Traces via Strategies in Two-Player Games

Benjamin Plummer[a,1,3] Corina Cîrstea[a,2,4]

[a] *School of Electronics and Computer Science*
University of Southampton
United Kingdom

Abstract

Traces form a coarse notion of semantic equivalence between states of a process, and have been studied coalgebraically for various types of system. We instantiate the finitary coalgebraic trace semantics framework of Hasuo et al. for controller-versus-environment games, encompassing both nondeterministic and probabilistic environments. Although our choice of monads is guided by the constraints of this abstract framework, they enable us to recover familiar game-theoretic concepts. Concretely, we show that in these games, each element in the trace map corresponds to a collection (a subset or distribution) of plays the controller can force. Furthermore, each element can be seen as the outcome of following a controller strategy. Our results are parametrised by a weak distributive law, which computes what the controller can force in a single step.

Keywords: Two-player game, Markov decision process, coalgebra, trace semantics, strategy

1 Introduction

The problem of program synthesis can be phrased using a two-player game between a controller and its environment: where the controller must achieve some linear-time property irrespective of environment choices. We give a *coalgebraic* framework for game-based synthesis, which allows us to uniformly treat modelling the environment as nondeterministic or stochastic.

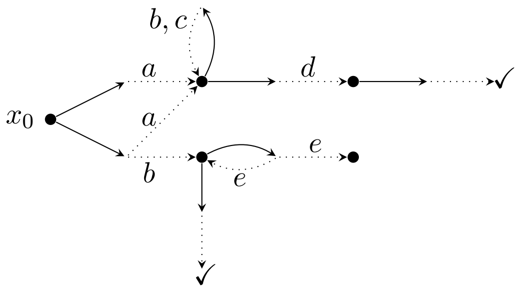
The process-theoretic notion of *trace* is well studied in program semantics. A trace arises from a sequence of choices a system can make, and records the observable behaviour of the resulting execution. *Plays* in two-player games are more general: players make interleaved choices and have opposing objectives. We work with an elegant coalgebraic representation of two-player games, where each player is modelled with a monad. Combining the controller and environment monads with a weak distributive law, we obtain a monad for the composite system, to which the general coalgebraic theory of finite traces [17] is shown to apply. We then prove a close connection between controller strategies and traces, phrasing game-theoretic notions like plays and strategies categorically along the way. Our parametric approach handles non-deterministic and probabilistic environments uniformly, with two different monads.

We focus on two-player games where the observable outcome of a completed play consists of a finite sequence of basic observations, each arising after a simple controller-environment interaction. To

[1] Email: bjp1g19@soton.ac.uk
[2] Email: cc2@ecs.soton.ac.uk
[3] The first author would like to thank Alexandre Goy for enlightening discussions about the contents of his thesis.
[4] Cîrstea was funded by a Leverhulme Trust Research Project Grant (RPG-2020-232).

illustrate, consider the game on the following page, made up of controller states (thick black dots), controller transitions (solid arrows), and environment transitions (dotted arrows). A single observation from $A = \{a, b, c, d, e\}$ is output after each transition (when a play has not terminated), and an observation from $B = \{\checkmark\}$ is output when a play terminates. We have chosen to model the environment as non-deterministic in this example.

A one-step controller-environment interaction is viewed as a single transition. First, the controller chooses an environment state, then the environment picks a pair of observation from A and controller state, or to terminate with an observation from B. Each completed play from a state can be associated with a *trace*: a sequence of observations from A ending with termination and final observation from B. Our main theorem shows that the trace semantics of a state, obtained by instantiating the coalgebraic theory of finite traces to our monad, gives the set of subsets of traces which the controller can force. For example, from x_0, the subsets of traces the controller can force include $\{b\checkmark, ad\checkmark\}$, $\{abd\checkmark, acd\checkmark\}$ and $\{b\checkmark, acd\checkmark, abcd\checkmark, abbcd\checkmark, abbbcd\checkmark, abbbbd\checkmark\}$, corresponding to history-dependent controller strategies which force them. When the environment is modelled probabilistically, the subsets of enforceable traces become distributions of traces that the controller can force.



In coalgebraic trace semantics, the type of system under consideration is modelled with a monad (describing the computation type, e.g. non-deterministic or probabilistic) composed with an endofunctor (describing the type of observations made along a trace). We begin, following work on alternating automata [12,14], by identifying a composite monad built from the weak distributive law of powerset over itself as a suitable model for controller-environment interactions in two-player games. We then refine our component monads - to satisfy the requirements of the finite trace semantics theorem in [17] - to full powerset for controller choices and finite, non-empty powerset for environment choices. In the process of examining the conditions of [17], we reveal two mistakes in the literature: namely that composition in the Kleisli category of the monad featured in [3] is not left-strict; and that the monad in [23] is not commutative. Section 4 proceeds by proving the conditions required to obtain a trace map, and then introduces *execution maps* (where executions are the coalgebraic counterpart of game plays) as a special kind of trace map. The required conditions also hold when using full powerset for controller choices, and the finite distribution monad for environment choices, giving us a second example where the environment evolves probabilistically. Finally, we show that the trace map factors through the execution map.

A standard definition of a player strategy in a two-player game, is a function from *partial plays* ending in a state controlled by that player, to a valid move for that player. Such strategies need only be defined over partial plays which they can force, so are partial functions. In Section 5, we give an equivalent definition of a controller strategy as a chain of maps in the Kleisli category of the monad modelling the environment. Then, in Section 6, we characterise the trace map in terms of collections (subsets or distributions) of finite traces that a controller strategy can force. We show that a collection of traces is in the image of some state x under the trace map if and only if there exists a strategy from x forcing exactly that collection. We see this as a direct result of the underlying weak distributive law computing one-step outcomes of strategies.

Our general goal is to develop theory to underpin a general coalgebraic synthesis tool. The main result in Section 6, confirms we have found the right monad for representing two-player games for controller synthesis. The practical contribution of Theorem 4, is that outcomes of strategies in a game, which are crucial in controller synthesis, are the finite trace semantics of said game, so can be computed with a least fixed-point computation. Our choice to model a non-deterministic environment using the *non-empty* finite powerset, also makes sense from a practical standpoint: a real environment can never deadlock, and it is common practice to restrict the possible environment inputs to be finite [5] to make the synthesis problem decidable. The composite monad will give *convex* sets of collections (subsets or distributions), meaning that sets of subsets are closed under binary union, and sets of distributions are closed under convex combination. In finitely branching games, with a non-deterministic environment, this yields a greatest choice at each controller state, the union of all the other choices. This can allow strategy synthesis by approximating down

[5] A similar assumption is made in geometric logic [29], where there is arbitrary disjunction and finite conjunction.

from the largest strategy, yielding the most permissive strategy. When the environment is probabilistic, the convexity requirement allows *randomised strategies*, which are standard in the theory of Markov Decision Processes [7], to also be accounted for.

Finally, we comment on the role of category theory in this paper. Our main result, which equates finite trace semantics with outcomes of strategies, cannot be known explicitly, as there is no definition of trace semantics of two-player games in the literature. What our coalgebraic approach gives us, is a ready-made definition of finite trace semantics: it is a fact (established in Section 4) that there is a final coalgebra in the category where our games live. Thus, the role of coalgebra, and more generally category theory, is pivotal in our work. In Section 5, we give a category-theoretic definition of strategies and plays, which is not only enlightening, but also provides a clear road map for the proof in Section 6.

1.1 Related Work

A composite monad which we do not consider, but is conceptually very similar, is full powerset combined with full finite powerset. This monad is a special case of the monad considered in [3], which combines full powerset with a *multiset monad over a semifield*, when we choose the Boolean semifield. However, as we will show, this monad is inadequate for trace semantics given by Theorem 3.6. Similarly, the monad considered in [19] could instantiate to non-empty powerset combined with finite powerset. This is also not suitable for Theorem 3.6, however is adapted to work in [19]. In doing so, they require an additional assumption which, when expressed in terms of games, amounts to the controller always being able to force the environment to deadlock immediately. This would be unsatisfactory in controller-versus-environment games, as the controller would always have a (trivial) winning strategy.

When looking for monads on the double powerset functor, the *neighbourhood monad* $\mathcal{N}$ comes to mind (generated from the contravariant powerset being dual adjoint to itself). The multiplication $\mu^{\mathcal{N}}$ does not model games, for example it only returns a non-empty set when given a subset containing an upwards closed set. Similarly, double covariant powerset on a function always returns upwards closed sets. To fix these oddities, a common choice for work on game logic [15,16] is to use the *monotone neighbourhood monad* $\mathcal{M}$; it has been suggested in [24] to use $\mathcal{M}$ to give a path-based semantics for a coalgebraic CTL. This monad is generated by restricting the dual adjunction between **Set** and itself to one between **Set** and **Poset** (see e.g. [20]) - and was derived independently in [2] in the context of alternating automata, using a distributive law. We view the monotone neighbourhood monad as "fixing" the neighbourhood monad because the contravariant nature is tamed: on upward closed sets $PP(f)$ agrees with $\mathcal{N}(f)$ and the multiplication agrees with the $\exists\,\forall$ behaviour of picking a strategy [6]. This monad is also not suitable for the assumptions required in Theorem 3.6, because its Kleisli category is not enriched in ω-cpos (see Counterexample 1). It may also be possible, analogous to the approach taken in this paper, to restrict to upwards closed sets of *finite* subsets, we leave investigating this to future work.

In [4], the authors consider a non-deterministic programming language, where game-like behaviour arises from I/O interaction. The key difference with our work, is that the input to the program is part of the (branching-time) behaviour - captured with a functor rather than a monad. The non-determinism of the program, is treated with the powerset monad, thus the trace semantics is a set of possible computations (rather than a set of subsets, like in ours).

Finally, while we have chosen to work with the Kleisli approach to finite trace semantics from [17], it is worth mentioning other approaches and variants. Our games are generative (of the shape TF), which rules out the Eilenberg-Moore approach (which treats systems of type GT). In [22], the authors give a way of casting the Kleisli approach to the Eilenberg-Moore one, using an natural transformation $TF \to GT$. We find it unlikely that a suitable functor G and natural transformation exist in our case. Work in [8] provides a different set of assumptions to obtain finite trace semantics, in regards to our work the only assumption that makes a difference is they do not assume a zero object in the Kleisli category of the monad, however they do still require the stronger condition of left-strictness (see [17, Lemma 3.5]). It may be possible to phrase our work in terms of graded monads [27]. Although, as we currently rely on structure in the Kleisli category of our monad (e.g. ω-cpo enrichedness and the existence of a certain limit) which is afforded to us by the approach in [17,21], hence we leave a treatment in terms of graded monads to future work.

[6] The monotone neighbourhood monad is not a submonad of $\widetilde{PP}$, because the units do not agree.

