal{X} \rightarrow \mathcal{X}) \rightarrow \rho \rho \mathcal{X})$$

For each term t of type $\rho \rho \mathcal{U}$, we define a term of type \mathcal{U} :

$$\tau t \equiv \Lambda \mathcal{X} : \square. \lambda f : (\rho \rho \mathcal{X} \rightarrow \mathcal{X}). \lambda p : \rho \mathcal{X}. (t \lambda x : \mathcal{U}. (p (f (\{x \ \mathcal{X}\} f))))$$

Definition $P (S : \text{Type}) : \text{Type} := S \rightarrow \text{Type}$.

Definition $U : \text{Type} := \text{forall } (x : \text{Type}), ((P (P x)) \rightarrow x) \rightarrow (P (P x))$.

Definition $\text{tau } (t : P (P U)) : U := \text{fun } x \text{ f p} \Rightarrow t (\text{fun } s \Rightarrow p (f (s \ x \ f)))$.

Definition $\text{sigma } (s : U) : P (P U) := s \ U \ \text{tau}$.

Definition $\text{Delta } (y : U) : \text{Type} := (\text{forall } (p : P U), \text{sigma } y \ p \rightarrow p (\text{tau } (\text{sigma } y))) \rightarrow \text{False}$.

Definition $\text{Omega} : U := \text{tau } (\text{fun } p \Rightarrow \text{forall } (x : U), \text{sigma } x \ p \rightarrow p \ x)$.

Definition $M (x : U) (\text{two} : \text{sigma } x \ \text{Delta}) : \text{Delta } x :=$

$\text{fun } \text{three} \Rightarrow \text{three } \text{Delta } \text{two } (\text{fun } p \Rightarrow \text{three } (\text{fun } y \Rightarrow p (\text{tau } (\text{sigma } y))))$.

Fail **Definition** $R (p : P U) (\text{one} : \text{forall } x, \text{sigma } x \ p \rightarrow p \ x) : p \ \text{Omega} :=$

$\text{one } \text{Omega } (\text{fun } x \Rightarrow \text{one } (\text{tau } (\text{sigma } x)))$.

Fail **Definition** $L (\text{zero} : \text{forall } p, (\text{forall } x, \text{sigma } x \ p \rightarrow p \ x) \rightarrow p \ \text{Omega}) : \text{False} :=$

$\text{zero } \text{Delta } M (\text{fun } p \Rightarrow \text{zero } (\text{fun } y \Rightarrow p (\text{tau } (\text{sigma } y))))$.

[suppose $0 : \forall p : \rho \mathcal{U}. [\forall x : \mathcal{U}. [(sx \ p) \Rightarrow (p \ \Omega)]]$.
 $[[\langle 0 \ \Delta \rangle \text{ let } x : \mathcal{U}. \text{suppose } 2 : (sx \ \Delta). \text{suppose } 3 : \forall p : \rho \mathcal{U}. [(sx \ p) \Rightarrow (p \ \tau sx)]]$.
 $[[\langle 3 \ \Delta \rangle 2] \text{ let } p : \rho \mathcal{U}. \langle 3 \ \lambda y : \mathcal{U}. (p \ \tau sy) \rangle]]$ let $p : \rho \mathcal{U}. \langle 0 \ \lambda y : \mathcal{U}. (p \ \tau sy) \rangle$
 let $p : \rho \mathcal{U}. \text{suppose } 1 : \forall x : \mathcal{U}. [(sx \ p) \Rightarrow (p \ x)]. [\langle 1 \ \Omega \rangle \text{ let } x : \mathcal{U}. \langle 1 \ \tau sx \rangle]]$

The paradox of trees (Coquand 1992)

ingredients:

- [original formulation] type-in-type + W types (“trees”) + equality; or
- **type-in-type + strong elimination + indexed inductives**

The paradox of trees (Coquand 1992)

data U : Type where

data Acc : U → Type where

wfu : (x : U) → Acc x

nwf : Acc (. . . .) → ⊥

nwf (wfu (. . . .)) : ⊥

The paradox of trees (Coquand 1992)

data U : Type where

u : (X : Type) → (X → U) → U

X many branches of U

the “limit” of all branches of U

leaf : U

leaf = u ⊥ absurd

The paradox of trees (Coquand 1992)

```
data U : Type where
  u : (X : Type) → (X → U) → U
```

