Coalgebraic Trace Semantics for Probabilistic Processes
Preliminary Proposal

Lawrence S. Moss
Indiana University
lsm@cs.indiana.edu

Chung-chieh Shan
Indiana University
ccshan@indiana.edu

Alexandra Silva
University College London
alexandra.silva@ucl.ac.uk

October 16, 2015

This proposal pertains to the *denotational semantics* of probabilistic processes. To get to the matter quickly, here some examples which interest us.

(A) Random walk with real outputs. Consider a measurable space M, and a probability measure π on it. Consider also a process P which starts with some $r_0 \in [0,1]$ visits the points of M according to π and output a real number in $[0,1]$. Running P intuitively produces a stream $r_0, r_1, r_2, \ldots$ of real numbers. We take it as given that probalistic programming languages would be capable of writing a program for P and that a basic question in the field would be to compute and work with "semantics" $[\![P]\!]$ in some useful way. Notice each run of P would produce different numbers, depending on π, of course. It is reasonable to assume that $[\![P]\!]$ would be a *probability measure* on sequences from $[0,1]$.

(B) A higher-order version of (A). Let us vary (A) by allowing π to change as time goes on, and also to depend on the point. So in addition to M, we begin with a space S of probability measures on M, and also with a probability measure μ on $S \times [0,1]$ and a starting real number r_0. From r_0, we use μ to get some π_0. From π_0, we get r_1. We keep going like this. Call this process Q. It is reasonable to assume that $[\![Q]\!]$ would be a measure on streams from $[0,1]$, just as $[\![P]\!]$ was.

(C) A formal language version of (A). Using a probabilistic context-free grammar of some natural language, we generate sentences.

(D) Games of imperfect information. Think of a two-player game where the players have probabilistic knowledge of each other and where they update this knowledge and use to formulate and modify probabilistic strategies that have probabilistic payoffs at various points.

Coalgebras and final coalgebras We start on the semantics of (A) - (D) by recalling that many state-based dynamical systems of interest in computer science are captured as *coalgebras* in some way or other (see [**?**]). To have a coalgebra, one needs a category C and a functor $F : C \to C$. In most cases, C is Set. A coalgebra is a pair $(c, f : c \to Fc)$ of an object in C together with a morphism in C that captures the type dynamics of the coalgebra. One can think of an arbitrary coalgebra as a specification of a process. Then the *final coalgebra*, if it exists, is a kind of semantic space where behaviors of individual coalgebras can be uniquely found. This is not the formal definition, of course.

Here are some examples, again from Set. These are supposed to be relevant to *non-probabilistic versions* of the examples (A) - (D) above. In (A), we have a set X playing the role of M, an element $x \in X$, and a function $f : X \to X$ for the dynamics. Forgetting x_0 for a moment, we have $X \to [0,1] \times X$ representing the output and the dynamics. Abstracting, we have a functor $F : \mathsf{Set} \to \mathsf{Set}$ given by $X \mapsto [0,1] \times X$. Then a deterministic process with real outputs is a coalgebra for this F. The final coalgebra would in this case be the set $[0,1]^\infty$ of infinite sequences of reals. As a coalgebra, it is $[0,1]^\infty \to [0,1] \times [0,1]^\infty$ given by $(\mathsf{head}, \mathsf{tail})$. The map from our example to the final coalgebra would take a starting state x_0 to the sequence out outputs which we would get by following our dynamics f. Similar abstractions can be associated to the examples (B) - (D), where in (D) we would need pairs of sets.

(A) - (D) themselves can be understood as *functors on the category* Meas *of measurable spaces*. The functor $\Delta : \mathsf{Meas} \to \mathsf{Meas}$ assigns to each Meas-space M the space of probability measures on M. This extends to the well-known *Giry monad*. *Measure polynomial functors (MPFs)* are the smallest class of functors on Meas containing the identity, constants M and closed under products, coproducts, and Δ. (A) - (C) are basically coalgebras of MPFs.

Can we find reasonable semantics into the final coalgebra? This raises the question of whether the final coalgebra exists in the first place. Yes: [**?**] shows that *every MPF has a final coalgebra*. The construction builds on *a probabilistic modal logic* $\mathcal{L}_F$ *associated to each MPF* F. Intuitively, each state s in each coalgebra has a *theory*: the set of sentences in $\mathcal{L}_F$ that hold of s. Then the set of all possible theories turns out to be the final coalgebra. This is not a trivial result; it extends results from economics/game theory on *Harsanyi type spaces*; these are the final coalgebra of a functor related to our game theory example (D).

The problem: the final coalgebra described in this way is (sadly) far from the intended semantics. In cases (A) - (C), we get a measurable space of all theories, not a space of sequences. Here is what we aim to show. Let $F : \mathsf{Meas} \to \mathsf{Meas}$ have a final coalgebra νF. (Think of $FX = M \times X$ for a fixed M, as in (A).) Suppose that there is a distributive law $\lambda : FM \to MF$ (a mild assumption). Then every coalgebra $X \to \Delta FX$ has a semantic map $[\![f]\!]$ into the space $M(\nu F)$ making the diagram below commute:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "{[\![f]\!]}"'] & MFX \arrow[r, "{MF[\![f]\!]}"] \arrow[d, "{MF[\![f]\!]}"] & MFMF(\nu F) \arrow[d, "MF\lambda"] \\
M(\nu F) & MF(\nu F) \arrow[l] & MMF(\nu F) \arrow[l]
\end{tikzcd}$$

In (A), for example, we would get a probability measure on sequences (what we want). [**?**] has some work on this kind of semantics, but it doesn't cover our examples (A) - (D).

References

[KK] Henning Kerstan and Barbara König, Coalgebraic trace semantics for continuous probabilistic transition systems. Logical Methods in CS, Vol. 9(4:16)2013, 134.

[MV] Lawrence S. Moss and Ignacio Viglizzo. *Final coalgebras for measurable spaces.* Information and Computation 4 (204), 2006 (610–636).

[R] Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.