2 Outline

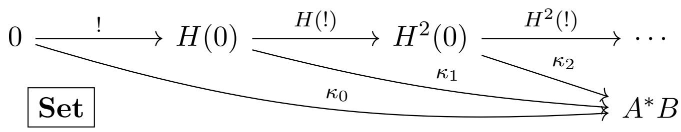
We now sketch the categorical ideas of our approach. Assume we model the controller with the full covariant powerset monad $P : \mathbf{Set} \to \mathbf{Set}$, and the environment with some monad $T : \mathbf{Set} \to \mathbf{Set}$. Suppose we have some way to combine these into a monad $\widetilde{PT} : \mathbf{Set} \to \mathbf{Set}$, which fits the framework for finite trace semantics in [17]. We model two-player games as $\widetilde{PT}H$ coalgebras, with a functor $H : \mathbf{Set} \to \mathbf{Set}$ describing the possible observations after one interaction. We will take $H(X) = B + A \times X$, focussing on games whose plays can terminate with an observation $b \in B$ or proceed to a new state $x \in X$ with an observation $a \in A$, following a controller-then-environment choice of moves. Instantiating [17], we get a trace map $\mathsf{tr}_c : X \to \widetilde{PT}(A^*B)$ for each coalgebra $c : X \to \widetilde{PT}H(X)$, where A^*B is the initial H-algebra. We answer two questions:

(i) There are no established notions of trace semantics in games, so what do the contents of $\mathsf{tr}_c(x)$ correspond to?

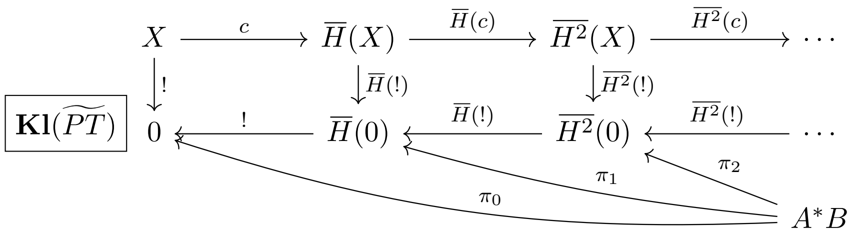
(ii) How do strategies fit into the categorical picture?

We will summarise our answers shortly. First, fix a coalgebra $c : X \to \widetilde{PT}H(X)$, that we view as a game. The states $x \in X$ correspond to controller states, whereas elements $U \in c(x) \subseteq T(AX + B)$ are collections of observations from one-step plays which the controller can force from x. These elements can be thought of as environment states. Similarly, iterating (in the Kleisli category of the monad $\widetilde{PT}$) the coalgebra gives a map $X \to \widetilde{PT}H^n(X)$, assigning each state to the set of collections of observations of n-step plays which the controller can force.

To answer the first question, we follow [17], and unpack the generic construction of the trace map. The colimit of the initial sequence of the endofunctor H gives the initial H-algebra A^*B. The elements of this algebra are the possible observable outcomes of plays, each given by some $\kappa_n(a_1 \dots a_{n-1}b)$ with $n \geq 1$:

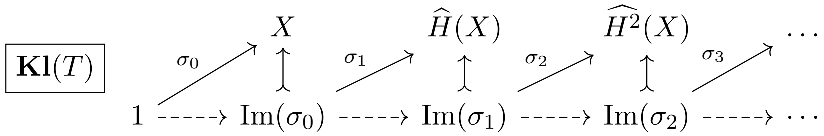


The trace map from [17] arises from the observation that the initial H-algebra is a final $\overline{H}$-coalgebra (where $\overline{H} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$ is the extension of $H : \mathbf{Set} \to \mathbf{Set}$).

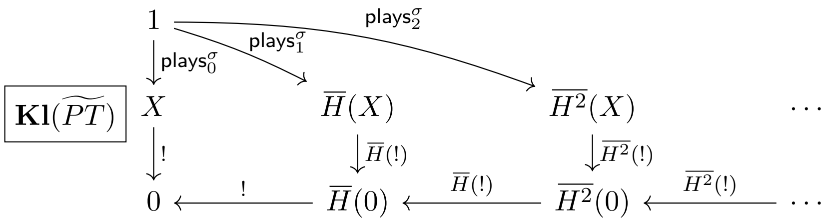


Specifically, iterating c up to some depth and then projecting into the final sequence of $\overline{H}$ yields a cone over this final sequence, and the map $\mathsf{tr}_c : X \to A^*B$ arises from the limiting property of A^*B in $\mathbf{Kl}(\widetilde{PT})$. For $x \in X$, each element of $\mathsf{tr}_c(x)$ is a collection of observations of completed plays, which the controller can force. This provides an answer to the first question.

For the second question, note that strategies (in the standard sense) resolve controller choices, so should exist in the Kleisli category of the monad T used to model environment choices. In Section 5, roughly speaking, we capture strategies using a family of maps $\sigma_{n+1} : \mathrm{Im}(\sigma_n) \to \widehat{H^{n+1}}(X)$ for each $n \in \omega$, depicted below.



Here, $\widehat{H}$ extends H to $\mathbf{Kl}(T)$. The map σ_0 picks the initial state of the play, whereas the maps σ_{n+1} pick subsequent controller moves from states that can be reached via $\sigma_0, \ldots, \sigma_n$, according to the allowed moves in those states (as specified by (X, c)). Just like $\overline{H^n}(X)$, the elements of $\widehat{H^n}(X)$ are either completed plays of length up to n or incomplete plays of length exactly n. By composing along the bottom of the above diagram we obtain maps $\mathsf{plays}^\sigma_n : 1 \to \widehat{H^n}(X)$, which give the set of complete plays (of length less than or equal to n) and partial plays (of length n) which the strategy σ can force. We show in Section 5 that we can lift these maps into $\mathbf{Kl}(\widetilde{PT})$, and they again form a cone over the final sequence.



This establishes a unique mediating map $1 \to A^*B$ in $\mathbf{Kl}(\widetilde{PT})$ for each strategy σ, which we see as the outcome of σ: which finite completed traces σ can force. We complete the answer to our second question in Section 6, by proving that the trace semantics $\mathsf{tr}_c(x)$ at a state x coincides with the union of all the finite observable outcomes of strategies from x. This gives us that there is a set $U \in \mathsf{tr}_c(x)$ if and only if, there is a strategy which forces a set of completed plays underlying the traces in U.

3 Preliminaries

3.1 Two-player Games

We start by introducing the kind of two-player game we are concerned with, and the subsequent game-theoretic concepts. These are represented coalgebraically in the remainder of the paper.

Fix two disjoint sets A and B, representing the continuing observations and terminating observations respectively. Our two-player games are player over *bipartite game graphs* (X, Y, E_1, E_2), where X and Y are disjoint sets of controller and environment states respectively, and $E_1 \subseteq X \times Y$ and $E_2 \subseteq Y \times (B + A \times X)$ are the controller and environment edge relations respectively. For technical reasons discussed in Section 4, we put two restrictions on the environment's edge relation E_2: that it is *image-finite* and *left-total*. Image-finiteness means $\{u \in B + A \times X \mid E_2(y, u)\}$ is finite for each environment state $y \in Y$. E_2 being left-total means for each environment state $y \in Y$, there exists some $u \in B + A \times X$ such that $E_2(y, u)$.

A *partial play* over a game graph is an element $x_0a_1x_1 \ldots a_nx_n \in (XA)^*X$ such that there is an environment state $y \in Y$ with $E_1(x_i, y)$ and $E_2(y, (a_{i+1}, x_{i+1}))$, for all $0 \leq i < n$. A *completed play* is an element $\rho x b \in (XA)^*XB$, such that ρx is a partial play and there exists an environment state $y \in Y$ with $E_1(x, y)$ and $E_2(y, b)$.

Let $\sigma : (XA)^*X \to Y$ be a *partial function* (i.e. not defined over the entire domain), which respects our game: $E_1(x, \sigma(\rho x))$ for every $\rho x \in (XA)^*X$ which σ is defined over. We say that a partial play $x_0a_1x_1 \ldots a_nx_n \in (XA)^*X$ *conforms to* σ, if for all $0 \leq i < n$: σ is defined over $x_0a_1x_1 \ldots a_ix_i$ and $E_2(\sigma(x_0a_1x_1 \ldots a_ix_i), (a_{i+1}, x_{i+1}))$. Similarly, a completed play $\rho x b \in (XA)^*XB$, made up of partial play ρx and a terminating observation b, *conforms to* σ, when ρx conforms to σ and $\sigma(\rho x)$ is defined with $E_2(\sigma(\rho x), b)$. We call a pair (x, σ) a *pointed standard strategy*, precisely when σ is defined **exactly** over the σ-conform partial plays which start in x. An *n-step partial outcome* of a pointed standard strategy (x, σ), denoted $\mathsf{plays}^\sigma_n(x)$, is the set of all partial plays of length n (elements of $(XA)^nX$) and of complete plays of length less than n (elements of $(XA)^{<n}XB$) which conform to σ. The *completed outcome* $\mathsf{plays}^\sigma(x) \subseteq (XA)^*XB$ of a pointed standard strategy (x, σ), is the set of all completed plays which conform to σ.

The objective of the game, from the point of view of the controller, is to force a completed play within a set of "good" outcomes, where "goodness" is a property of the *trace* $a_1 \ldots a_nb$ underlying a completed play $x_0a_1x_1 \ldots a_nx_nb$. The reader may have noticed that our notion of play differs slightly from the standard one [26], in that it does not record environment states. This is not an issue, precisely because the environment states visited along a play have no impact on whether the resulting outcome is good or not; and as a result, strategies can not benefit from recording environment states in the play history. The controller states also do not affect the goodness of an outcome, but they need to be explicit because the strategy may depend on them.

3.2 Markov Decision Processes

We also consider a probabilistic variant of these games, which are essentially *Markov decision processes* (MDPs). These games are still four-tuples (X, Y, E_1, E_2), but now E_2 is a function $Y \times (B + A \times X) \to [0, 1]$ such that $E_2(y) : B + A \times X \to [0, 1]$ is a finitely supported probability distribution.

Here, a *partial play* is an element $x_0 a_1 x_1 \dots a_n x_n \in (XA)^* X$ such that for all $0 \leq i < n$, we have that there exists some $y \in Y$ with $E_1(x_i, y)$ and $E_2(y, (a_{i+1}, x_{i+1})) > 0$. A *completed play* is a sequence $\rho x b \in (XA)^* XB$, such that ρx is a partial play, and there exists a $y \in Y$ with $E_1(x, y)$ and $E_2(y, b) > 0$.

Let $\sigma : (XA)^* X \to Y$ be a partial function, which respects our MDP: $E_1(x, \sigma(\rho x))$ for every $\rho x \in (XA)^* X$ which σ is defined over. A partial play $x_0 a_1 x_1 \dots a_n x_n$ *conforms to* σ, if for all $0 \leq i < n$ we have $E_2(\sigma(x_0 a_0 x_1 \dots a_i x_i), (a_{i+1}, x_{i+1})) > 0$. Again, a completed play $\rho x b$ *conforms to* σ, if ρx is a partial play which conforms to σ, and $E_2(\sigma(\rho x), b) > 0$. We call a pair (x, σ) a *pointed standard strategy*, precisely when σ is defined **exactly** over the σ-conform partial plays which start in x. The n-step partial outcome and the n-step completed outcome, then become distributions over partial and completed plays. We leave the definitions to the reader.

