Towards coalgebraic semantics of higher-order behaviours

Marco Peressotti
Dept. of Mathematics, Computer Science and Physics, University of Udine, Italy
marco.peressotti@uniud.it

It is well known that *higher-order systems*, i.e. systems which can pass around systems of the same kind, like the λ-calculus [1], the calculus of higher-order communicating systems (CHOCS) [10], the higher-order π-calculus (HOπ) [7], HOcore [4], etc., are difficult to reason about. Many bisimulations and proof methods have been proposed also in recent works [2,4,5,8,9]. This effort points out that a definition of *abstract* higher-order behaviour is still elusive. In this paper, we show how these abstract behaviours can be modelled as the *final coalgebras* of suitable *higher-order behavioural functors*.

Coalgebras are a well established framework for modelling and studying concurrent and reactive systems. Despite these results, a general coalgebraic treatment of higher-order systems is still missing. In fact, defining these functors for higher-order behaviours is challenging. In order to describe the problem, let us consider first a functor for representing the behaviour of a first-order calculus, like CCS with value passing:

$$B\colon \mathsf{Set} \to \mathsf{Set} \qquad B = \mathcal{P}_\omega(C \times V \times \mathcal{I}d + C \times \mathcal{I}d^V + \mathcal{I}d) \tag{1}$$

where C is a set of channels and V is the set of values [3]. This functor is well-defined, and it admits a final coalgebra which we denote by νB; the carrier of this coalgebra is the set $|\nu B|$.

In a higher-order calculus like HOπ, the values that processes can communicate are processes themselves. However, actions communicating semantically equivalent (herein, strongly bisimilar) processes have to be considered equivalent even if the values/processes exchanged are syntactically different. In other words, this means that from the semantics perspective higher-order behaviours communicate *behaviours*. To reflect this fundamental observation in the definition (1) we must replace the set of exchanged values V with the set of all possible behaviours i.e. the carrier of the final B-coalgebra yielding the following definition:

$$B_{ho}\colon \mathsf{Set} \to \mathsf{Set} \qquad B_{ho} = \mathcal{P}_\omega(C \times |\nu B_{ho}| \times \mathcal{I}d + C \times \mathcal{I}d^{|\nu B_{ho}|} + \mathcal{I}d) \tag{2}$$

But this means that we are defining B_{ho} using its own final coalgebra νB_{ho}, which can be defined (if it exists) only after B_{ho} is defined—a circularity!

We think that this circularity is the gist of higher-order behaviours: any attempt to escape it would be restricting and distorting. One may be tempted to take as V some (syntactic) representation of behaviours (e.g., processes), but this would fall short. First, the resulting behaviours would not be really higher-order, but rather behaviours manipulating some *ad hoc* representation of behaviours. Secondly, we would need some mechanism for moving between behaviours and their representations–which would hardly be complete. Third, the resulting functor would not be abstract and independent from the syntax of processes, thus hindering the possibility of reasoning about the computational aspect on its own, and comparing different models sharing the same kind of behaviour.

We provide a general and syntax-agnostic characterisation endofunctors modelling higher-order operational behaviours in terms of solutions to certain recursive equations. The key idea is to consider the definition as an instance of an endofunctor $\mathfrak{F}(V)\colon \mathsf{Set} \to \mathsf{Set}$ parameterised in the object of values V. Then, we are interested in those instances whose final coalgebra is carried by the object of values i.e. those such that values are exactly all (abstract) behaviours. Actually, since this parameter may occur in both covariant and contravariant position, the functor is *biparametric*. In our example, $\mathfrak{F}\colon \mathsf{Set}^{op} \times \mathsf{Set} \to [\mathsf{Set}, \mathsf{Set}]$ given as:

$$\mathfrak{F}(X, Y) = \mathcal{P}_\omega(C \times Y \times \mathcal{I}d + C \times \mathcal{I}d^X + \mathcal{I}d) \tag{3}$$

where $[\mathsf{Set}, \mathsf{Set}]$ denotes the category of endofunctors over Set. More generally, we consider functors $F\colon \mathsf{C}^{op} \times \mathsf{C} \to [\mathsf{C}, \mathsf{C}]$. Then, we show how to define an initial sequence of endofunctors together with their final coalgebras such that its limit $(B, |\nu B|)$ exists and satisfies:

$$B \cong \mathfrak{F}(|\nu B|, |\nu B|). \tag{4}$$

Hence, B is the requested higher-order behavioural endofunctor. We prove that (4) has a unique solution if $\mathfrak{F}$ is Cpo-enriched and C is Cpo-algebraically compact. Under mild conditions, this result generalises to Cpo-functors like $\mathfrak{F}\colon \mathsf{D}^{op} \times \mathsf{D} \to [\mathsf{C}, \mathsf{C}]$. We refer the interested reader to [6] for further details and examples.

References

1. H. Barendregt. *The lambda calculus: its syntax and its semantics*. Studies in Logic and the Foundations of Mathematics. North-Holland, 1984.
2. L. Birkedal, R. E. Møgelberg, J. Schwinghammer, and K. Støvring. First steps in synthetic guarded domain theory: step-indexing in the topos of trees. *LMCS*, 8(4), 2012.
3. M. P. Fiore and D. Turi. Semantics of name and value passing. In *LICS*, pages 93–104. IEEE, 2001.
4. I. Lanese, J. A. Pérez, D. Sangiorgi, and A. Schmitt. On the expressiveness and decidability of higher-order process calculi. In *LICS*, pages 145–155. IEEE, 2008.
5. S. Lenglet, A. Schmitt, and J. B. Stefani. Characterizing contextual equivalence in calculi with passivation. *Information and Computation*, 209(11):1390–1433, 2011.
6. M. Peressotti. Endofunctors modelling higher-order behaviours. *CoRR*, abs/1602.06221, 2016.
7. D. Sangiorgi. Bisimulation for higher-order process calculi. *Information and Computation*, 131(2):141–178, 1996.
8. D. Sangiorgi, N. Kobayashi, and E. Sumii. Environmental bisimulations for higher-order languages. In *LICS*, pages 293–302. IEEE, 2007.
9. K. Støvring and S. B. Lassen. A complete, co-inductive syntactic theory of sequential control and state. In *SAS*, pages 329–375. Springer, 2009.
10. B. Thomsen. Plain CHOCS: a second generation calculus for higher order processes. *Acta informatica*, 30(1):1–59, 1993.