X many branches of U

the “limit” of all branches of U
compare:

```
leaf : U
leaf = u ⊥ absurd
```

```
data Ord : Type where
  zero  : Ord
  succ  : Ord → Ord
```

the “limit” of the Nat-indexed ordinals { limit : (Nat → Ord) → Ord

The paradox of trees (Coquand 1992)

data U : Type where

u : (X : Type) → (X → U) → U

X many branches of U

the “limit” of all branches of U



lem : $\forall X, f, x. f\ x < u\ X\ f$

lem : $\forall f, n. f\ n < \text{limit}\ X\ f$

leaf : U

leaf = u \perp absurd

data Ord : Type where

zero : Ord

succ : Ord → Ord

limit : (Nat → Ord) → Ord

The paradox of trees (Coquand 1992)

data U : Type where

u : (X : Type) → (X → U) → U

data Acc : U → Type₂ where

acc : (X : Type) → (f : X → U) →
 ((x : X) → Acc (f x)) → Acc (u X f)

if all branches of U are accessible, then the whole U is accessible

The paradox of trees (Coquand 1992)

data U : Type where

u : (X : Type) → (X → U) → U

data Acc : U → Type₂ where

acc : (X : Type) → (f : X → U) →
((x : X) → Acc (f x)) → Acc (u X f)

induction hypothesis

subgoal

goal

elimU : $\forall P. (\forall X f. (\forall x. P (f x)) \rightarrow P (u X f)) \rightarrow \forall u. P u$

The paradox of trees (Coquand 1992)

data U : Type where

u : (X : Type) → (X → U) → U

data Acc : U → Type where

acc : (X : Type) → (f : X → U) →
((x : X) → Acc (f x)) → Acc (u X f)

wfu : (x : U) → Acc x

wfu (u X f) = acc X f (λx. wfu (f x))

TFAE:

= “all U are accessible”

= “U eventually has a base case”

= “we can do induction on U”

The paradox of trees (Coquand 1992)

data U : Type where

u : (X : Type) → (X → U) → U

data Acc : U → Type where

acc : (X : Type) → (f : X → U) →
((x : X) → Acc (f x)) → Acc (u X f)

wfu : (x : U) → Acc x

wfu (u X f) = acc X f (λx. wfu (f x))

nwf : Acc (u U id) → ⊥

nwf (acc .U .id h) =

h : (x : U) → Acc (id x)
h (u U id) : Acc (u U id)

The paradox of trees (Coquand 1992)

```
data U : Type where
```

```
  u : (X : Type) → (X → U) → U
```

```
data Acc : U → Type where
```

```
  acc : (X : Type) → (f : X → U) →  
        ((x : X) → Acc (f x)) → Acc (u X f)
```

```
wfu : (x : U) → Acc x
```

```
wfu (u X f) = acc X f (λx. wfu (f x))
```

```
nwf : Acc (u U id) → ⊥
```

```
nwf (acc .U .id h) = nwf (h (u U id))
```

```
nwf (wfu (u U id)) : ⊥
```

The paradox of trees (Coquand 1992)

```
data Ord : Type where
  succ : Ord → Ord
  limit : (X : Type) → (X → Ord) → Ord

data · < · : Ord → Ord where
  ...
```

```
data Acc : Ord where
  acc : (n : Ord) → ((m : Ord) → m < n → Acc m) → Acc n
```

```
∞ : Ord          loop : ∞ < ∞    acc∞ : Acc ∞ → ⊥
∞ = limit Ord id loop = ...    acc∞ (acc .∞ h) = h ∞ loop
```

Ingredients for a paradox

dependent functions, plus one of:

- type-in-type
- two impredicative universes
- strong impredicative pairs
- strong elimination + large impredicative inductives
- negative inductive types
- typecase + parametrized inductive types
- proof irrelevance + strong elimination
- substitution + dependent elimination + observable effects
- nonstrictly-positive inductive types + impredicativity

This list is [incomplete](#); you can help by [expanding it](#).