3.3 Linear Functors

We require a restriction of set-based polynomial functors (which are standard in coalgebra, see [21, p. 49]) to *linear functors*. To reduce clutter, we often use juxtaposition to denote the product of functors (we never need to denote the application of a constant functor), e.g. $AB := A \times B$. Technically, linear functors are a class of functors built inductively out of arbitrary coproducts and 1, a constant functor assigning every set to a singleton set. Linear functors have a general form $H(Y) \cong A + BY$, a consequence of $C(A + BY) \cong CA + CBY$ and $(A_0 + B_0 Y) + (A_1 + B_1 Y) \cong (A_0 + A_1) + (B_0 + B_1) Y$ (extended to arbitrary coproducts). We reserve H for a linear functor, and use H_X for the linear functor $H_X(Y) = X \times H(Y)$.

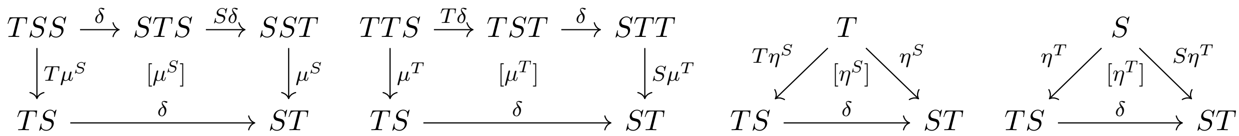
Shorthand $Y^n \cong Y^{\{0,\dots,n-1\}} \cong \prod_{0 \leq i \leq n-1} Y$ is used for lists of elements of Y of length n. Similarly, we use $Y^{<n} \cong \coprod_{0 \leq i < n} Y^i$ for lists of elements of Y of length less than n. Finally, we use $Y^* = \coprod_{n \in \omega} Y^{<n}$ for finite lists. We record the n-fold compositions and initial algebras of H and H_X in the table below.

F	$F^n(Y)$	Initial algebra
H	$A^{<n} B + A^n Y$	$A^* B$
H_X	$(XA)^{<n} XB + (XA)^n Y$	$(XA)^* XB$

3.4 Distributive Laws

We assume knowledge of the definition of a monad, whose multiplication and unit, with functor part T, are referred to as μ^T and η^T. Let P, P^+, Q denote the full powerset monad, the non-empty powerset monad, and the non-empty finite powerset monad respectively. We have submonads $Q \rightarrowtail P^+ \rightarrowtail P$. Let D denote the finite distribution monad, mapping a set X to the set of distributions over X with finite support. We will the monad morphism $\mathsf{supp} : T \to P$, for $T = Q, D$, which is an inclusion when $T = Q$, and maps a distribution to its *support* when $T = D$.

Definition 3.1 Given two monads $S, T : \mathbb{C} \to \mathbb{C}$, a *distributive law* of T over S is a natural transformation $\delta : TS \to ST$ such that



commute. A *weak distributive law* of T over S is a natural transformation $TS \to ST$ such that $[\mu^S, \mu^T, \eta^S]$ hold. Note that any distributive law is a weak distributive law. A *functor-monad distributive law* of a functor F over a monad S is a natural transformation $FS \to SF$ such that $[\mu^S, \eta^S]$ (substituting T for F) hold. In this paper, we refer to a functor-monad distributive law $FS \to SF$ as a distributive law when we are not considering any monad structure on F.

We also require strength maps for a monad. Given a monad $T : \mathbf{Set} \to \mathbf{Set}$, the *left strength map* is a functor-monad distributive law $\mathsf{stl}^T_{A,X} : A \times T(X) \to T(A \times X)$ between $A \times (-)$ and T which interacts well with the coherence isomorphisms for $\times$. This map must exist and is unique in **Set**. There is also a unique *right strength map* $\mathsf{str}^T_{X,A} : T(X) \times A \to T(X \times A)$, defined analogously. A monad T is *commutative* when the following holds for all sets X and Y:

$$\mu^T_{X\times Y} \circ T(\mathsf{str}_{X,Y}) \circ \mathsf{stl}_{TX,Y} = \mu^T_{X\times Y} \circ T(\mathsf{stl}_{X,Y}) \circ \mathsf{str}_{X,TY} \tag{1}$$

Example 3.2 This paper is concerned with the following distributive laws.

(i) There is a weak distributive law (first appearing in [9]) $\delta^{PP} : PP \to PP$:

$$\delta^{PP}(\{U_i\}_{i\in I}) := \{\bigcup_{i\in I} V_i \mid \forall i \in I\, (\emptyset \subset V_i \subseteq U_i)\}$$

that restricts to a law $\delta^{PP^+} : P^+P \to PP^+$. There is a variant $\delta^{PQ} : QP \to PQ$, which only takes finite subsets [12]:

$$\delta^{PQ}(\{U_i\}_{i\in I}) := \{\bigcup_{i\in I} V_i \mid \forall i \in I\, (\emptyset \subset V_i \subseteq_\omega U_i)\}$$

(ii) There is a weak distributive law $\delta^{PD} : DP \to PD$, distributing probability over nondeterminism [13]:

$$\delta^{PD}([U_i \mapsto p_i]_{i\in I}) := \{\mu^D_X[\varphi_i \mapsto p_i] \mid \forall i \in I\, (\varphi_i \in D(X) \text{ and } \mathsf{supp}\,\varphi_i \subseteq U_i)\}$$

(iii) We have already noted that str is a functor-monad distributive law. The right strength is defined w.r.t. the monoidal product $(\times, 1)$ in **Set**. We can define another strength map w.r.t. the monoidal product $(+, 0)$ in **Set**:

$$[\eta^T_{B+X} \circ \mathsf{inl}, T(\mathsf{inr})] : B + T(X) \to T(B + X)$$

Composing these functor-monad distributive laws we get a distributive law $\lambda_X : A + B(T(X)) \to T(A + B(X))$, i.e. linear functors distribute over any **Set** monad.

3.5 Composite Monads

Any weak distributive law induces a composite monad [9]. To combine two **Set** monads S and T with $\delta^{ST} : TS \to ST$, we define the associated *convex closure operator* [3]: a natural transformation $\mathsf{cl}^{ST}_X : ST(X) \to ST(X)$, defined by $S\mu^T \circ \delta^{ST}T \circ \eta^T ST$. The functor part of the composite monad is the image of the convex closure operator, denoted by $\widetilde{ST}$. The multiplication and unit of the composite, relying on [12, Lemma 2.10, p. 48], can be given as the standard expressions $\mu^S\mu^T \circ S\delta^{ST}$ and $\eta^S\eta^T$. When δ^{ST} is a (full) distributive law, $\widetilde{ST} \cong ST$.

Example 3.3 The closure operator associated with δ^{PQ} can be calculated:

$$\mathsf{cl}^{PQ}_X(\mathcal{U}) = \{\bigcup \mathcal{V} \mid \emptyset \subset \mathcal{V} \subseteq_\omega \mathcal{U}\}$$

It closes a set of finite subsets under finite, non-empty union. We call a set of finite subsets $\mathcal{U}$ *convex* precisely when $\mathsf{cl}^{PQ}_X(\mathcal{U}) = \mathcal{U}$, i.e. when $U, V \in \mathcal{U} \implies U \cup V \in \mathcal{U}$. The functor part $\widetilde{PQ}(X)$ is the set containing all convex sets of subsets. The closure operator for δ^{PD} is given by

$$\mathsf{cl}^{PD}_X(U) = \{\mu^D(\Phi) \mid \Phi \in DD(X),\, \mathsf{supp}\,\Phi \subseteq U\})$$

Similarly, we call a set U of distributions *convex* precisely when $\mathsf{cl}^{PD}_X(U) = U$. $\widetilde{PD}(X)$ is the set of all convex sets of distributions. Both of these closure operators satisfy standard closure operator properties: $\mathcal{U} \subseteq \mathsf{cl}(\mathcal{U})$, $\mathcal{U} \subseteq \mathcal{U}' \Rightarrow \mathsf{cl}(\mathcal{U}) \subseteq \mathsf{cl}(\mathcal{U}')$ and $\mathsf{cl}(\mathcal{U}) \subseteq \mathsf{cl} \circ \mathsf{cl}(\mathcal{U})$.

Remark 3.4 The category of algebras associated with the monad $\widetilde{PQ}$ is a complete lattice $(X, \bigvee)$ with a meet semi-lattice (without top) $(X, \wedge)$, such that $x \wedge \bigvee_i y_i = \bigvee_i (x \wedge y_i)$ holds, i.e. $\wedge$ distributes over $\bigvee$. Similarly, the algebraic theory of the monad $\widetilde{PD}$ is a complete lattice $(X, \bigvee)$ and a convex algebra $(X, +_r)$ such that $+_r$ distributes over $\bigvee$. This is discussed in [13, Section 3.2.2].

3.6 Kleisli Categories

Given a monad $T : \mathbb{C} \to \mathbb{C}$, we can form its Kleisli category, denoted $\mathbf{Kl}(T)$, which has the objects of $\mathbb{C}$ and homsets $\mathbf{Kl}(T)(X, Y) := \mathbb{C}(X, T(Y))$. Kleisli composition $\odot$ is defined by composition with μ^T: $g \odot f := \mu^T \circ T(g) \circ f$, and has the identity η^T_X for each object X. We have an identity-on-objects functor $J_T : \mathbb{C} \to \mathbf{Kl}(T)$, defined as $J_T(f) = \eta^T \circ f$, which is left adjoint to the forgetful functor $\mathbf{Kl}(T) \to \mathbb{C}$. We denote $J_T(f)$ as $\overline{f}$. An *extension* of $F : \mathbb{C} \to \mathbb{C}$ to $\mathbf{Kl}(T)$ is a functor $\overline{F} : \mathbf{Kl}(T) \to \mathbf{Kl}(T)$ such that $\overline{F} J_T = J_T F$. Kleisli extensions are in one-to-one correspondence with functor-monad distributive laws (see e.g. [19, Chapter 5]). Given a distributive law $\lambda : FT \to TF$, we can define $\overline{F}(f) := \lambda \circ F(f)$. We often require extensions of iterated functors in this paper, so highlight that extension distributes over functor composition: $\overline{F^n} = \overline{F}^n$. We also have $\overline{F}(\overline{f}) = \overline{F(f)}$.

3.7 Coalgebraic Traces

A *finite trace* is an element of the initial H-algebra A^*B, i.e. a finite sequence of observations $a_1 \dots a_n b$ with $a_i \in A$ and $b \in B$.

To state the main trace semantics theorem from [17], we require some domain-theoretic concepts. An *ω-complete partial order* (ω-cpo), is a partial order $(X, \leq)$ such that every ω-chain $x_0 \leq x_1 \leq \cdots$ has a join $\bigvee \{x_n\}_{n \in \omega}$. A category is ω-**cpo**-enriched when homsets can be equipped with ω-cpo structure, and composition preserves joins of ω-chains separately in both arguments. In an ω-**cpo**-enriched category, a morphism $i : X \to Y$ is an *embedding* precisely when there is a (necessarily unique) *projection* $p : Y \to X$ such that $p \circ i = \mathrm{id}_X$ and $i \circ p \leq \mathrm{id}_Y$ (where $\leq$ is the order on morphisms $X \to X$). We also need the concept of a *zero map* between objects X to Y, which exist in categories where there is a final and initial object 0. The zero map is the composite $X \xrightarrow{!} 0 \xrightarrow{!} Y$. Finally, a functor $F : \mathbb{C} \to \mathbb{D}$ between two ω-**cpo**-enriched categories is *locally monotone* when $f \leq g$ implies $F(f) \leq F(g)$.

