Sliced bananas on opaque data*

The expression lemma

Ralf Lämmel[1] and Ondrej Rypacek[2]

[1] The University of Koblenz-Landau, Germany
[2] The University of Nottingham, UK

Abstract. Algebraic data types and catamorphisms (folds) play a central role in functional programming as they allow programmers to define recursive data structures and operations on them uniformly by structural recursion. Likewise, in object-oriented (OO) programming, recursive hierarchies of object types with virtual methods play a central role for the same reason. There is a semantical correspondence between these two situations which we reveal and formalize categorically. To this end, we assume a coalgebraic model of OO programming with functional objects. In practical terms, the development prepares for refactorings that turn sufficiently disciplined functional folds into OO programs of a designated shape (and v.v.).

Key words: expression lemma, expression problem, functional object, catamorphism, fold, composite, program calculation, distributive law, free monad, cofree comonad.

* This is an abstract for a paper that appears, in its short version, in "Mathematics in Program Construction 2008" and that, in its full version, is being submitted for journal publication. Please use the following URL to access the paper and the source-code distribution: `http://www.uni-koblenz.de/~laemmel/expression/`.