Finitary Traces in Two-Player Games

Benjamin Plummer

University of Southampton, United Kingdom
bjp1g19@soton.ac.uk

A variety of system types related to computer science are naturally modelled coalgebraically, where final coalgebras provide a description of branching-time semantics. Some work has focussed on linear-time behaviour where states are assigned a collection of traces through the system. Here monads are a core component used to flatten multiple steps into one. Monads in general fail to compose, so traces for two-player system-versus-environment games have so far escaped attention. An example are coalgebras for the double covariant powerset functor $PP : \mathbf{Set} \to \mathbf{Set}$; with there being no way to equip PP with monad structure [8]. In this work we have two monads S and T modelling the system and environment respectively; we think of the system as picking moves and the environment as nondeterministically branching. We use weak distributive laws [3, 4, 1] to reverse environment-then-system branching by picking a one-step strategy for the system, resulting in a composite monad $\overline{ST} : \mathbf{Set} \to \mathbf{Set}$. We isolate conditions when the Kleisli [5] approach and the EM approach [7] can be applied for defining finite traces for reactive and generative systems respectively.

Returning to double powerset, the neighbourhood monad $(\mathcal{N}, \mu^{\mathcal{N}}, \eta^{\mathcal{N}}) : \mathbf{Set} \to \mathbf{Set}$ does exist (when the powersets are contravariant), however the monad multiplication appears too cut-throat: for example $\{\{\{\{x_1\}\}\}, \{\{\{x_2\}\}\}\} \overset{\mu^{\mathcal{N}}}{\mapsto} \emptyset$. According to our intuition, the system can "pick" either $\{\{\{x_1\}\}\}$ or $\{\{\{x_2\}\}\}$ with the remaining choices being fixed, determining the result $\{\{x_1\}, \{x_2\}\}$. One fix is to restrict to the monotone neighbourhood functor, where sets of sets are forced to be upwards closed, and the approach using weak laws is closely related. The multiset monad M is used to model system choices with the full law $MP \to PM$, which maps $[\![U_1, \dots, U_n]\!] \mapsto \{[\![x_1, \dots, x_n \mid x_i \in U_i]\!]\}$. We think of ending up nondeterministically in $U_1, \dots, U_n$, and the system picking a "one-step strategy" x_i from each U_i, forcing a play into states $x_1, \dots, x_n$. This intuition slightly loosens when looking at non-full laws, for example $\lambda^{\overline{PP}} : PP \to PP$ is defined on an element $\mathcal{X} \in PP(X)$, with $\lambda(\mathcal{X}) = \{V \subseteq \bigcup \mathcal{X} \mid \forall U \in \mathcal{X} : U \cap V \neq \emptyset\}$. This corresponds to introducing some nondeterminism into one-step system strategies. Consider $\{\{x_1\}, \{x_2, x_3\}\} \overset{\lambda^{\overline{PP}}}{\mapsto} \{\{x_1, x_2\}, \{x_1, x_3\}, \{x_1, x_2, x_3\}\}$, in the final option $\{x_1, x_2, x_3\}$ the system has essentially chosen x_2 and x_3 in combination. Below we collect examples of (weak) distributive laws from [1, 4, 3], M_S is the semiring monad for some semiring S, and D is the finite distribution monad. These laws all have the flavour just discussed, with $DP \to PD$ including convex combinations of system choices, essentially corresponding to randomised strategies.

Full Laws	**Weak Monotone Laws**
$MP \to PM$	$M_S P \to PM_S$ for a positive semifield S
$MM \to MM$	$M_S P_f \to P_f M_S$ for a positive semifield S
	$PP \to PP$
	$DP \to PD$

Generative Systems These are systems which generate output. In our setting, we have coalgebras of the shape $X \to \overline{ST}H(X)$. A standard approach to giving finitary trace semantics is coinduction in the Kleisli category of the monad [5]: requiring that H distributes over $\overline{ST}$. Importantly, none of the above laws result in composite monads which are commutative (they are not laws of commutative monads in the sense of [6]); however their component monads are, yielding canonical distributive laws $\alpha : HS \to SH$ and $\beta : HT \to TH$. We can build a law $H\overline{ST} \to \overline{ST}H$ compositionally from α and β, when H belongs the class of functors **GBF** (for generative behaviour functors), defined below.

$$H := \mathrm{id} \mid A \times \mathrm{id} \mid \coprod H \mid H \circ H$$

The weak laws we have are laws of strong monads (from [6]), meaning the equation $\lambda \circ S(\mathrm{str}^T) \circ \mathrm{str}^S = T(\mathrm{str}^S) \circ \mathrm{str}^T \circ \mathrm{id} \times \lambda$ holds, where str are the monads strength maps.