We require two results: the first from work in domain theory [28], and the second which applies this result to obtain finite coalgebraic trace semantics [17]. Both of these results feature in [21, Chapter 5.3], which our formulation is based on, but phrased in terms of the slightly more general ω-cpos (following [17]) rather than dcpos, (the proofs in [21] only rely on joins of ω-chains).

Proposition 3.5 ([21, Proposition 5.3.3]) *Let $\mathbb{C}$ be an ω-**cpo**-enriched category with some ω-chain $X_0 \xrightarrow{i_0} X_1 \xrightarrow{i_1} X_2 \xrightarrow{i_2} \cdots$ of embeddings, with colimit $(A, \kappa_n : X_n \to A)$ in $\mathbb{C}$. Each coprojection κ_n is an embedding, call the associated projection $\pi_n : A \to X_n$. (A, π_n) is a limit of the ω-cochain of the projections $X_0 \xleftarrow{p_0} X_1 \xleftarrow{p_1} X_2 \xleftarrow{p_2} \cdots$. The mediating morphism of a cone $(f_n : Y \to X_n)_{n \in \omega}$ can be calculated with the join of an ω-chain of morphisms $Y \to A$:*

$$\bigvee_{n \in \omega} (Y \xrightarrow{f_n} X_n \xrightarrow{\kappa_n} A)$$

The following theorem, from [17], gives a categorical account of finite traces for systems modelled as coalgebras of certain functors on **Set**.

Theorem 3.6 ([21, Proposition 5.3.4]) *Given a monad $T : \mathbf{Set} \to \mathbf{Set}$ and a functor $F : \mathbf{Set} \to \mathbf{Set}$, if*

(i) *$\mathbf{Kl}(T)$ is ω-**cpo**-enriched*

(ii) *$T0 \to 1$ is an isomorphism*

(iii) *Zero maps are bottom elements in the Kleisli homsets*

(iv) *We have a distributive law $\lambda : FT \to TF$*

(v) *The extension $\overline{F} : \mathbf{Kl}(T) \to \mathbf{Kl}(T)$ is locally monotone*

(vi) *The initial F-algebra $F(I_F) \xrightarrow[\sim]{\alpha} I_F$ is the colimit of $0 \xrightarrow{!} F(0) \xrightarrow{F!} F^2(0) \to \cdots$*

Then J takes the initial F-algebra to a final $\overline{F}$-coalgebra, i.e. $J(\alpha^{-1}) : I_F \to \overline{F}(I_F)$ is final in $\mathbf{Coalg}_{\mathbf{Kl(T)}}(\overline{F})$.

In the above theorem, the monad T describes the type of computation (e.g. non-deterministic), whereas the functor F specifies what can be observed along single computation paths. The final $\overline{F}$-coalgebra gives us a unique $\overline{F}$-coalgebra morphism from any $\overline{F}$-coalgebra $c : X \to TF(X)$ into $J(\alpha^{-1}) : I_F \to TF(I_F)$. This coalgebra morphism $(X, c) \to (I_F, J(\alpha^{-1}))$ is usually referred to as the *trace map*, because it recovers the definition of finite traces in many different examples [17].

4 Traces and Executions

We now show that a two-player game modelled as a coalgebra $X \to \widetilde{PT}H(X)$, is susceptible to the coalgebraic framework for finite trace semantics given by Hasuo et al. [17]. We have two different choices for the monad T. We either take $T = Q$, which gives us something similar to the two-player games discussed in Section 3.1, or $T = D$, which give the MDPs from Section 3.2.

Thus, we must show that the requirements of Theorem 3.6 are met by the monads $\widetilde{PQ}$ and $\widetilde{PD}$, and a linear functor H. To motivate our choice of monads P and Q, and our restriction to linear functors, we also examine what doesn't work. Two of these failures exist, in some form, as mistakes in the literature. These are outlined in Counterexamples 2 and 3. Once we have established the existence of trace maps, we introduce *execution maps* (as a special type of trace maps) and show that every trace map factors through an execution map.

4.1 Kleisli Enrichment and Zero Maps

This subsection examines the requirements for Theorem 3.6 for the monad $\widetilde{PT}$, namely whether (i) $\mathbf{Kl}(\widetilde{PT})$ is ω-**cpo** enriched, (ii) $\widetilde{PT}(0) \cong 1$ and (iii) the zero map $X \to 0 \to Y$ in $\mathbf{Kl}(\widetilde{PT})$ is the least element in $\mathbf{Kl}(\widetilde{PT})(X, Y)$. When choosing a powerset monad T to model the environment non-deterministically, we explain why the finiteness and non-emptiness restrictions are required. To ensure (ii), we must choose a non-empty variant. Counterexample 1 exhibits that $\mathbf{Kl}(\widetilde{PP^+})$ is in fact not ω-**cpo** enriched (at least not in the natural way), meanwhile Proposition 4.1 shows that $\mathbf{Kl}(\widetilde{PQ})$ is. Hence, we are left with the choice of finite non-empty powerset Q to model a non-deterministic environment. For a probabilistic environment, the standard choice of the finite distribution monad D works out of the box: it already contains finite collections, and distributions summing to one mean they can't be "empty".

To show $\mathbf{Kl}(\widetilde{PT})$ is ω-**cpo** enriched, we follow the general approach in [3, Chapter 5]. Let $T = P, Q$ or P^+ (we briefly consider P^+ only to rule it out after the next counterexample). Firstly, $\widetilde{PT}(Y)$ is a complete lattice with an order given by standard subset inclusion and joins given by $\bigvee := \mathsf{cl}^{PT} \circ \bigcup$. This order is lifted pointwise to an order $\sqsubseteq$ on morphisms $X \to \widetilde{PT}(Y)$. What is left is to show that Kleisli composition preserves joins in each argument:

$$\bigvee_{n\in\omega} g_n \odot f = \bigvee_{n\in\omega} (g_n \odot f) \qquad g \odot \bigvee_{n\in\omega} f_n = \bigvee_{n\in\omega} (g \odot f_n)$$

We see that the left condition breaks in the case of $\widetilde{PP^+}$.

Counterexample 1 *Take $f : \{x\} \to \widetilde{PP^+}(\omega)$ and a ω-chain $\{g_i : \omega \to \widetilde{PP^+}(\{w, z\})\}_{i\in\omega}$. Define $f(x) := \{\omega\}$ and*

$$g_i(n) = \begin{cases} \{\{a\}\} & \text{if } n \leq i \\ \{\{a\}, \{b\}, \{a, b\}\} & \text{otherwise} \end{cases}$$

*We find $\{b\} \in \bigvee\{g_i\}_{i\in\omega} \odot f(x)$ because $\{b\} \in g_i(i)$ for each $i \in \omega$. However $\{b\} \notin \bigvee\{g_i \odot f\}_{i\in\omega}(x)$ because there is no $i \in \omega$ such that $\{b\} \in g_i(n)$ for all $n \in \omega$. Hence $\mathbf{Kl}(\widetilde{PP^+})$ is not ω-**cpo**-enriched. The same reasoning will apply to the monotone neighbourhood monad.*

Arguably the most difficult property to establish is that $\mathbf{Kl}(\widetilde{PT})$ is ω-**cpo**-enriched. Note, that the similar property of being enriched in the category directed complete partial orders has been proven for the similar monad considered in [19], in [5] and [10].

Proposition 4.1 *$\mathbf{Kl}(\widetilde{PT})$ is ω-**cpo**-enriched, for $T = Q, D$.*

Proof. We show where the finiteness assumption for $T = Q$ is required (note that the proof for $T = D$ is along same lines). Given $f : X \to \widetilde{PQ}(Y)$ and some ω-chain $\{g_n : Y \to \widetilde{PQ}(Z)\}_{n\in\omega}$:

$$\begin{aligned}
\bigvee_{n\in\omega} g_n \odot f(x) &= \mu^P\mu^Q \circ P\delta^{PQ} \circ PQ(\bigvee_{n\in\omega} g_n) \circ f(x) \\
&= \{\bigcup \bigcup_{y\in V} \mathcal{W}_y \mid V \in f(x), \forall y \in V : \mathcal{W}_y \subseteq^+_\omega \mathsf{cl} \bigcup_{n\in\omega} g_n(y)\} \\
&\overset{(*)}{=} \mathsf{cl}\{\bigcup \bigcup_{y\in V} \mathcal{W}_y \mid n \in \omega, V \in f(x), \forall y \in V : \mathcal{W}_y \subseteq g_n(y)\} \\
&= \bigvee_{n\in\omega} (\mu^P\mu^Q \circ P\delta^{PQ} \circ PQ(g_n) \circ f(x)) = \bigvee_{n\in\omega} (g_n \odot f)(x)
\end{aligned}$$

$(*)(\subseteq)$ Assume some $V \in f(x)$ and $\mathcal{W}_y \subseteq^+_\omega \mathsf{cl} \bigcup_{n\in\omega} g_n(y)$. Let $\mathcal{W}_y = \{\bigcup \mathcal{W}_{yj}\}_{j\in J}$ where $\mathcal{W}_{yj} \subseteq_\omega \bigcup_{n\in\omega} g_n(y)$ for each j in a finite set J. Let $\mathcal{A}_{ny} := \bigcup_{j\in J} \mathcal{W}_{yj} \cap g_i(y)$. Notice $\mathcal{A}_{0y} \subseteq \mathcal{A}_{1y} \subseteq \cdots$ (from $g_0 \sqsubseteq g_1 \sqsubseteq \cdots$), and

$$\bigcup_{n\in\omega} \mathcal{A}_{ny} = \bigcup_{n\in\omega} (\bigcup_{j\in J} \mathcal{W}_{yj} \cap g_n(y)) = \bigcup_{j\in J} \mathcal{W}_{yj} \cap \bigcup_{n\in\omega} g_n(y) = \bigcup_{j\in J} \mathcal{W}_{yj}$$

is finite, so for all $y \in V$, we have some $k_y \in \omega$ with $\mathcal{A}_{k_y y} = \bigcup_{n\in\omega} \mathcal{A}_{ny}$. We set $k := \max_{y\in V} k_y$, so we have that $\bigcup \bigcup_{y\in V} \mathcal{A}_{ky}$ is an element of the RHS. We still need to show that it is equal to the element we started with: $\bigcup \bigcup_{y\in V} \mathcal{A}_{ky} = \bigcup \bigcup_{y\in V} \bigcup_{j\in J} \mathcal{W}_{yj} = \bigcup_{y\in V} \bigcup_{j\in J} \bigcup \mathcal{W}_{yj} = \bigcup_{y\in V} \bigcup \mathcal{W}_y = \bigcup \bigcup_{y\in V} \mathcal{W}_y$.

