

Abstract

We present a category-theoretic framework for providing intensional semantics of programming languages and establishing connections between semantics given at different levels of intensional detail. We use a comonad to model an abstract notion of computation, and we obtain an intensional category from an extensional category by the co-Kleisli construction; thus, while an extensional morphism can be viewed as a function from values to values, an intensional morphism is akin to a function from computations to values. We state a simple category-theoretic result about cartesian closure. We then explore the particular example obtained by taking the extensional category to be **Cont**, the category of Scott domains with continuous functions as morphisms, with a computation represented as a non-decreasing sequence of values. We refer to morphisms in the resulting intensional category as algorithms. We show that the category **Alg** of Scott domains with algorithms as morphisms is cartesian closed. We define an intensional partial order on algorithms, with respect to which application, currying and composition are continuous. We show that every algorithm determines a continuous input-output function, and that every continuous function is the input-output function of some algorithm. This is in contrast to the sequential algorithms model of Berry and Curien, in which algorithms determine the sequential input-output functions (between concrete domains). Since the continuous functions include inherently non-sequential functions such as parallel-or, we designate our algorithms as parallel. Two algorithms are input-output equivalent iff they have the same input-output function. The intensional ordering on algorithms collapses, under input-output equivalence, onto the pointwise ordering on their input-output functions. We define an intensional semantics of the simply typed λ -calculus, and relate it to the standard extensional semantics. We discuss related work and we propose some topics for further research.