Interaction morphisms and Turing computation

Tarmo Uustalu

Dept. of Software Science, Tallinn University of Technology, Estonia
tarmo@cs.ioc.ee

I will introduce interaction morphisms are a means to specify how an effectful (e.g., non-deterministic, interactive I/O or stateful) computation is to be run on an abstract state machine. An interaction morphism is given by a monad $T = (T, \eta, \mu)$ and a comonad $D = (D, \varepsilon, \delta)$ on a category with finite products together with a family of maps $\psi_{X,Y} : TX \times DY \to X \times Y$ natural in X and Y and agreeing suitably with η, ε, μ, δ. Intuititively, $\psi_{X,Y}$ takes a computation and a behavior from an initial state and sends them into a return value and a final state. Interaction morphisms enjoy a number of neat properties. Interaction morphisms are the same as monoids in a certain monoidal category; the category of interaction morphisms comes with a rich structure. Interaction morphisms of T and D are in a bijective correspondence with carrier-preserving functors between the categories of coalgebras of D and stateful runners of T (monad morphisms from T to state monads); they are in a bijective correspondences with comonad morphisms from the comonad determined by the algebraic theory for T to the comonad D; they are also in a bijective correspondence with monad morphisms from T to a monad induced in a certain way by D.

I will illustrate interaction morphisms on the example of Turing computation, i.e., computations interacting with a reading-writing head moving along a bi-infinite tape storing symbols from a finite alphabet. Identifying and describing the useful monads of different levels of intensionality/extensionality for this particular case as well as the different interacting comonads is an instructive exercise. The options for wellfounded Turing computations range from the relevant free monad to its quotient down to a submonad of the store monad where the store records the position of the head and the contents of the tape. The different monads can be decomposed into combinations of simpler monads accounting for reading/writing an individual tape cell, moving only, reading and moving only etc.; similarly for the corresponding comonads. Replacing wellfounded computations with rational computations (i.e., the free monad with the free iterative monad etc.) gives further variations.

The work on interaction morphisms, which is ongoing and joint with Shin-ya Katsumata, continues my earlier work [4] on stateful runners. It shares some motivation, but departs technically from Plotkin and Power's work on tensors of models and comodels [3]. The extensional monad for wellfounded Turing computations was described by Goncharov et al. [1]. Kůrka [2] initiated a line of study of Turing machines in topological dynamics, which is quite relevant for this work.

References

1. S. Goncharov, S. Milius, A. Silva. Towards a coalgebraic Chmomsky hierarchy (extended abstract). In J. Díaz, I. Lanese, D. Sangiorgi, eds., *Proc. of 8th IFIP TC1/WG 2.2 Int. Conf. on Theor. comput. Sci., TCS 2014*, v. 8705 of *Lect. Notes in Comput. Sci.*, pp. 265–280, Springer, 2014. Extended version: arXiv preprint 1401.5277, 2014.
2. P. Kůrka. On topological dynamics of Turing machines. *Theor. Comput. Sci.*, v. 174, n. 1–2, pp. 203–216, 1997.
3. G. Plotkin, J. Power. Tensors of comodels and models for operational semantics. In A. Bauer, M. Mislove, eds., *Proc. of 24th Conf. on Mathematical foundations of Programming Semantics, MFPS XXIV*, v. 218 of *Electron. Notes in Theor. Comput. Sci.*, pp. 295–311. Elsevier, 2008.
4. T. Uustalu. Stateful runners for effectful computations. In D. Ghica, ed., *Proc. of 31st Conf. on Mathematical foundations of Programming Semantics, MFPS XXXI*, v. 319 of *Electron. Notes in Theor. Comput. Sci.*, pp. 403–421. Elsevier, 2015.