$(*)(\supseteq)$ Because the LHS is a convex set, we can ignore the closure on the outside of the RHS. Hence, assume we have some $n \in \omega$, $V \in f(x)$, and $\mathcal{W}_y \subseteq^+_\omega g_n(y)$ for each $y \in V$. We can immediately see that $\mathcal{W}_y \subseteq^+_\omega \mathsf{cl} \bigcup_{n\in\omega} g_n(y)$, so are done. □

[3, Theorem 5.14] also proves that composition in Kleisli is *left strict*: $\bot \odot f = \bot$. We find this to be a mistake, because left strictness is a sufficient condition for the isomorphism $\widetilde{PT}(0) \cong 1$ (see [17, Lemma 3.5]), which also does not hold for the choice of T in [3]. We construct a direct counterexample for $T = P_f$, the finite powerset monad, below.

Counterexample 2 *Take $f : \{x\} \to \widetilde{PP_f}(Y)$ as $x \mapsto \{\emptyset\}$. We have $\bot \odot f(x) = \{\emptyset\}$, which means this* *composite is not equal to* $\bot$ *(we should have* $x \mapsto \emptyset$*).*

$$\bot \odot f(x) = \mu^P \mu^{P_f} \circ P\delta^{P_f P} \circ PP_f(\bot)(\{\emptyset\}) = \mu^P \mu^{P_f} \circ P\delta^{P_f P}(\{\emptyset\}) = \mu^P \mu^{P_f}(\{\{\emptyset\}\}) = \{\emptyset\}$$

The condition of left strictness says that *controller deadlocks are preserved by pre-composition.* In our counterexample, the environment deadlocks first (in the f above), and as a result, the environment also deadlocks in the composition: $\bot \odot f(x) = \{\emptyset\}$. This is different from the controller deadlocking: $\bot(x) = \emptyset$. This counterexample also exists in monad considered in [3] (we believe for any positive semifield: by having f point to the singleton "null distribution").

The requirement which is violated in Theorem 3.6 is that $\widetilde{PP_f}(0) \cong \{\emptyset, \{\emptyset\}\} \ncong 1$. A natural fix is to prevent one player from deadlocking; we choose to disallow environment deadlocks, as this ensures bottom elements remain intact.

Proposition 4.2 *Zero maps in* $\mathbf{Kl}(\widetilde{PT})$ *form bottom elements, for* $T = Q, D$.

Proof. We have $\widetilde{PT}(0) = \{\emptyset\}$ for $T = Q, D$, hence:

$$\begin{array}{ccccccc} X & \xrightarrow{\;!\;} & \widetilde{PT}(0) & \xrightarrow{\widetilde{PT}(!_Y)} & \widetilde{PT}\widetilde{PT}(Y) & \xrightarrow{\mu^{\widetilde{PT}}} & \widetilde{PT}(Y) \\ x & \longmapsto & \emptyset & \longmapsto & \emptyset & \longmapsto & \emptyset \end{array}$$

because the multiplication $\mu^{\widetilde{PT}}$ (and the distributive law δ^{PT}) preserves the empty set. □

4.2 Commutativity and a Functor-Monad Distributive Law

Now we have an appropriate ω-**cpo** structure on the Kleisli homsets, we turn our attention to the behaviour functor $F : \mathbf{Set} \to \mathbf{Set}$, and meeting the requirements (iv), (v) and (vi) of Theorem 3.6. For (iv), we require a functor-monad distributive law $F\widetilde{PT} \to \widetilde{PT}F$. It is known that any polynomial functor distributes over a commutative monad [17, Lemma 2.4], and [19, Lemma 5.2] claims their similar monad is commutative. Unfortunately, this appears to be a mistake, as the following counterexample demonstrates. Note that this was also noticed recently (and independently) in [25].

Counterexample 3 *Condition (1), required for the commutativity of* $\widetilde{PQ}$*, fails for the sets* $\{\{x_1\}, \{x_2\}, \{x_1, x_2\}\}$ *and* $\{\{y_1, y_2\}\}$*. If we think of these sets as one-step interactions in games, composing them in different orders give different sets that can be forced. This counterexample also exists for* $\widetilde{PD}$*, the monotone neighbourhood monad, and the semifield monads considered in [19,3].*

The underlying reason for this failure is that δ^{PQ} is not a *weak distributive law of commutative monads* in the sense of [18]. To solve this issue, we restrict F to be a linear functor H, and obtain the required functor-monad distributive law $\lambda : H\widetilde{PT} \to \widetilde{PT}H$ automatically (recall Example 3.2 (ii)). Point (iv) automatically holds for a linear functor (see Section 3.4), and so does point (vi). Finally, (v) is easily proven.

Proposition 4.3 *The extension* $\overline{H} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$ *is locally monotone.*

Proof. Given some $f, g : X \to \widetilde{PT}(Y)$ such that $f \sqsubseteq g$, we must establish $\lambda \circ Hf \sqsubseteq \lambda \circ Hg$, where $\lambda : H\widetilde{PT} \to \widetilde{PT}H$ is the functor-monad distributive law, this is easily verified. □

4.3 Trace Maps

Instantiating Theorem 3.6 for $\widetilde{PT}H$-coalgebras gives us a final $\overline{H}$-coalgebra in $\mathbf{Kl}(\widetilde{PT})$. Hence, given a coalgebra $c : X \to \overline{H}(X)$, we can form the *trace map* $\mathsf{tr}_c : X \to A^*B$ in $\mathbf{Kl}(\widetilde{PT})$ as the unique $\overline{H}$-morphism into the final $\overline{H}$-coalgebra. Recall (from Section 3.3) that A^*B is the carrier of the initial H-algebra.

$$\begin{array}{ccc} X & \overset{\mathsf{tr}_c}{\dashrightarrow} & A^*B \\ \downarrow c & & \uparrow \wr \\ \overline{H}(X) & \overset{\overline{H}(\mathsf{tr}_c)}{\dashrightarrow} & \overline{H}(A^*B) \end{array}$$

This takes a state and maps it to a convex set of collections of A^*B. We can give this map directly using Proposition 3.5. Let $\kappa_n : H^n(0) \to A^*B$ be the coprojection in **Set** over the initial sequence, and recall that c_n is the iterated coalgebra map $X \xrightarrow{c} \overline{H}(X) \xrightarrow{\overline{H}(c)} \dots \xrightarrow{\overline{H^{n-1}}(c)} \overline{H^n}(X)$. Then

$$\mathsf{tr}_c = \bigvee_{n \in \omega} (X \xrightarrow{c_n} \overline{H^n}(X) \xrightarrow{\overline{H^n}(!)} \overline{H^n}(0) \xrightarrow{\overline{\kappa_n}} A^*B)\,.$$

4.4 Execution Maps

Recall that a finite trace is an element of A^*B. Traces describe the *observable* outcomes of plays. To capture the actual plays, we also need to incorporate information about the controller states they visit. We call the resulting concept executions. Formally, an *execution* is an element of the initial algebra $(XA)^*XB$ of the functor H_X from Section 3.3, i.e. a sequence $x_0a_1x_1 \dots a_nx_nb$.

Following [6], we modify an $\overline{H}$-coalgebra c to include state information, obtaining a $\overline{H_X}$-coalgebra c^*:

$$c^* := (X \xrightarrow{\langle \mathrm{id}, c\rangle} X \times TH(X) \xrightarrow{\mathrm{stl}} T(X \times H(X)))$$

$$\begin{array}{ccc} X & \overset{\mathsf{exec}_c}{\dashrightarrow} & (XA)^*XB \\ \downarrow{\scriptstyle c^*} & & \uparrow{\scriptstyle \wr} \\ \overline{H_X}(X) & \overset{\overline{H_X}(\mathsf{exec}_c)}{\dashrightarrow} & \overline{H_X}((XA)^*XB) \end{array}$$

Then, as H_X is still a linear functor, we can apply Theorem 3.6 again to obtain the *execution map* $\mathsf{exec}_c = \mathsf{tr}_{c^*} : X \to (XA)^*XB$ in $\mathbf{Kl}(\widetilde{PT})$.

We will use the following standard result, from the theory of coalgebras, to prove that a trace map for a coalgebra c arises from an execution map for c^*, followed by forgetting the states.

Proposition 4.4 *Suppose functors $F, G : \mathbb{C} \to \mathbb{C}$ with final coalgebras $\zeta_F : Z_F \xrightarrow{\sim} F(Z_F)$ and $\zeta_G : Z_G \xrightarrow{\sim} G(Z_G)$ respectively. A natural transformation $\alpha : F \to G$ induces a G-coalgebra morphism $f_\alpha : Z_F \to Z_G$ and a functor $E_\alpha : \mathbf{Coalg}(F) \to \mathbf{Coalg}(G)$ such that for any coalgebra $c : X \to F(X)$ we have that*

$$X \overset{\mathrm{beh}_c}{\dashrightarrow} Z_F \overset{f_\alpha}{\dashrightarrow} Z_G, \qquad X \overset{\mathrm{beh}_{E_\alpha(c)}}{\dashrightarrow} Z_G$$

Proposition 4.5 (Traces via Executions) *The projection $\pi_2 : H_X \to H$ is a natural transformation, inducing a function $f_{\pi_2} : (XA)^*XB \to A^*B$. The trace map can be factored via the execution map and f_{π_2}: we have $f_{\overline{\pi_2}} \odot \mathsf{exec}_c = \mathsf{tr}_c$ for any $c : X \to H(X)$.*

Proof. This follows from Proposition 4.4 with the fact that $E_{\overline{\pi_2}}(c^*) = \overline{\pi_2} \odot c^* = \mu \circ \eta \circ \pi_2 \circ c^* = c$. □

5 Strategies

This section shows how we recover (i) the usual notion of strategy in a two-player game, as a chain of maps in the Kleisli category of the monad T, and (ii) the outcome of a strategy, by lifting this chain to $\mathbf{Kl}(\widetilde{PT})$ (where our games, and the appropriate limit, exist). Recall that a strategy σ maps incomplete plays which conform to σ ending in a controller state x, to a successor of x (an environment state). The latter is an element of $TH(X)$, and thus the natural home for strategies is $\mathbf{Kl}(T)$.

The reason to lift each $\sigma_n : \mathrm{Im}(\sigma_n) \to \widehat{H_X^n}(X)$ to $\mathbf{Kl}(\widetilde{PT})$ is threefold. Firstly, we need $(XA)^*XB$ to be the limit of the final sequence, which gives us a unique mediating map $1 \to (XA)^*XB$ in $\mathbf{Kl}(\widetilde{PT})$. We also need to be able to reason that σ_n picks a successor in c, which is a morphism in $\mathbf{Kl}(\widetilde{PT})$, the order structure in $\mathbf{Kl}(\widetilde{PT})$ lets us do this. The final reason is conceptual, we want the outcome of a strategy to be from the *controllers perspective*, which is captured in $\mathbf{Kl}(\widetilde{PT})$, rather than $\mathbf{Kl}(T)$ which is the environments perspective. This distinction is particularly prominent when a particular strategy does not force a collection of completed plays (see Remark 5.6).

This lifting is provided by a functor K in the following proposition.

