syn.1 Terms

lam:syn:trm:sec The terms of the lambda calculus are built up inductively from an infinite supply of variables v_0, v_1, ..., the symbol "λ", and parentheses. We will use x, y, z, ... to designate variables, and M, N, P, ... to desginate terms.

lam:syn:trm:defn:term **Definition syn.1** (Terms)**.** The set of *terms* of the lambda calculus is defined inductively by:

lam:syn:trm:defn:term-var 1. If x is a variable, then x is a term.

lam:syn:trm:defn:term-abs 2. If x is a variable and M is a term, then $(\lambda x.\, M)$ is a term.

lam:syn:trm:defn:term-app 3. If both M and N are terms, then (MN) is a term.

If a term $(\lambda x.\, M)$ is formed according to (2) we say it is the result of an *abstraction*, and the x in λx is called a *parameter*. A term (MN) formed according to (3) is the result of an *application*.

The terms defined above are fully parenthesized. This can get rather cumbersome, as the term $(\lambda x.\, ((\lambda x.\, x)(\lambda x.\, (xx))))$ demnostrates. We will introduce conventions for avoiding parentheses. However, the official definition makes it easy to determine how a term is constructed according to Definition syn.1. For example, the last step of forming the term $(\lambda x.\, ((\lambda x.\, x)(\lambda x.\, (xx))))$ must be abstraction where the parameter is x. It results by abstraction from the term $((\lambda x.\, x)(\lambda x.\, (xx)))$, which is an application of two terms. Each of these two terms is the result of an abstraction, and so on.

Problem syn.1. Describe the formation of $(\lambda g.\, (\lambda x.\, (g(xx)))(\lambda x.\, (g(xx))))$.

Photo Credits

Bibliography