Computational Comonads and Intensional Semantics

Stephen Brookes[*] *Shai Geva*[*]

Abstract

We explore some foundational issues in the development of a theory of intensional semantics, in which program denotations may convey information about computation strategy in addition to the usual extensional information. Beginning with an "extensional" category $\mathcal{C}$, whose morphisms we can think of as functions of some kind, we model a notion of computation using a comonad with certain extra structure and we regard the Kleisli category of the comonad as an intensional category. An intensional morphism, or algorithm, can be thought of as a function from computations to values, or as a function from values to values equipped with a computation strategy. Under certain rather general assumptions the underlying category $\mathcal{C}$ can be recovered from the Kleisli category by taking a quotient, derived from a congruence relation that we call extensional equivalence. We then focus on the case where the underlying category is cartesian closed. Under further assumptions the Kleisli category satisfies a weak form of cartesian closure: application morphisms exist, currying and uncurrying of morphisms make sense, and the diagram for exponentiation commutes up to extensional equivalence. When the underlying category is an ordered category we identify conditions under which the exponentiation diagram commutes up to an inequality. We illustrate these ideas and results by introducing some notions of computation on domains and by discussing the properties of the corresponding categories of algorithms on domains.