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.

\[(\lambda x . t) \equiv t[x \mapsto u]\]

This is the notion used for type conversion: If \(t : A\) and \(A \equiv B\) 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 \(\text{refl } u : u =_A u\).

If \(u \equiv v\) then \(\text{refl } u\) can witness \(u = v\) by type conversion.

Additional principles for equality

Univalence
In homotopy type theory, the univalence axiom states that the trivial map from the type \(A = B\) to the type \(A \sim B\) 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).

Uniqueness of identity proofs
This principle (written \(\text{UIP}\)) states that any two proofs of equality \(p \equiv q : u = v\) 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).

\[\text{From reflection and the elimination principle of identity types (called } J\text{) we can deduce UIP.}\]

A restricted reflection compatible with univalence

Type theory with reflection only on boolean equalities
We translate from a type theory with our extra reflection on \(\text{bool}\) rule into HTS which makes the distinction between strict (with reflection) equality and univalent equality. In this system, \(\text{bool}\) 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).

HTS or homotopy type system, a type theory featuring types with univalence and types with reflection
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.

2-level type system a type theory featuring the same distinction but with only UIP (no reflection) for the non-univalent part
With this we conclude that we can indeed assume reflection for specific types — in this case \(\text{bool}\) — in a univalent setting.

References
– Andrej Bauer et al. The hott library: A formalization of homotopy type theory in coq.