MODELLING COMPUTATIONS: A 2-CATEGORICAL FRAMEWORK

R.A.G. SEELY

Dept. of Mathematics, John Abbott College, P.O. Box 2000
Ste. Anne de Bellevue, Quebec H9X 3L9
and
Dept. of Mathematics, McGill University, Montreal, Quebec

ABSTRACT

An introduction to 2-categories is given by illustrating how the structure of typed lambda calculus may naturally be viewed as a 2-category. In this vein, the structure of computations or conversions gives rise to notions of lax 2-adjointness.

0. Introduction

It has become standard, when modelling the lambda calculus (first or higher order), to treat beta conversion as equality (and frequently eta conversion as well.) In particular, all categorical semantics (eg. for lambda calculus, in terms of C-monoids -- for untyped -- or cartesian closed categories -- for typed -- (LAMBEK-SCOTT [1986]), and for polymorphic lambda calculus, in terms of PL categories (SEELY [1986])) have this property. However, there is no doubt that something is lost with such an approach, since beta conversion

$$(\lambda\, x \text{ in } A.\ a)\ (b) = a[x:=b]$$

explicitly equates each stage in a computation process with the result of the computation.

There are several standard approaches to semantics of lambda calculus which do not make such identifications, particularly for the untyped lambda calculus, and particularly in the setting of domain models. For example, a model of untyped lambda calculus may be given by a triple (D,h,k), where D is a domain, h a map $D \longrightarrow (D \Rightarrow D)$ and k a map $(D \Rightarrow D) \longrightarrow D$, satisfying suitable conditions. (For the moment, let me not specify what "domain" means, nor what kinds of maps h and k are.) In such a context, if h and k are inverse isomorphisms, then D is an extensional model, where both beta and eta conversions are interpreted as identities. Non-extensional models arise from weakening the condition on h, k: for instance, if

$$hk = id(D \Rightarrow D) \quad \text{and} \quad kh \leq id(D)$$

where $\leq$ is defined, usually pointwise, in terms of the order on D, then we get a non-extensional model in which beta conversion is identity, and eta conversion is increasing:

$$(\lambda\, x.\ a(x)) \leq a \quad (\text{for } x \text{ not free in } a).$$

Categorically, we would say that k is left adjoint to h, and so $(D \Rightarrow D)$ is a coreflective subcategory (or subdomain) of D. (SCOTT [1972] calls $(D \Rightarrow D)$ a projection of D.)

Alternatively,

$$hk = id(D \Rightarrow D) \quad \text{and} \quad id(D) \leq kh$$

gives a non-extensional model in which eta conversion is decreasing: ie. h is a left adjoint to k and so $(D \Rightarrow D)$ is a reflective subcategory of D. This is essentially the situation of Scott's $\mathcal{P}\omega$ model, SCOTT [1976].

Finally, replacing $hk = id(D \Rightarrow D)$ with, for instance,

$$hk \leq id(D \Rightarrow D)$$

yields a lambda structure with increasing beta conversion:

$$(\lambda\, x.\ a)(b) \leq a[a:=b].$$

Pairing with this a decreasing eta conversion, $id(D) \leq kh$, yields an adjunction: h is left adjoint to k, (but $(D \Rightarrow D)$ need not be a subcategory of D.)

This last will be the motivating example for the structures of this paper. However, we generalise the usual domain framework by replacing poset structure with categorical structure. We shall work with the typed lambda calculus, for simplicity -- however, I shall indicate at the end how this extends to polymorphic lambda calculus, which is why we use the typed structure. The lambda structure we wish to abstract in this way consists of the types, the terms, and the conversions. This will produce a 2-category, rather than just a category: the types will be the objects, the terms the morphisms, and the conversions will be the 2-cells of this 2-category. The point of this is that the structure of $\Rightarrow$ given by beta and eta conversion amounts to a weak ("lax") notion of adjunction, corresponding in the usual categorical setting to cartesian closedness.