Impredicative Observational Equality

19 January 2023

Loïc Pujet, Nicolas Tabareau

What is impredicativity?

What is impredicativity?

Impredicativity is a somewhat vague concept that has been around in logic since the early 20th century.

“A definition is impredicative if it involves a set that contains the thing that is being defined.”

Example: let G be a group, and A be a subset of G. The subgroup generated by A is the intersection of all the subgroups of G that contain A.

Impredicativity in Dependent Type Theory

In dependent type theory, a sort is said to be impredicative if it is closed under dependent products indexed over *any type.*

$x : A \vdash B : \mathrm{Prop} \longrightarrow \Pi\,(x : A)\,.\,B : \mathrm{Prop}$ Impredicative

$x : A \vdash B : \mathrm{Type}\ \ell \longrightarrow \Pi\,(x : A)\,.\,B : \mathrm{Type}\ \ell'$ Predicative

Impredicativity in Dependent Type Theory

Having an impredicative sort is useful:

- It reduces issues with universe levels
- It is necessary to define many mathematical abstractions, for instance any non-trivial complete lattice and Tarski's fixed point theorem
- Some theorems *require* impredicativity:
  normalization of System F…

Impredicativity in Dependent Type Theory

Proof assistants have diverging stances regarding impredicativity

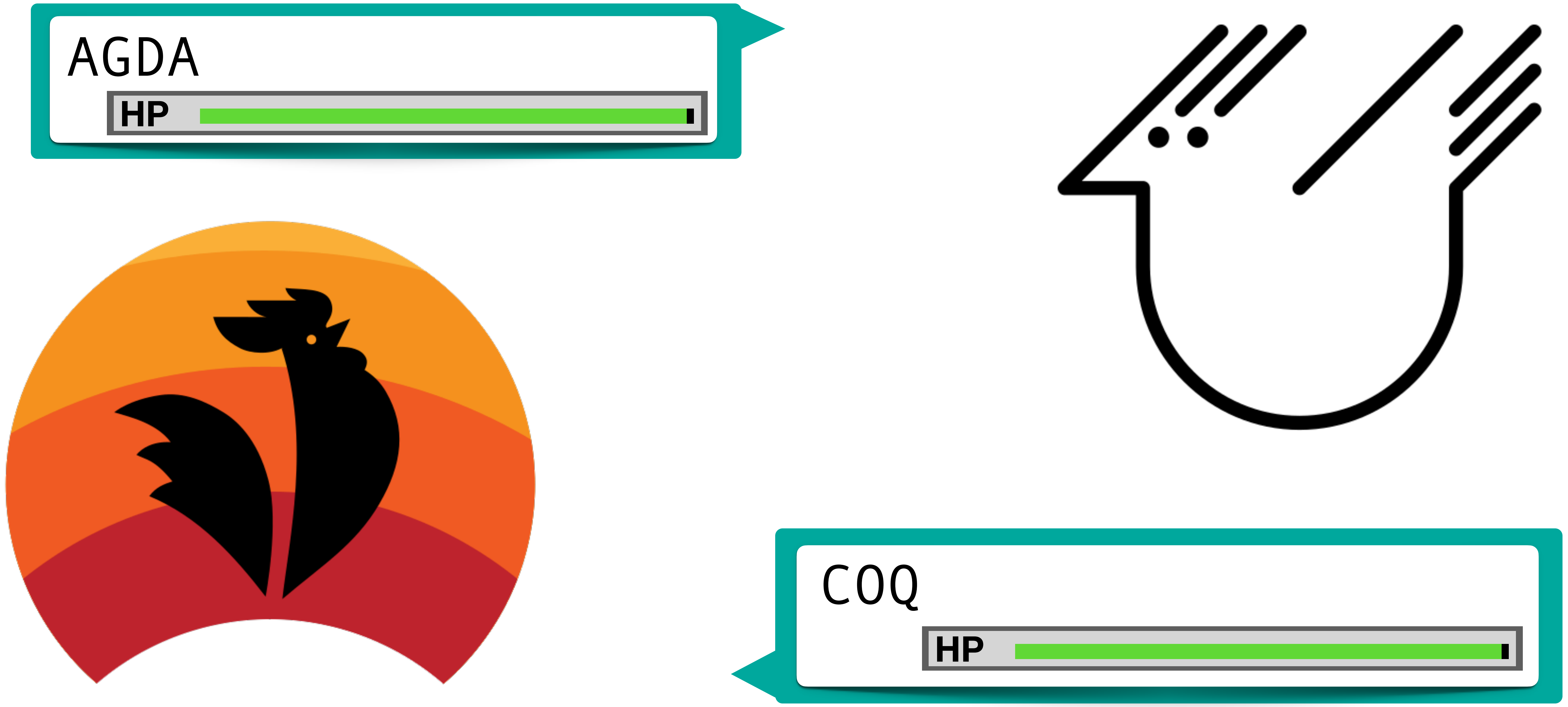
Coq	Lean	Agda
Impredicative	Impredicative	Predicative

Impredicativity in Dependent Type Theory

Impredicativity increases the logical strength of Coq by a lot!

