Proving Equality in Two Different Ways

by Sven Nilsen, 2017

In this paper I talk about two ways of proving equality. To avoid confusion I color one in orange and the other in blue.

When proving equality between two kinds of types, the common way is to define two functions that when composed with each other gives the identity function:

$$f \cdot g \Leftrightarrow id_B$$

$$f : A \rightarrow B$$
$$g : B \rightarrow A$$

Isomorphism means every object in one type corresponds to an object in the other type, and is defined as two equations:

$$f \cdot g \Leftrightarrow id_B$$
$$g \cdot f \Leftrightarrow id_A$$

In the definition above, `g` can be written `$f^{-1}$`. Normally, this is how we think about inverse functions.

It turns out that there is another way of thinking about inverse functions. Instead of proving equality of two types `A` and `B` in terms of each other, one can start with the identity function of a third type `C` and then take the following path:

$$id_C[f \rightarrow g] \Leftrightarrow g \cdot f^{-1}$$

$$f : C \rightarrow A$$
$$g : C \rightarrow B$$

The type `C` behaves as a “witness” of the equality, and the existence of the path is the proof. The isomorphism is proved by finding two paths:

$$id_C[f \rightarrow g] : A \rightarrow B$$
$$id_C[g \rightarrow f] : B \rightarrow A$$

Notice that the type of these paths corresponds to the types of `f` and `g`. Under the strict notion of inverse functions, they are logically equivalent.

The benefit of the second way of proving equality is generalization. By generalizing the concept of inverse functions in various ways, one can derive learning algorithms for isomorphisms.