Proposition 1 *Given a weak distributive law of strong monads $\lambda : TS \to ST$ where S and T are commutative, and a functor H from* **GBF***, the following Yang-Baxter equation holds between the canonical laws α, β and the (weak) law λ.*

$$\begin{array}{ccccc} HTS & \xrightarrow{\beta S} & THS & \xrightarrow{T\alpha} & TSH \\ \downarrow H\lambda & & & & \downarrow \lambda H \\ HST & \xrightarrow{\beta S} & THS & \xrightarrow{T\alpha} & STH \end{array}$$

The Yang-Baxter equation holding gives us a way to define a full distributive law $H(ST) \to (ST)H$ when λ is full, and has been adapted to the weak setting in Theorem 4.5 from [2]. It is then left to check the following conjecture, which is partly handled for the composite monad $M_S P \to PM_S$ in [1].

Conjecture 1 *For any of our (weak) laws, the Kleisli category $\mathbf{Set}_{\overline{ST}}$ is ω-cppo-enriched and when an H from* **GBF** *is lifted to $H_{\overline{ST}} : \mathbf{Set}_{\overline{ST}} \to \mathbf{Set}_{\overline{ST}}$ on the Kleisli category, it is locally monotone.*

We illustrate what the trace semantics will give us with $HX := 1 + A \times X$ with $\overline{PP}$. The initial algebra A^* is lifted to the final coalgebra in $\mathbf{Set}_{\overline{PP}}$ and we get a map, given a coalgebra $X \xrightarrow{c} \overline{PP}(1 + A \times X)$, assigning each state to its trace semantics $\mathsf{tr} : X \to \overline{PP}(A^*)$. Each element $U \in \mathsf{tr}(x)$ is a set of traces which the system can force. There are certain elements in $\mathsf{tr}(x)$ which correspond exactly to a single system strategy, however other elements correspond to several strategies (or a single strategy where the system can make multiple choices).

Reactive Systems For reactive systems of the type $X \to G\overline{ST}(X)$, results from [7] can be applied. There is a distributive law $TG \to GT$ for any strong monad T (which **Set** monads automatically are), and any functor belonging to

$$G := \mathrm{id} \mid B \mid G \times G \mid G^A \mid G \circ G$$

where B has a T-algebra structure $b : T(B) \to B$. A determinisation procedure then lifts $c : X \to G\overline{ST}(X)$ into $\hat{c} : \overline{ST}X \to G(\overline{ST}X)$ where G's final coalgebra can be used to obtain trace semantics. For example, we can treat alternating automata as $2 \times \overline{PP}(X)^A$-coalgebras, and obtain a map $X \xrightarrow{\eta} \overline{PP}(X) \xrightarrow{\mathsf{beh}_{\hat{c}}} 2^{A^*}$, which gives the language where the system has a strategy to force an accept state. Note that states $\mathcal{X} \in \overline{PP}(X)$ of $\hat{c}$ are sets of states which the system can force, and accordingly the $\overline{PP}(2) \xrightarrow{b} 2$ structure is $b(\mathcal{X}) = 1 \iff \{1\} \in \mathcal{X}$, meaning we only accept when the system can force a set that consists of one or more accept states.

Future Work As future work we would like to link traces and strategies more closely. For example, in generative systems with the $\overline{PP}$ monad if $U \in \mathsf{tr}_c(x)$, can we extract a strategy which would guarantee those given traces? Conversely, we could ask is there such a U for every possible strategy? Once we have a suitable notion of strategy, can we define classes of strategies such as finite memory or memoryless, and then link these with different classes of properties (such as ω-regular)?

Another avenue of research we are keen to explore is give an account of infinite traces. These traces have important applications in model checking, where properties like "something bad will never happen" are used.

References

[1] F. Bonchi and A. Santamaria. Convexity via Weak Distributive Laws. 2022.

[2] A. Goy. Weakening and Iterating Laws using String Diagrams. 2023.

[3] A. Goy and D. Petrişan. Combining probabilistic and non-deterministic choice via weak distributive laws. 2020.

[4] A. Goy, D. Petrişan, and M. Aiguier. Powerset-Like Monads Weakly Distribute over Themselves in Toposes and Compact Hausdorff Spaces. 2021.

[5] I. Hasuo, B. Jacobs, and A. Sokolova. Generic Trace Semantics via Coinduction. 2007.

[6] B. Jacobs. Semantics of weakening and contraction. 1994.

[7] B. Jacobs, A. Silva, and Ana Sokolova. Trace semantics via determinization. 2015.

[8] B. Klin and J. Salamanca. Iterated Covariant Powerset is not a Monad. 2018.