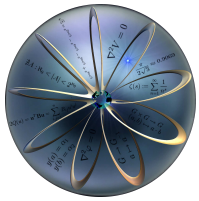

planetmath.org

Math for the people, by the people.

A.2.11 Definitions

Canonical name	A211Definitions
Date of creation	2013-11-09 5:44:46
Last modified on	2013-11-09 5:44:46
Owner	PMBookProject (1000683)
Last modified by	PMBookProject (1000683)
Numerical id	1
Author	PMBookProject (1000683)
Entry type	Feature
Classification	msc 03B15

Although the rules we listed so far allows us to construct everything we need directly, we would still like to be able to use named constants, such as isequiv, as a matter of convenience. Informally, we can think of these constants simply as abbreviations, but the situation is a bit subtler in the formalization.

For example, consider function composition, which takes $f : A \to B$ and $g : B \to C$ to $g \circ f : A \to C$. Somewhat unexpectedly, to make this work formally, $\circ$ must take as arguments not only f and g, but also their types A, B, C:

$$\circ \equiv \lambda(A : \mathcal{U}_i).\, \lambda(B : \mathcal{U}_i).\, \lambda(C : \mathcal{U}_i).\, \lambda(g : B \to C).\, \lambda(f : A \to B).\, \lambda(x : A).\, g(f(x)).$$

From a practical perspective, we do not want to annotate each application of $\circ$ with A, B and C, as the are usually quite easily guessed from surrounding information. We would like to simply write $g \circ f$. Then, strictly speaking, $g \circ f$ is not an abbreviation for $\lambda(x : A).\, g(f(x))$, because it involves additional **implicit arguments** which we want to suppress.

Inference of implicit arguments, typical ambiguity (`http://planetmath.org/13universesandf` ensuring that symbols are only defined once, etc., are collectively called **elaboration**. Elaboration must take place prior to checking a derivation, and is thus not usually presented as part of the core type theory. However, it is essentially impossible to use any implementation of type theory which does not perform elaboration; see [**?**, **?**] for further discussion.