Bialgebraic Semantics and Recursion

Gordon Plotkin [⋆]

Division of Informatics, University of Edinburgh, King's Buildings,
Edinburgh EH9 3JZ, Scotland

In [9] a unifying framework was given for operational and denotational semantics. It uses bialgebras, which are combinations of algebras (used for syntax and denotational semantics) and coalgebras (used for operational semantics and solutions to domain equations). Here we report progress on the problem of adapting that framework to include recursion. A number of difficulties were encountered. An expected one was the need to treat orders in the general theory; much less expected was the need to give up defining bisimulations in terms of spans of functional bisimulations, and move to a relational view. Even so the outcome is not yet satisfactory because of well-known difficulties involved in prebisimulations. Our work can be compared to, e.g., that of [2]. The principal difference is that we aim, following [9], at a conceptual overview of the area using appropriate categorical tools.

The main idea of [9] is to represent rules for operational semantics by a natural transformation:

$$\rho : \Sigma(X \times BX) \to BTX$$

where Σ is the signature functor associated to an algebraic signature of the same name, B is a behaviour functor and T is the term monad associated to Σ. For suitable choices of these over the category of sets, image-finite sets of rules in GSOS format yield such natural transformations.

Models of such rules are bialgebras

$$\Sigma X \xrightarrow{\alpha} X \xrightarrow{\gamma} BX$$

satisfying a suitable pentagonal condition. The category of these models has an initial object consisting of the programming language $\Sigma L \overset{\iota}{\cong} L$ and its operational semantics $L \xrightarrow{\gamma} BL$. Every model gives an adequate compositional denotational semantics for the language L. Suppose now that the final coalgebra $M \overset{\sigma}{\cong} BM$ exists; it can be thought of as the solution to the "domain equation" $X \cong BX$ in the category at hand. Then it automatically gives a final object which incorporates a semantic algebra $\Sigma M \xrightarrow{\alpha} M$.

One can model bisimulation by spans of coalgebra maps (first done in [3]). With this, under mild conditions, one has that the semantics given by the final coalgebra is fully abstract and that there is a greatest bisimulation which is a congruence. These conditions are that kernel pairs exist and that weak kernel pairs are preserved by B.

[⋆] This work has been done with the support of EPSRC grant GR/M56333: The Structure of Programming Languages: Syntax and Semantics.

Turning to recursion we need first to consider syntax. One way would be to add μ-terms $\mu x.t$ to the usual algebraic terms over a given signature Σ, as in [2]. However, taken literally, we would then also have to incorporate an extension of the framework to handle binding operators, as in [5]. While that would be interesting, it would obscure the issues caused by recursion; we prefer to consider infinite terms, regarding the μ-terms as syntactic sugar for their unwinding.

We therefore consider the functor $\Sigma : \mathbf{CPPO} \to \mathbf{CPPO}$ over the category of cppos and strict continuous functions given by:

$$\Sigma(D) = \coprod_{f \in \Sigma} (D_\perp^{\mathrm{arity}(f)})$$

Its initial algebra is the free continuous algebra with signature Σ. A natural choice of behaviour functor is

$$B(D) = \mathcal{P}(A_\perp \otimes D_\perp)$$

where A is a set of actions, and $\mathcal{P}(D)$ is the free continuous semilattice with a zero over D (on bifinite cppos this is the convex powerdomain with a zero). The final coalgebra of B is the solution to Abramsky's domain equation for bisimulation [1].

Every image finite set of GSOS rules again gives rise to a natural transformation as above, but now over **CPPO**, and we may attempt to apply the general theory. However weak pullbacks are not preserved (because of convexity phenomena) and full abstraction fails in the sense that there are not enough bisimulations to equate processes with the same final semantics.

One can instead develop a general theory for locally ordered categories (better: **CPPO**-enriched ones). One can define simulations (or ordered bisimulations) as spans of lax and and oplax coalgebra maps; it is shown in [4] that these include partial bisimulations [1] (also known as prebisimulations). There is then a new version of the general theory but now with the conditions that insertors exist and that weak insertors are preserved. Unfortunately, the weak preservation condition fails.

This motivates us to give up considering bisimulations as spans and, instead, to work directly with relations. One introduces a category of relations, lifts the behaviour functor to the category and takes bisimulations to be coalgebras in the category of relations—see, e.g., [8]. Thus one asks that the coalgebra map is "logical," preserving the relation. One can argue this is less arbitrary than the span view where one has to turn the relation itself into a coalgebra, and there is not necessarily a canonical way to do this (and in our case there may be no way!).