Negative inductive types

data Bad : Type where

bad : (Bad → ⊥) → Bad

badless : Bad → ⊥

badless (bad nb) = nb (bad nb)

false : ⊥

false = badless (bad badless)

negative occurrence of Bad

case analysis on Bad

looping term

Typecase + parametrized inductives

```
data Bad (F : Type → Unit) : Type
```

negative occurrence of Type

```
typeful : Type → Unit
```

```
typeful (Bad F) = F (Bad F)
```

case analysis on Type

```
typeful _ = unit
```

```
loop : Unit
```

```
loop = typeful (Bad typeful)
```

looping term

Proof irrelevance + strong elimination

```
pirrel : (P : Prop) → (p q : P) → p ≡ q
```

```
data PBool : Prop where  
  ptrue pfalse : PBool
```

```
lift : PBool → Prop
```

strong elimination of PBool

```
lift ptrue = T
```

```
lift pfalse = ⊥
```

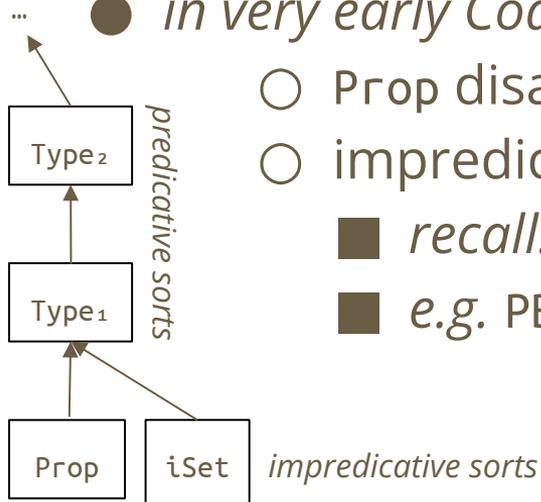
```
discr : ptrue ≡ pfalse → ⊥
```

```
discr p q e = coerce unit (congr lift e : T ≡ ⊥)
```

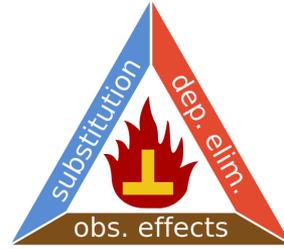
Berardi's paradox (Barbanera and Berardi 1996)

- [original formulation] LEM + AC \Rightarrow proof irrelevance
- [modern formulation] LEM + impred. pairs \Rightarrow proof irrelevance
- proof irrelevance + strong elimination \Rightarrow paradox
- *in very early Coq:*

- Prop disallows strong elimination to allow LEM, AC, etc.
- impredicative Set allows strong elim. of small inductives
 - *recall:* strong elim. of large inductives \Rightarrow paradox
 - e.g. PBool is small; U is large



The Fire Triangle (Pédrot and Tabareau 2020)



Definition 3 (Observable Effects). A type theory is *observably effectful* if there exists a closed term $\vdash t : \mathbb{B}$ that is not observationally equivalent to a value, that is, there exists a context C such that $C[\text{true}] \equiv \text{true}$ and $C[\text{false}] \equiv \text{true}$, but $C[t] \equiv \text{false}$ (where \equiv denotes definitional equality).

$\Gamma \vdash \text{refl} : \text{true} = C[\text{true}]$

$\Gamma \vdash \text{refl} : \text{true} = C[\text{false}]$

 $\Gamma \vdash \text{if } t \text{ then refl else refl} : \text{true} = C[t] \quad \Gamma \vdash C[t] \equiv \text{false}$

 $\Gamma \vdash \text{if } t \text{ then refl else refl} : \text{true} = \text{false}$

Ingredients for a paradox

dependent functions, plus one of:

- type-in-type
- two impredicative universes
- strong impredicative pairs
- strong elimination + large impredicative inductives
- negative inductive types
- typecase + parametrized inductive types
- proof irrelevance + strong elimination
- substitution + dependent elimination + observable effects
- nonstrictly-positive inductive types + impredicativity ? ? ?

“Reynold’s paradox” (*A New Paradox in Type Theory*, Coquand 1994)

References

- [Girard 1972] Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur
- [Coquand 1986] An Analysis of Girard's Paradox
- [Hook, Howe 1986] Impredicative Strong Existential Equivalent to Type:Type
- [Pfenning, Paulin 1995] Inductively defined types in the Calculus of Constructions
- [Coquand, Paulin 1990] Inductively defined types
- [Hurkens 1995] A Simplification of Girard's Paradox
- [Coquand 1995] A new paradox in type theory
- [Barbanera, Berardi 1996] Proof-irrelevance out of excluded middle and choice in the calculus of constructions