A new framework for the λ-calculus

Rob Nederpelt
and
Fairouz Kamareddine
Department of Mathematics and Computing Science
Eindhoven University of Technology
Eindhoven, the Netherlands

August 1992

λ-calculus is fundamental for the foundation of Logic, Mathematics, Computer and Cognitive Science. This makes it indispensable to formalize the λ-calculus in a way which avoids most of the complications associated with, amongst others, substitution, variable renaming and the search of bound and free occurrences of variables. Such a formulation will have advantages for all areas that use λ-calculus, whether theoretical (such as Logic and Foundation of Mathematics) or applied (such as Functional and Logic programming).

We address this problem by setting the ground for a λ-calculus notation that strongly mirrors the two fundamental operations of term construction, namely *abstraction* and *application*. In particular, we single out those parts of a term, called *items* in our framework, that are added during abstraction and application. This item notation proves to be a powerful device for the representation of basic substitution steps, giving rise to different versions of β-reduction including local and global β-reduction. In other words substitution is formalized as an object language notion rather than remaining a meta language one. This substitution is, we show, the most general up to date as it can accommodate most of the known reduction strategies. In addition, we show that our notation is useful for term and variable manipulation and for locating the type of a term.

After setting out this notation and introducing object level substitution, we exploit the new framework by showing that it proves powerful in accommodating important notions of the λ-calculus. We discuss the role of types in the presented setting and provide a type operator which gives a representative type for a typeable term. Moreover, in accommodating types in our system, it turns out that a general framework for many typed lambda calculi can be obtained in the general setting. Another attraction of our new approach is that by specifying a number of parameters, one defines one system of typed lambda calculus or another. In fact, it turns out that many known systems of typed λ-calculus fit in the proposed setting, in particular the ones connected with "Barendregt's cube". The general framework leads naturally to a number of generalizations. It gives much freedom and is at the same time simple and perspicuous. It allows theorists to compare the different systems as to important properties and enables practical users to make their choices at the relevant place.