A RESTRICTION OF REFLECTION COMPATIBLE WITH UNIVALENCE

Théo Winterhalter, supervised by Andrej Bauer and Matthieu Sozeau

Two notions of equality in type theory

Conversion

Identify things up to congruence and the β-rule of λ-calculus.

`(λx.t) u ≡ t[x ← u]`

This is the notion used for *type conversion*: if `t : A` and `A ≡ B` then `t : B`. You cannot refer to it from within the theory.

Identity types

Internal notion to talk about equality. The type $u =_A v$ represents equalities between **u** and **v** at type **A**. It can be manipulated within the theory, meaning you can prove propositional equalities. It is usually defined as an inductive type with *reflexivity* as its only constructor: for `u : A` we have $\mathtt{refl}\ u : u =_A u$.
If `u ≡ v` then `refl u` can witness `u = v` by type conversion.

Additional principles for equality

Reflection

Reflection is a rule that can be added to type theory, when doing so we say the theory is *extensional*.

$$\frac{u =_A v}{u \equiv v : A}$$

However, this also means that *type checking* becomes undecidable. Extensional type theories are still considered and are even at the core of several proof assistants like Andromeda and NuPrl.

Uniqueness of identity proofs

This principle (written **UIP**) states that any two proofs of equality `p q : u = v` are themselves equal: there exists a proof `r : p = q`. This unprovable property can be reformulated as the identity types being proof irrelevant, there is at most one inhabitant of an equality (it is either true or false, but does not hold any complexity or structure).

From **reflection** and the **elimination principle of identity types** (called **J**) we can deduce **UIP**.

INCOMPATIBILITY

Univalence and UIP don't go well together: indeed univalence implies that the type `bool = bool` is inhabited by both reflexivity and the negation on `bool` (which is an equivalence and thus an equality), contradicting **UIP**. This means in particular that **unrestricted** reflection and univalence are not compatible, even though people would like to consider them together.

Univalence

In homotopy type theory, the **univalence axiom** states that the trivial map from the type `A = B` to the type `A ~ B` of equivalences between type **A** and type **B**—basically the bijections—is itself an equivalence. This means that equality is considered **up to equivalence** (since from any equivalence we can deduce an equality).

A restricted reflection compatible with univalence

Type theory with **reflection** only **on boolean** equalities → **HTS** (or homotopy type system, a type theory featuring types with univalence and types with reflection) → **2-level type system** (a type theory featuring the same distinction but with only UIP (no reflection) for the non-univalent part)

We translate from a type theory with our extra reflection on **`bool`** rule into HTS which makes the distinction between strict (with reflection) equality and univalent equality. In this system, **`bool`** has the property that the univalent equality implies the strict one, so it validates reflection for it (we translate every type to its univalent counterpart, including equality).

Finally, we translate from HTS into 2-level type system which does not feature reflection by adapting Oury's translation to this setting, all the while managing a little optimisation in order not to require any extra axiom.
With this we conclude that we can indeed assume reflection for specific types—in this case `bool`—in a univalent setting.

References
–Thorsten Altenkirch *et al*. Extending homotopy type theory with strict equality.
–Andrej Bauer *et al*. The hott library: A formalization of homotopy type theory in coq.
–Martin Hofman *et al*. The groupoid model refutes uniqueness of identity proofs. In Logic in Computer Science, 1994.
–Nicolas Oury. Extensionality in the calculus of constructions. In International Conference on Theorem Proving in Higher Order Logics.

école normale supérieure paris—saclay
2016—2017