Proposition 5.1 *There is an identity-on-objects functor* $K : \mathbf{Kl}(T) \to \mathbf{Kl}(\widetilde{PT})$*, defined by*

$$K(X \xrightarrow{f} T(Y)) := (X \xrightarrow{f} T(Y) \xrightarrow{\eta^P} PT(Y) \xrightarrow{\mathsf{cl}^{PT}} \widetilde{PT}(Y))$$

Proof. Functoriality of K follows from $\mathsf{cl}^{PT} \circ \eta^P$ being a monad morphism, see [12, Proposition 2.7]. □

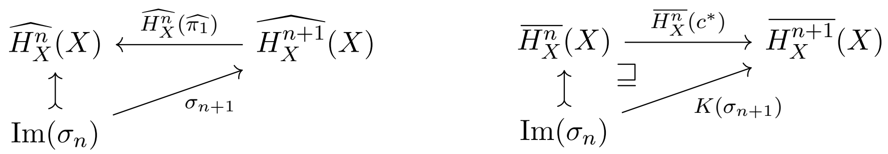
We are now ready to give a categorical definition of a strategy. It will involve a chain of maps that live in $\mathbf{Kl}(T)$, so we also need liftings and extensions to $\mathbf{Kl}(T)$. To differentiate Kleisli categories, we use $\widehat{f}$ for lifting functions to $\mathbf{Kl}(T)$, and $\widehat{F}$ for extensions to $\mathbf{Kl}(T)$.

In the following definition, we also use the projection $\pi_1 : X(B + AX) \to X$ out of the product. For the reader's intuition, we spell out the map

$$H^n_X(\pi_1) : (XA)^{<n}XB + (XA)^n X(B + AX) \to (XA)^{<n}XB + (XA)^n X$$

as preserving complete plays of length n or less, and mapping complete and partial plays of length $n+1$ to the partial plays of length n which they extend. We also use image $\mathrm{Im}(-)$ slightly informally. Technically, $\mathrm{Im}(-)$ maps a $\mathbf{Kl}(T)$ morphism $f : X \to TY$ to the set $\bigcup \circ P(\mathsf{supp}) \circ Pf(X)$.

Definition 5.2 Let (X, c) be an $\overline{H}$-coalgebra, and $x \in X$ be a state. A *strategy* σ *from* x is a chain of maps $\{\sigma_n\}_{n\in\omega}$ in $\mathbf{Kl}(T)$, where $\sigma_0 : 1 \to X$ picks $\eta^T(x)$, while $\sigma_{n+1} : \mathrm{Im}(\sigma_n) \to \widehat{H^{n+1}_X}(X)$ for $n \in \omega$, is such that



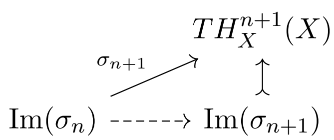
commute. Denote the set of strategies from x as $\Sigma_c(x)$. Define an *n-step strategy from* x as a finite collection of maps $\{\sigma_0, \ldots, \sigma_n\}$ with the same conditions. Denote the set of n-step strategies from x by $\Sigma_{n,c}(x)$.

The left condition states that σ_{n+1} preserves completed plays and extends incomplete plays (by either completing them or by adding a successor). The right condition states that σ_n chooses out of the successors in (X, c). Notice that σ_n cannot force a play ending in a state $x \in X$ with no successors (i.e. a controller deadlock), because then σ_{n+1} is not definable, as there is no way to satisfy the right condition above. Hence, there are n-step strategies $\{\sigma_0, \ldots, \sigma_n\}$ which cannot be extended to a (full) strategy $\{\sigma_n\}_{n\in\omega}$.

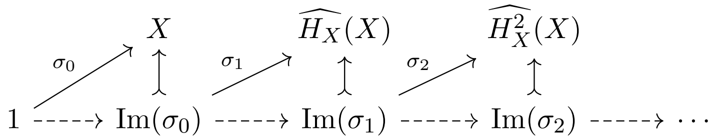
We recall (see Section 3.1 and 3.2), that a standard strategy (x, σ) for a $\widetilde{PT}H$-coalgebra is a partial function from $(XA)^*X$ to controller states, defined over exactly the partial plays which conform to it, which extends partial plays ending in x with a move chosen from the controllers edge relation. With a few inconsequential assumptions on the two-player games and MDPs defined in Section 3.1 and 3.2, they are equivalently coalgebras $X \to \widetilde{PT}H(X)$, for $T = Q$ and $T = D$ respectively. Furthermore, standard strategies are then in one-to-one correspondence with strategies defined in Definition 5.2.

Given an strategy σ at a state x, the set of plays from x up to some depth n naturally arises from composition in $\mathbf{Kl}(T)$. For example, in the $T = Q$ case, composition takes a union at each step, so composing with σ_{n+1} takes a set of complete plays (of length $\leq n$) and partial plays (of length n) to a set of complete (of length $\leq n+1$) and partial plays (of length $n+1$). This is the content of our next definition. Again, this can be seen to agree with the n-step partial outcome of a standard strategy in discussed Section 3.

We require the dashed morphism and injection in the commutative diagram below (the unique surjective-injective factorisation of σ_{n+1} in $\mathbf{Set}$), which can be lifted into $\mathbf{Kl}(T)$ with J_T.



Definition 5.3 Let σ be a strategy from x. For $n \in \omega$, define $\mathsf{plays}^\sigma_n(x) : 1 \to \widehat{H^n_X}(X)$ as the composition of the dashed arrows to $\mathrm{Im}(\sigma_n)$ with the injection to $\overline{H^n_X}(X)$.



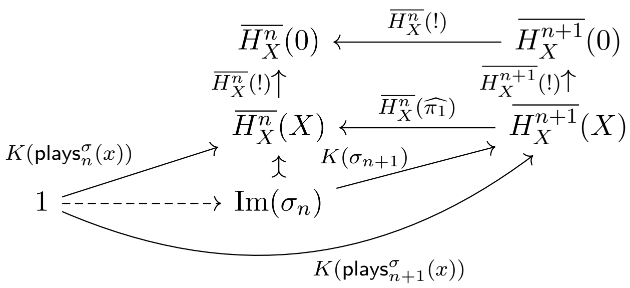
We refer to $\mathsf{plays}^\sigma_n(x)$ as the n-step partial outcome of σ.

For simplicity, the following discusses $T = Q$, but the same reasoning applies to $T = D$ too. The map $\mathsf{plays}^\sigma_n(x) : 1 \to \widehat{H^n_X}(X)$ gives the set of complete plays of length less than or equal to n and partial plays of length n, which can be obtained from fixing the controller strategy σ from x. We now want the *completed outcome* of a strategy: all σ-conform complete plays from a state x. It is tempting to try to apply Theorem 3.6 to obtain a final $\overline{H}$-coalgebra in $\mathbf{Kl}(Q)$, and then to obtain a mediating map from a cone over the final sequence. However we can see, due to Q being *finite* subsets, that joins of chains of morphisms are not guaranteed to exist. For example, the union of $\{x_0\} \subseteq \{x_0, x_1\} \subseteq \{x_0, x_1, x_2\} \subseteq \ldots$ is not a finite set. This failure means $\mathbf{Kl}(Q)$ is not ω-**cpo** enriched, so Theorem 3.6 and Proposition 3.5 do not apply.

We solve this problem by using K to lift the chain to $\mathbf{Kl}(\widetilde{PT})$, where we use it to form a cone over the final sequence. Note that, from the proof of Theorem 3.6 in [21], we know that $0 \xleftarrow{!} \overline{H_X}(0) \xleftarrow{\overline{H_X}(!)} \overline{H^2_X}(0) \leftarrow \cdots$ is the final sequence of $\overline{H_X}$ in $\mathbf{Kl}(\widetilde{PT})$.

Proposition 5.4 *Fix a strategy σ from x, in some $\overline{H}$-coalgebra.* $(1 \xrightarrow{K(\mathsf{plays}^\sigma_n(x))} \overline{H^n_X}(X) \xrightarrow{\overline{H^n_X}(!)} \overline{H^n_X}(0))_{n\in\omega}$ *defines a cone over the final sequence of $\overline{H_X}$* $0 \xleftarrow{!} \overline{H_X}(0) \xleftarrow{\overline{H_X}(!)} \overline{H^2_X}(0) \leftarrow \cdots$.

Proof. The top square commutes by finality (0 is final in $\mathbf{Kl}(\widetilde{PT})$), the rest commute from definitions.



□

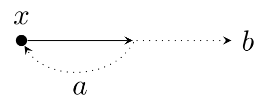
We are now ready to define the completed plays that conform to an (X, c)-strategy from a state x.

Definition 5.5 Fix an $\overline{H}$-coalgebra (X, c) and a state $x \in X$. For $\sigma \in \Sigma_c(x)$, define $\mathsf{plays}^\sigma_c(x) : 1 \to (XA)^*XB$ in $\mathbf{Kl}(\widetilde{PT})$, as the unique mediating map from the cone in Proposition 5.4. This represents the completed outcome from a state x using a strategy σ. Explicitly, from Proposition 3.5, we have:

$$\mathsf{plays}^\sigma_c(x) = \bigvee_{n\in\omega} (1 \xrightarrow{K(\mathsf{plays}^\sigma_n(x))} \overline{H^n_X}(X) \xrightarrow{\overline{H^n_X}(!)} \overline{H^n_X}(0) \xrightarrow{\overline{\kappa_n}} (XA)^*XB)$$

where $\kappa_n : H^n_X(0) \to (XA)^*XB$ is the coprojection from the initial chain for H_X into the initial H_X-algebra.

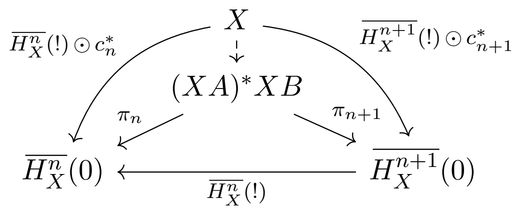
Remark 5.6 We stress that the outcome of a strategy from a state is a map into $\widetilde{PT}((XA)^*XB)$, rather than $T((XA)^*XB)$. The set $\mathsf{plays}^\sigma_c(x)$ will be a singleton if all plays which conform to σ are completed in a finite number of steps, and will otherwise be empty. For example, consider the game below with a non-deterministic environment, where there is only one strategy.



We compute the n-step partial outcomes (which include incomplete plays) as $\{x\}, \{b, xax\}, \{b, xaxb, xaxax\}, \ldots$ The set $\mathsf{plays}^{\sigma}_{c}(x)$ will however be empty, as none of the n-step partial outcomes will ever be a set of completed plays. In Definition 5.5, the map $\overline{H^n_X}(!)$ performs this operation of removing sets which contain incomplete plays, this is discussed further in Proposition 6.2. This example also points to another reason why defining the outcome of a strategy σ should be done in $\mathbf{Kl}(\widetilde{PT})$ and not in $\mathbf{Kl}(T)$: unlike $\overline{H^n_X}(!)$, if a similar map $\widehat{H^n_X}(!)$ existed in $\mathbf{Kl}(T)$, it would remove the incomplete plays. So even if joins of chains of morphisms did exist in $\mathbf{Kl}(T)$, the resulting joins would describe the outcomes that the *environment* could force when playing against σ, and not the set of completed outcomes which the controller can force when playing σ.

