The topological behaviour category of an algebraic theory

RICHARD GARNER

Macquarie University
School of Mathematical and Physical Sciences
NSW 2109, Australia richard.garner@mq.edu.au

In computer science, algebraic theories are used to encode computational effects [2, 3]: operations of a theory encode new language primitives which may, for example, request input from, or return output to, an external source; read and write values in a store; branch probabilistically or non-deterministically; and so on.

Many computational effects involve interaction with an external environment, and an important insight of Power and Shkaravska [4] is that the environments in question can be modelled by *comodels* of one's algebraic theory. For example, a comodel of the theory of input is a state machine which provides input tokens on demand, while a comodel of the theory of store is a state machine which handles requests to read and update the values in the store.

One can also consider *topological* comodels of an algebraic theory, where the topology tells us how much of the hidden state of a comodel is revealed via *finite* interactions with a program. The goal of this talk is to explain how the topological comodels of a given theory $\mathbb{T}$ admit a particularly nice classification: they are precisely the topological $\mathbb{B}$-sets for a certain source-étale ample topological category $\mathbb{B}$, which we call the *topological behaviour category* of the theory $\mathbb{T}$. This extends results of [1] for non-topological comodels.

If time permits, we will discuss how the kinds of topological groupoid arising in the study of combinatorial C^*-algebras can be re-found as topological behaviour categories of computationally natural algebraic theories.

References

[1] GARNER, R. The costructure–cosemantics adjunction for comodels for computational effects. *Mathematical Structures in Computer Science* (2021), to appear.

[2] MOGGI, E. Notions of computation and monads. *Information and Computation 93* (1991), 55–92.

[3] PLOTKIN, G., AND POWER, J. Notions of computation determine monads. In *Foundations of software science and computation structures (Grenoble, 2002)*, vol. 2303 of *Lecture Notes in Computer Science*. Springer, Berlin, 2002, pp. 342–356.

[4] POWER, J., AND SHKARAVSKA, O. From comodels to coalgebras: state and arrays. In *Proceedings of the Workshop on Coalgebraic Methods in Computer Science* (2004), vol. 106 of *Electronic Notes in Theoretical Computer Science*, Elsevier, pp. 297–314.