Homotopy Level Zero of Sub-Types

by Sven Nilsen, 2020

In this paper I show that elements of type-checked sub-types have homotopy level zero.

An arbitrary sub-type can be written in the form:

    x : [f] a

A homotopy level zero is a contractible type `T` such that:

    contractible(T : type) = ∃ a : T { ∀ b : T { b == a } }

In the homotopy-theoretic interpretation, this means that homotopy level zero is a contractible space.
This definition applies to notions of equality in homotopy type theory.

To translate this over to sub-types, I use an equivalence `x ~= y` as the existence of a homotopy path between `x` and `y` in `[f] a`. By applying `f` to this equivalence, one gets the equivalence `a ~= a`. This equivalence is a tautology construction by reflection, meaning that its type is always inhabited.

∵ x : [f] a   y : [f] a

∴ f(x ~= y)
∴ f(x) ~= f(y)
∴ a ~= a

This proof uses equivalence operator overloading from Sized Type Theory.
The use of equivalence here is a trick to prove that `x ~= y` is inhabited in `[f] a`.

This holds for any equivalence between pairs of elementes in `[f] a`, as shown by the following:

    ∃ a : T { ∀ x : [f] a { f(x) == a } }

Notice that this is almost the same as the definition of homotopy level zero in homotopy type theory.

However, the univalence axiom states that equality is equivalent to equivalence:

    (A == B) ~= (A ~= B)

By showing that the equivalence `x ~= y` is inhabited in `[f] a`, this is equivalent to showing that they are “equal” in the sense of homotopy type theory. The sub-type can be contracted to a single element:

    ∃ y : [f’] a { ∀ x : [f’] a { x == y } }   `f’` is contracted version of `f`

This shows that elements of type-checked sub-types have homotopy level zero.