6 Executions via Strategies

We are now ready to prove our main result, Theorem 6.5, which states that the executions from a state x are precisely the unions of the completed outcomes of all strategies from x. The proof is spread over the three next lemmas. The goal is to show that the map which sends a state to the union of the outcomes of all strategies for that state, depicted using a dashed arrow, makes the diagram on the right commute. The limiting property of $(XA)^*XB$ (recall that this is the limit of the final sequence of $\overline{H_X}$) then equates this map with the execution map. Let T be either Q or D, and fix an arbitrary $\widetilde{PT}H$-coalgebra (X, c) for the remainder of the section.



Our first lemma describes how n-step partial outcomes of n-step strategies from x correspond to elements of $c^*_n(x) \in \widetilde{PT}H^n_X(X)$. It is important we only consider n-step strategies rather than full strategies here, as a deadlock in the future prevents some σ_{n+i} from existing (see the comments under Definition 5.2). This lemma is an inductive version of what happens when applying the weak distributives once: where they reverse the branching by giving the one-step outcome of each of the controller choices.

Lemma 6.1 *Unfolding c^* n times at a state x gives the set of n-step partial outcomes of all n-step strategies starting at x.*

$$c^*_n(x) = \{\mathsf{plays}^{\sigma}_{n}(x) \mid \sigma \in \Sigma_{n,c}(x)\}$$

Proof. (Sketch) The proof proceeds by induction. The base case ($n = 0$) holds as both sides are $\{\eta^T(x)\}$. Now we provide a sketch the inductive case for $T = Q$. To simplify notation, we assume that $\mathsf{plays}^{\sigma}_{n}(x) = \{\rho^{\sigma}_1 x^{\sigma}_1, \rho^{\sigma}_2 x^{\sigma}_2\}$ for every $\sigma \in \Sigma_{n,c}(x)$, i.e. that each pointed n-step strategy (x, σ) forces a set of two partial plays after n steps - a similar argument works in general.

$$\begin{aligned} c^*_{n+1}(x) &= \overline{H^n_X}({}^*c) \odot c^*_n(x) = \mu^P \mu^Q \circ P\delta^{PQ} \circ PQ(\lambda_n) \circ H^n_X(c) \circ c^*_n(x) \\ &= \mu^P \mu^Q \circ P\delta^{PQ} \circ PQ(\lambda_n) \circ H^n_X(c)(\{\{\rho^{\sigma}_1 x^{\sigma}_1, \rho^{\sigma}_2 x^{\sigma}_2\} \mid \sigma \in \Sigma_{n,c}(x)\}) \\ &= \mu^P \mu^Q(\{\delta^{PQ}(\{\{\{\rho^{\sigma}_i x^{\sigma}_i \cdot U\} \mid U \in c^*(x^{\sigma}_i)\} \mid i \in 1,2\}) \mid \sigma \in \Sigma_{n,c}(x)\}) \\ &= \mu^P \mu^Q(\{\{\mathcal{V}^{\sigma}_1 \cup \mathcal{V}^{\sigma}_2 \mid \mathcal{V}^{\sigma}_i \subseteq \{\{\rho^{\sigma}_i x^{\sigma}_i \cdot U\} \mid U \in c^*(x)\} \text{ for } i \in 1,2\} \mid \sigma \in \Sigma_{n,c}(x)\}) \\ &= \{\textstyle\bigcup(\mathcal{V}^{\sigma}_1 \cup \mathcal{V}^{\sigma}_2) \mid \sigma \in \Sigma_{n,c}(x),\ \mathcal{V}^{\sigma}_i \subseteq \{\{\rho^{\sigma}_i x^{\sigma}_i \cdot U\} \mid U \in c^*(x)\} \text{ for } i \in 1,2\} \end{aligned}$$

The equality after the first line follows by assumption, and the inductive hypothesis. We also find that:

$$\{\mathsf{plays}^{\sigma}_{n+1}(x) \mid \sigma \in \Sigma_{n+1,c}(x)\} = \{\sigma_{n+1}(\rho^{\sigma}_1 x^{\sigma}_1) \cup \sigma_{n+1}(\rho^{\sigma}_2 x^{\sigma}_2) \mid \sigma \in \Sigma_{n+1,c}(x)\}$$

We now show these two sets are equal. ($\supseteq$) Assume some $(n+1)$-step partial strategy $\sigma \in \Sigma_{n+1,c}(x)$. We immediately have an n-step partial strategy $\{\sigma_i\}_{i \leq n}$ (and we know it forces $\rho^{\sigma}_1 x^{\sigma}_1$ and $\rho^{\sigma}_2 x^{\sigma}_2$ by assumption).

Choosing $\mathcal{V}_1^\sigma := \{\sigma_{n+1}(\rho_1^\sigma x_1^\sigma)\}$ and $\mathcal{V}_2^\sigma := \{\sigma_{n+1}(\rho_2^\sigma x_2^\sigma)\}$, recovers the element $\bigcup(\mathcal{V}_1^\sigma \cup \mathcal{V}_2^\sigma) = \sigma_{n+1}(\rho_1^\sigma x_1^\sigma) \cup \sigma_{n+1}(\rho_1^\sigma x_1^\sigma)$ in $c^*_{n+1}(x)$. For the ($\subseteq$) direction, assume some n-step partial strategy $\sigma \in \Sigma_{n,c}(x)$, and appropriate sets of subsets $\mathcal{V}_i^\sigma$. We can extend σ to a $(n+1)$-step strategy by choosing $\sigma_{n+1}(\rho_1^\sigma x_1^\sigma) := \bigcup \mathcal{V}_1^\sigma$ and $\sigma_{n+1}(\rho_2^\sigma x_2^\sigma) := \bigcup \mathcal{V}_2^\sigma$. This gives us an element in the RHS, corresponding to each element in $c^*_{n+1}(X)$ because $\bigcup(\mathcal{V}_1^\sigma \cup \mathcal{V}_2^\sigma) = \bigcup \mathcal{V}_1^\sigma \cup \bigcup \mathcal{V}_2^\sigma$. Checking this is a well-defined $(n+1)$-step partial strategy follows from $\mathcal{V}_i^\sigma \subseteq \{\{\rho_i^\sigma x_i^\sigma \cdot U\} \mid U \in c^*(x)\}$. ☐

The next lemma describes how composing with $\overline{H_X^n}(!)$ equates the set of all the n-step partial outcomes of n strategies in $\Sigma_n(x)$, and the set of all the n-step partial outcomes of whole strategies in $\Sigma_c(x)$. The map $\overline{H_X^n}(!) : (XA)^{<n}XB + (XA)^n X \to (XA)^{<n}XB$ in $\mathbf{Kl}(\widetilde{PT})$ removes collections which contain incomplete plays. By direct calculation we can see:

$$\overline{H_X^n}(!) \odot U = \bigcup_{u \in U} (P\mu^T \circ \delta^{PT} \circ T(\lambda_n \circ H_X^n(!))(u))$$

Now $\overline{H_X^n}(!) = \lambda_n \circ H_X^n(!) : H_X^n(X) \to \widetilde{PT} H_X^n(0)$ maps a completed execution χ to $\{\eta^T(\chi)\}$ and an incomplete execution ρ to $\emptyset$. Both laws $\delta^{PT} : TP \to PT$ we use have the property $\delta(u) = \emptyset \iff \emptyset \in \mathsf{supp}\, u$, Thus, in the expression above, only u's which only contain completed traces will be kept. We summarise our finding in the following proposition.

Proposition 6.2 *Recall that* $! : X \to \widetilde{PT}(0)$ *is the unique morphism into the final object in* $\mathbf{Kl}(\widetilde{PT})$*. Composition with* $\overline{H_X^n}(!) : H_X^n(X) \to \widetilde{PT} H_X^n(0)$ *in* $\mathbf{Kl}(\widetilde{PT})$ *filters out collections (elements of* $TH_X^n(X)$*) which contain incomplete executions. Formally: given some* $U \in \widetilde{PT} H_X^n(X)$*,*

$$\overline{H_X^n}(!) \odot U = \{u \in TH_X^n(X) \mid u \in U \text{ and } \mathsf{supp}\, u \subseteq H_X^n(0)\}$$

Lemma 6.3 *The* n*-step partial outcomes of* n*-step strategies which only contain completed plays, are equal to the* n*-step partial outcomes of full strategies which only contain completed plays. For all* $x \in X$*, we have*

$$\overline{H_X^n}(!) \odot \{\mathsf{plays}_n^\sigma(x) \mid \sigma \in \Sigma_{n,c}(x)\} = \overline{H_X^n}(!) \odot \{\mathsf{plays}_n^\sigma(x) \mid \sigma \in \Sigma_c(x)\}$$

Proof. ($\subseteq$) To extend some $\sigma \in \Sigma_{n,c}(x)$, we take

$$\sigma_{n+i+1} := (\mathrm{Im}(\sigma_{n+i}) \rightarrowtail H_X^{n+i}(0) \rightarrowtail H_X^{n+i+1}(X) \xrightarrow{\eta^T} TH_X^{n+i+1}(X))$$

for all $i \in \omega$, which just maps a completed trace χ to $\eta^T(\chi)$. This clearly has no impact on the n-step partial outcome. ($\supseteq$) Suppose $\sigma \in \Sigma_c(x)$, we can immediately see that $\{\sigma_i\}_{i \leq n} \in \Sigma_{n,c}(x)$, and both will result in the same n-step partial outcome. ☐

Now for the final lemma. We require the map $\pi_n : (XA)^*XB \to \widetilde{PT} H_X^n(0)$, which is the unique projection of the embedding $\overline{\kappa_n}$ which is lifted from **Set**. It is easily calculated as sending $\chi \in (XA)^*XB$ to $\eta^{\widetilde{PT}}(\chi)$ if $\chi \in H_X^n(0)$ and $\emptyset$ otherwise. Composition with π in $\mathbf{Kl}(\widetilde{PT})$ thus filters out collections which contain complete executions which have length greater than n (are not elements of $H_X^n(0)$).

$$\pi_n \odot U = \{u \in TH_X^n(0) \mid u \in U\}$$

Lemma 6.4 *The union of the outcomes of strategies which only contain completed plays of length less than* n*, equals the set of* n*-step partial outcomes of strategies which only contain completed plays.*

$$\pi_n \odot \bigcup_{\sigma \in \Sigma_c(x)} \mathsf{plays}_c^\sigma(x) = \overline{H_X^n}(!) \odot \{\mathsf{plays}_n^\sigma(x) \mid \sigma \in \Sigma_c(x)\}$$

Theorem 6.5 (Executions via Strategies) *The execution map recovers, at a state $x \in X$, the union of the completed outcomes of all the strategies originating from x.*

$$\mathsf{exec}_c(x) = \bigcup_{\sigma \in \Sigma_c(x)} \mathsf{plays}_c^\sigma(x)$$

Proof. We prove the triangle commutes discussed at the top of the section:

$$\begin{aligned}
\overline{H_X^n}(!) \odot c_n^*(x) &= \overline{H_X^n}(!) \odot \bigcup_{\sigma \in \Sigma_n(x)} K(\mathsf{plays}_n^\sigma(x)) && \text{Lemma 6.1} \\
&= \overline{H_X^n}(!) \odot \bigcup_{\sigma \in \Sigma_c(x)} K(\mathsf{plays}_n^\sigma(x)) && \text{Lemma 6.3} \\
&= \pi_n \odot \bigcup_{\sigma \in \Sigma_c(x)} \mathsf{plays}_c^\sigma(x) && \text{Lemma 6.4}
\end{aligned}$$

which implies that $\bigcup_{\sigma \in \Sigma_c(-)} \mathsf{plays}_c^\sigma(-) : X \to (XA)^*XB$ is the unique map by the limiting property of $(XA)^*XB$ with projections $\pi_n : (XA)^X B \to \widetilde{PT}H_X^n(X)$. □

Recall $f_{\pi_2} : (XA)^*XB \to A^*B$ from Proposition 4.5.