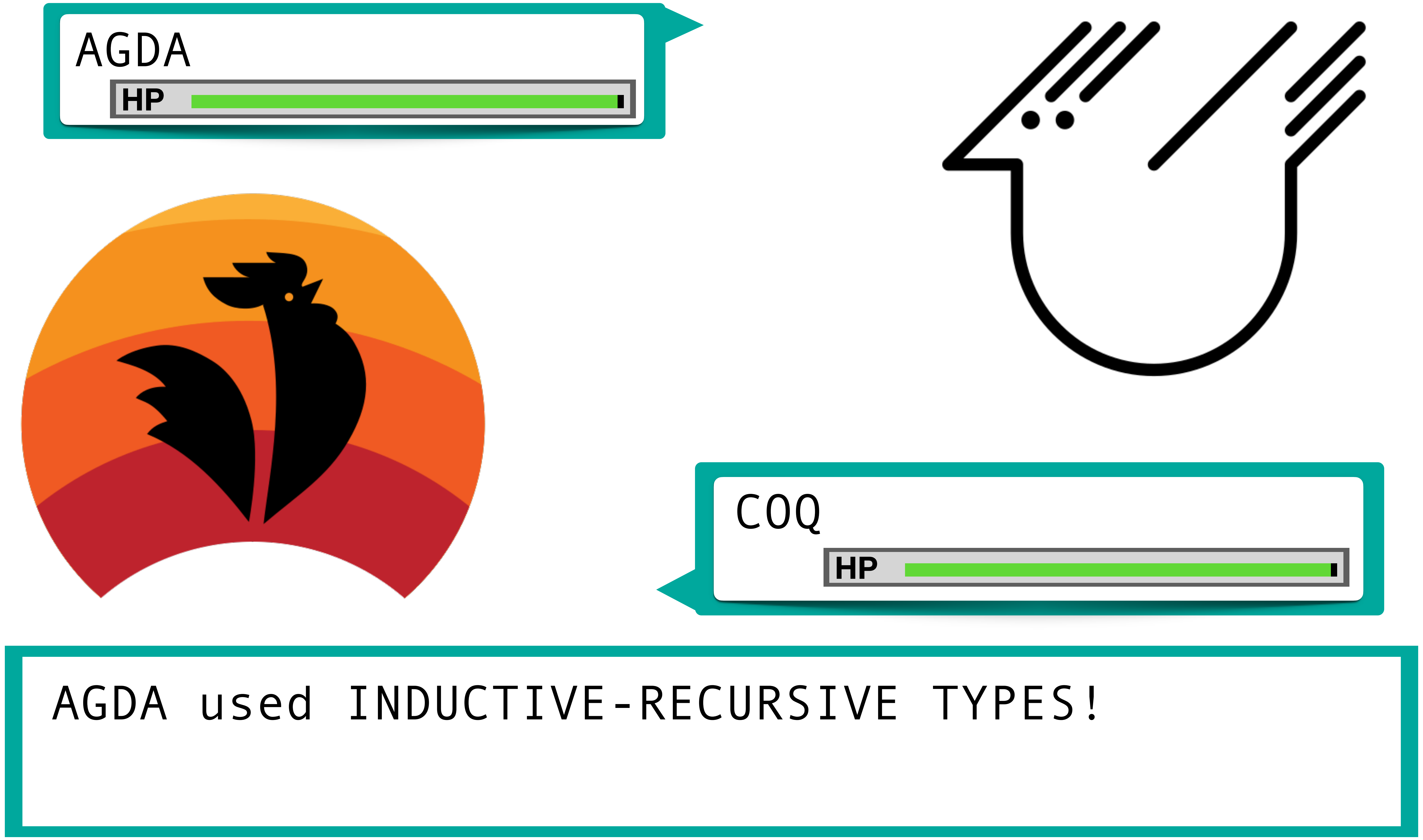
Impredicativity in Dependent Type Theory

Impredicativity increases the logical strength of Coq by a lot!