A first attempt at a category for relations is $\mathbf{CPPO}_a$ whose objects are pairs (D, R) of cppos and admissible relations on them, and whose morphisms are the strict continuous functions preserving the relations. The various functors involved have natural liftings to the category of relations: liftings of a functor F correspond to families F_D of monotone actions on relations over D such that for any $f : (D, Q) \to (E, R)$,

$$F_D(f^{-1}(R)) \subseteq F(f)^{-1} F_E(R)$$

In the case of the powerdomain functor $\mathcal{P}$ there is a minimal such action which makes singleton, empty set and union logical. It is then natural to assume that ρ is logical, in the sense that it lives in $\mathbf{CPPO}_a$, and indeed this is the case when ρ arises from GSOS rules as above.

One can then show that full abstraction holds if a partial converse to the above condition on actions holds, viz that

$$F(h)^{-1} F_M(=_M) \subseteq F_L(h^{-1}(=_M))$$

where h is the unique coalgebra map from the operational semantics to the solution of the domain equation, and $=_M$ is the equality relation on M. However, this is false and full abstraction again fails; intuitively this is because we are again trying to handle equality (bisimulation) when convexity phenomena are present.

Instead one should use simulations, which we take to be admissible relations R such that:

$$\leq \circ R \circ \leq \;\subseteq\; R$$

and move to the subcategory $\mathbf{CPPO}_s$ whose objects are pairs of cpppos and simulations. The condition now needed is that:

$$F(h)^{-1} F_M(\leq_M) \subseteq F_L(h^{-1}(\leq_M))$$

where $\leq_M$ is the inequality relation on M. This holds under the condition that h preserves finiteness. Unfortunately that does not hold in general, though it does if the set of GSOS rules is compact in the sense of [2]. However we lack a conceptual proof of that assertion, and in any case we would prefer a general result. The essential difficulty is that an infinite process may be simulated by a set of processes rather than any single one. It may be possible to account for this by changing the kind of relations used, taking advantage of the scope of the fibrational approach to logical relations as discussed in, e.g., [7].

As regards possible further developments, one evident possibility for future work is to combine recursion and value-passing (e.g. as in the work of [6]). More immediately it would be good to treat other examples than GSOS. In particular probabilistic computation presents a challenge with its need for continuous domain theory.

References

1. S. Abramsky, *A Domain Equation for Bisimulation*, in Inf. and Comp.,, Vol. 92, pp. 161–218, 1991.
2. L. Aceto and A. Ingólfsdóttir, *CPO Models for GSOS Languages—Part I: Compact GSOS Languages*, in Inf. and Comp., Vol. 129, No. 2, pp. 107–141, 1994.

3. P. Aczel and N. Mendler, A Final Coalgebra Theorem, in *Proc. CTCS '89* (eds. D. H. Pitt, D. E. Ryeheard, P. Dybjer, A. M. Pitts and A. Poigné), LNCS Vol. 389, pp. 357–365, Berlin: Springer-Verlag, 1989.
4. M. P. Fiore, *A Coinduction Principle for Recursive Data Types Based on Bisimulation*, in Information and Computation, Vol. 127, No. 2, pp. 186–198, 1996.
5. M. P. Fiore, G. D. Plotkin and D. Turi, *Abstract Syntax and Variable Binding*, in Proc. LICS '99, pp. 193–202, Washington: IEEE Computer Society Press, 1999.
6. M. P. Fiore and D. Turi *Semantics of Name and Value Passing*, to appear in Proc. LICS 2001.
7. C. Hermida, and B. Jacobs, *Structural Induction and Coinduction in a Fibrational Setting*, Inf. & Comp., Vol. 145, No. 2, pp. 107–152, 1998.
8. B. Jacobs, *Mongruences and Cofree Coalgebras*, in Algebraic Methodology and Software Technology (eds. V. S. Alagar and M. Nivat), LNCS Vol. 936, pp. 245–260, Berlin: Springer-Verlag, 1995.
9. D. Turi and G. D. Plotkin, Towards a Mathematical Operational Semantics, in *Proc. LICS 97*, pp. 268–279, Washington: IEEE Press, 1997.