Corollary 6.6 (Traces via Strategies) $\mathsf{tr}_c(x) = f_{\overline{\pi_2}} \odot \bigcup_{\sigma \in \Sigma_c(x)} \mathsf{plays}_c^\sigma(x)$

7 Conclusion

We have shown how to fit two-player controller-versus-environment games into the finite trace semantics framework of Hasuo, Jacobs and Sokolova [17], by identifying that the monads $\widetilde{PQ}$ and $\widetilde{PD}$, built from a weak distributive laws, and the class of linear functors satisfy the necessary requirements. Along the way we uncovered two mistakes in [3] and [19], which manifest in our work as composition in $\widetilde{PP}$ not being left-strict, and $\widetilde{PQ}$ not being a commutative monad respectively. We gave a categorical definition of strategies in games, and showed how the execution map recovers the set of collections of plays which can be forced by a controller strategy.

Building on the automata-theoretic approach to synthesis, future work will include using a non-deterministic automaton to describe the desired linear-time behaviour of game plays, and using a product construction to aid synthesis. By our main result, if the set of collections of traces from a state in the product is non-empty (and if traces on the product give traces on the game), then the controller has a strategy to realise the desired behaviour. We have some preliminary results in this direction.

In the future we would like to extend our results to *infinite* traces, since in the context of verification and synthesis one is mainly interested in linear-time properties of potentially infinite computations. Results in [11] show the appropriate weak distributive laws exist for monads modelling continuous probability on certain categories of topological spaces, which will be required when moving to infinite traces. Another possible extension would be incorporating simple stochastic games into our results, however work in [1] indicates that the natural approach of lifting the weak distributive law $PP \to PP$ to the category of convex algebras may not be possible.

References

[1] Aristote, Q., *Monotone Weak Distributive Laws over the Lifted Powerset Monad in Categories of Algebras*, in: O. Beyersdorff, M. Pilipczuk, E. Pimentel and N. e. K. Thang, editors, *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, volume 327 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:20, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2025), ISBN 978-3-95977-365-2, ISSN 1868-8969.
https://doi.org/10.4230/LIPIcs.STACS.2025.10

[2] Bertrand, M. and J. Rot, *Coalgebraic Determinization of Alternating Automata* (2018). ArXiv:1804.02546 [cs].
https://doi.org/10.48550/arXiv.1804.02546

[3] Bonchi, F. and A. Santamaria, *Convexity via Weak Distributive Laws*, Logical Methods in Computer Science **Volume 18, Issue 4**, page 8389 (2022), ISSN 1860-5974. ArXiv:2108.10718 [cs, math].
https://doi.org/10.46298/lmcs-18(4:8)2022

[4] Bowler, N., P. B. Levy and G. Plotkin, *Initial algebras and final coalgebras consisting of nondeterministic finite trace strategies*, Electronic Notes in Theoretical Computer Science **341**, pages 23–44 (2018), ISSN 1571-0661. Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).
https://doi.org/https://doi.org/10.1016/j.entcs.2018.11.003

[5] Brengos, T., *Weak bisimulation for coalgebras over order enriched monads*, Logical Methods in Computer Science **Volume 11, Issue 2**, 14 (2015), ISSN 1860-5974.
https://doi.org/10.2168/LMCS-11(2:14)2015

[6] Cîrstea, C., *Maximal traces and path-based coalgebraic temporal logics*, Theoretical Computer Science **412**, pages 5025–5042 (2011), ISSN 0304-3975.
https://doi.org/10.1016/j.tcs.2011.04.025

[7] Filar, J. and K. Vrieze, *Competitive Markov decision processes*, Springer-Verlag, Berlin, Heidelberg (1996), ISBN 0387948058.

[8] Frank, F., S. Milius and H. Urbat, *Coalgebraic Semantics for Nominal Automata*, in: H. H. Hansen and F. Zanasi, editors, *Coalgebraic Methods in Computer Science*, pages 45–66, Springer International Publishing, Cham (2022), ISBN 978-3-031-10736-8.
https://doi.org/10.1007/978-3-031-10736-8_3

[9] Garner, R., *The Vietoris Monad and Weak Distributive Laws*, Applied Categorical Structures **28**, pages 339–354 (2020), ISSN 1572-9095.
https://doi.org/10.1007/s10485-019-09582-w

[10] Goncharov, S. and D. Pattinson, *Coalgebraic weak bisimulation from recursive equations over monads*, in: J. Esparza, P. Fraigniaud, T. Husfeldt and E. Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 196–207, Springer (2014).
https://doi.org/10.1007/978-3-662-43951-7_17

[11] Goubault-Larrecq, J., *Weak distributive laws between monads of continuous valuations and of non-deterministic choice*, ArXiv **abs/2408.15977** (2024).
https://api.semanticscholar.org/CorpusID:271974412

[12] Goy, A., *On the compositionality of monads via weak distributive laws*, Ph.D. thesis, Université Paris-Saclay (2021).
https://theses.hal.science/tel-03426949

[13] Goy, A. and D. Petrişan, *Combining probabilistic and non-deterministic choice via weak distributive laws*, in: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, pages 454–464, Association for Computing Machinery, New York, NY, USA (2020), ISBN 978-1-4503-7104-9.
https://doi.org/10.1145/3373718.3394795

[14] Goy, A., D. Petrişan and M. Aiguier, *Powerset-Like Monads Weakly Distribute over Themselves in Toposes and Compact Hausdorff Spaces*, in: *DROPS-IDN/v2/document/10.4230/LIPIcs.ICALP.2021.132*, Schloss-Dagstuhl - Leibniz Zentrum für Informatik (2021).
https://doi.org/10.4230/LIPIcs.ICALP.2021.132

[15] Hansen, H. H. and C. Kupke, *Weak completeness of coalgebraic dynamic logics*, in: R. Matthes and M. Mio, editors, *Proceedings Tenth International Workshop on Fixed Points in Computer Science, FICS 2015, Berlin, Germany, September 11-12, 2015*, volume 191 of *EPTCS*, pages 90–104 (2015).
https://doi.org/10.4204/EPTCS.191.9

[16] Hansen, H. H., C. Kupke, J. Marti and Y. Venema, *Parity games and automata for game logic (extended version)*, CoRR **abs/1709.00777** (2017). 1709.00777.
http://arxiv.org/abs/1709.00777

[17] Hasuo, I., B. Jacobs and A. Sokolova, *Generic Trace Semantics via Coinduction*, Logical Methods in Computer Science **Volume 3, Issue 4**, page 864 (2007), ISSN 1860-5974. ArXiv:0710.2505 [cs].
https://doi.org/10.2168/LMCS-3(4:11)2007

[18] Jacobs, B., *Semantics of weakening and contraction*, Annals of Pure and Applied Logic **69**, pages 73–106 (1994), ISSN 0168-0072.
https://doi.org/https://doi.org/10.1016/0168-0072(94)90020-5

[19] Jacobs, B., *Coalgebraic Trace Semantics for Combined Possibilitistic and Probabilistic Systems*, Electronic Notes in Theoretical Computer Science **203**, pages 131–152 (2008), ISSN 1571-0661.
https://doi.org/10.1016/j.entcs.2008.05.023

[20] Jacobs, B., *A Recipe for State-and-Effect Triangles*, in: *DROPS-IDN/v2/document/10.4230/LIPIcs.CALCO.2015.116*, Schloss-Dagstuhl - Leibniz Zentrum für Informatik (2015).
https://doi.org/10.4230/LIPIcs.CALCO.2015.116

[21] Jacobs, B., *Monads, Comonads and Distributive Laws*, page 246–333, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press (2016).

[22] Jacobs, B., A. Silva and A. Sokolova, *Trace Semantics via Determinization*, in: D. Pattinson and L. Schröder, editors, *Coalgebraic Methods in Computer Science*, pages 109–129, Springer, Berlin, Heidelberg (2012), ISBN 978-3-642-32784-1.
https://doi.org/10.1007/978-3-642-32784-1_7

[23] Jacobs, B. and A. Sokolova, *Traces, Executions and Schedulers, Coalgebraically*, in: A. Kurz, M. Lenisa and A. Tarlecki, editors, *Algebra and Coalgebra in Computer Science*, pages 206–220, Springer, Berlin, Heidelberg (2009), ISBN 978-3-642-03741-2.
https://doi.org/10.1007/978-3-642-03741-2_15

[24] Kojima, R., C. Cîrstea, K. Muroya and I. Hasuo, *Coalgebraic CTL: fixpoint characterization and polynomial-time model checking*, in: B. König and H. Urbat, editors, *Coalgebraic Methods in Computer Science - 17th IFIP WG 1.3 International Workshop, CMCS 2024, Colocated with ETAPS 2024, Luxembourg City, Luxembourg, April 6-7, 2024, Proceedings*, volume 14617 of *Lecture Notes in Computer Science*, pages 1–22, Springer (2024).
https://doi.org/10.1007/978-3-031-66438-0_1

[25] Liell-Cock, J. and S. Staton, *Compositional imprecise probability: A solution from graded monads and markov categories*, Proc. ACM Program. Lang. **9** (2025).
https://doi.org/10.1145/3704890

[26] Mazala, R., *Infinite Games*, pages 23–38, Springer Berlin Heidelberg, Berlin, Heidelberg (2002), ISBN 978-3-540-36387-3.
https://doi.org/10.1007/3-540-36387-4_2

[27] Milius, S., D. Pattinson and L. Schröder, *Generic Trace Semantics and Graded Monads*, in: L. S. Moss and P. Sobocinski, editors, *6th Conference on Algebra and Coalgebra in Computer Science (CALCO 2015)*, volume 35 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 253–269, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2015), ISBN 978-3-939897-84-2, ISSN 1868-8969.
https://doi.org/10.4230/LIPIcs.CALCO.2015.253

[28] Smyth, M. B. and G. D. Plotkin, *The Category-Theoretic Solution of Recursive Domain Equations*, SIAM Journal on Computing **11**, pages 761–783 (1982), ISSN 0097-5397. Publisher: Society for Industrial and Applied Mathematics.
https://doi.org/10.1137/0211062

[29] Vickers, S., *Topology via logic*, Cambridge University Press, USA (1989), ISBN 0521360625.