Impredicativity in Dependent Type Theory

Impredicativity increases the logical strength of Coq by a lot!



Impredicativity in Dependent Type Theory

Impredicativity increases the logical strength of Coq by a lot!

AGDA
HP

COQ
HP

AGDA used INDUCTIVE-RECURSIVE TYPES!

Impredicativity in Dependent Type Theory

Impredicativity increases the logical strength of Coq by a lot!

AGDA

HP

COQ

HP

COQ used IMPREDICATIVE PROPOSITIONS!

Impredicativity in Dependent Type Theory

Impredicativity increases the logical strength of Coq by a lot!

AGDA
HP

COQ
HP

COQ used IMPREDICATIVE PROPOSITIONS!
It's super effective!

Difficulties with impredicativity

Difficulties with impredicativity

It is difficult to build models for impredicative theories:

- It is impossible to prove consistency of impredicativity in a predicative theory
- The interdependency between “small” and “large” types makes models more rigid
- We still don’t have a solid normalization model with impredicative propositions, universes and η

Difficulties with impredicativity

As a consequence, impredicativity clashes with several other principles:

- Injectivity of type constructors
- Excluded middle + large elimination for inductive propositions
- An equality with definitional uniqueness of identity proofs

Impredicativity is a bit of a deal with the devil!

Hur. “Agda with excluded middle is inconsistent”, Agda mailing list
Barbanera, Berardi. Proof-irrelevance out of Excluded-middle and Choice in the CoC (1996)
Abel, Coquand. Failure of Normalization in Impredicative Type Theory […] (2019)

Contributions

Taming the devil with proof irrelevance

1st contribution: a new way to build normalization models for systems with impredicative, proof-irrelevant propositions

$$\frac{A : \mathrm{SProp} \qquad e, e' : A}{e \equiv e'}$$

Basically we can fit SProp in a predicative model construction if

- we give a very weak interpretation for types in SProp
- we don't normalize their inhabitants (which are irrelevant anyway!)

Taming the devil with proof irrelevance

Does proof-irrelevance forces us to have a system with two separate, non-interacting layers?

Not quite: we can add support for large elimination principles of propositions

- the False proposition
- the equality proposition (implies uniqueness of identity proofs)

Uniqueness of Identity Proofs

We need a computation rule to eliminate equality

$$\frac{P : A \to \mathrm{Type} \qquad x, y : A \qquad t : P\ x \qquad e : x = y}{\mathrm{transport}(P, x, y, t, e) : P\ y}$$

Uniqueness of Identity Proofs

We need a computation rule to eliminate equality

$$\frac{P : A \to \mathrm{Type} \qquad x, y : A \qquad t : P\ x \qquad e : x = y}{\mathrm{transport}(P, x, y, t, e) : P\ y}$$

We cannot do pattern matching on e anymore, since it is irrelevant

Uniqueness of Identity Proofs

We need a computation rule to eliminate equality

$$\frac{P : A \to \text{Type} \qquad x, y : A \qquad t : P\ x \qquad e : x = y}{\text{transport}(P, x, y, t, e) : P\ y}$$

We could compute by checking convertibility of x and y

Abel and Coquand used this rule to build a non-normalizing term in Lean, but it seems fine if we do not reduce irrelevant terms.

Abel, Coquand. Failure of Normalization in Impredicative Type Theory […] (2019)

Uniqueness of Identity Proofs

We need a computation rule to eliminate equality

$$\frac{P : A \to \mathrm{Type} \qquad x, y : A \qquad t : P\ x \qquad e : x = y}{\mathrm{cast}(P\ x, P\ y, \mathrm{ap}\ P\ e, t) : P\ y}$$

The second option is to inspect P x and P y

We can express transport from a more primitive type-casting operator, that computes by recursion on the universe.

→ This is Observational Type Theory!

Impredicative Observational Type Theory

So, can we prove normalization for an impredicative version of OTT?

There seems to be a fundamental conflict here: the traditional way to model OTT is with an inductive description of the universe, which is not traditionally compatible with impredicativity

But our new model construction is actually *predicative*!

Impredicative Observational Type Theory

2nd contribution: a formal proof of normalization, canonicity and decidability of conversion for impredicative OTT

Our system supports large elimination for False, equality, and all non-recursive propositions, but does not support elimination of the accessibility predicate.

Theorem: impredicative OTT can define the same integer functions as predicative type theory, but has the logical power of impredicativity.

Thank you!