Traces via Strategies

Ben Plummer, Corina Cîrstea

University of Southampton

June, 2025

Finite trace semantics

- Linear-time approach in concurrency theory:

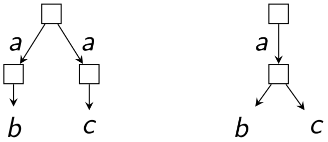
  process equivalence $\sim$ trace equivalence

Finite trace semantics

- Linear-time approach in concurrency theory:

  process equivalence $\sim$ trace equivalence

- e.g. $(a.b) + (a.c)$ and $a.(b + c)$ are trace equivalent:

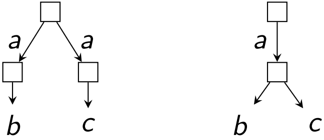


Finite trace semantics

- Linear-time approach in concurrency theory:

  process equivalence $\sim$ trace equivalence

- e.g. $(a.b) + (a.c)$ and $a.(b + c)$ are trace equivalent:



- The models are labelled transition systems:

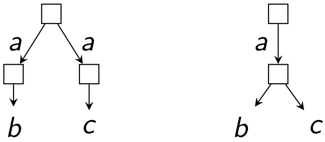
$$\frac{X \to P(B + A \times X)}{R \subseteq X \times (B + A \times X)}$$

Finite trace semantics

- Linear-time approach in concurrency theory:

  process equivalence ∼ trace equivalence

- e.g. $(a.b) + (a.c)$ and $a.(b + c)$ are trace equivalent:



- The models are labelled transition systems:

$$\frac{X \to P(B + A \times X)}{R \subseteq X \times (B + A \times X)}$$

- Finite trace semantics have enjoyed a general coalgebraic treatment in [Hasuo et al., 2007], which covers LTSs, MCs, CFGs.

Two-player games

- A useful model for a controller ($\exists$) in an environment ($\forall$)

Two-player games

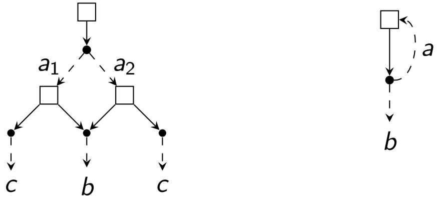
- A useful model for a controller ($\exists$) in an environment ($\forall$)
- Arose when considering Church's synthesis problem:

Two-player games

- A useful model for a controller ($\exists$) in an environment ($\forall$)
- Arose when considering Church's synthesis problem:
  - Take a specification φ, e.g. "terminate with a b"

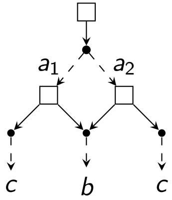
Two-player games

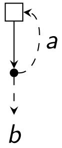
- A useful model for a controller (∃) in an environment (∀)
- Arose when considering Church's synthesis problem:
  - Take a specification φ, e.g. "terminate with a b"
  - Take a game modelling a controller in its environment



Two-player games

- A useful model for a controller (∃) in an environment (∀)
- Arose when considering Church's synthesis problem:
  - Take a specification φ, e.g. "terminate with a b"
  - Take a game modelling a controller in its environment

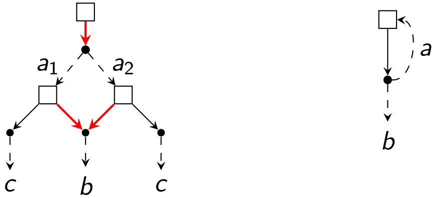




  - Implementation for the controller is a winning strategy for □

Two-player games

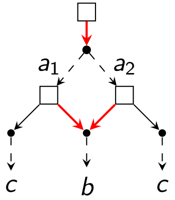
- A useful model for a controller (∃) in an environment (∀)
- Arose when considering Church's synthesis problem:
  - Take a specification φ, e.g. "terminate with a b"
  - Take a game modelling a controller in its environment

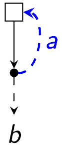


  - Implementation for the controller is a winning strategy for □

Two-player games

- A useful model for a controller (∃) in an environment (∀)
- Arose when considering Church's synthesis problem:
  - Take a specification φ, e.g. "terminate with a b"
  - Take a game modelling a controller in its environment

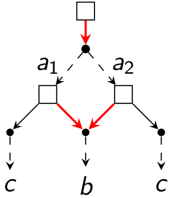


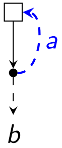


  - Implementation for the controller is a winning strategy for □

Two-player games

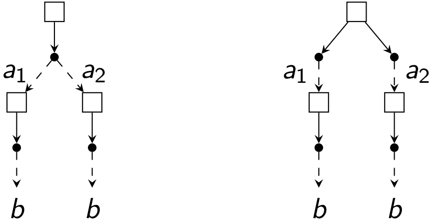
- A useful model for a controller (∃) in an environment (∀)
- Arose when considering Church's synthesis problem:
  - Take a specification φ, e.g. "terminate with a b"
  - Take a game modelling a controller in its environment



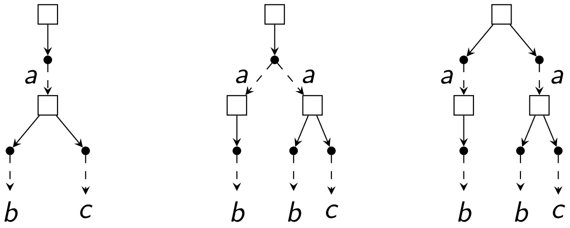


  - Implementation for the controller is a winning strategy for □
- A Markov decision process is a also a type of game, where the environment probabilistically updates.

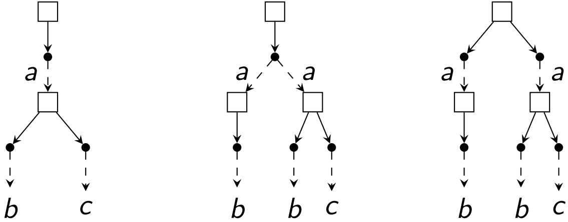
When are games trace equivalent?



When are games trace equivalent?



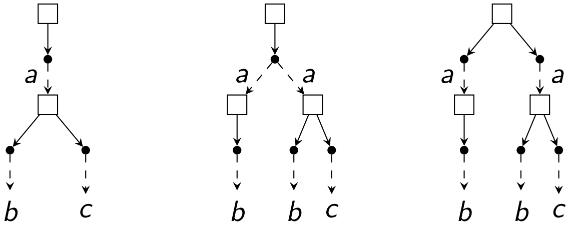
When are games trace equivalent?



Questions:

- Can we make sense of trace semantics in games?
- If so, what are the trace semantics at a state in a game?

When are games trace equivalent?



Questions:

- Can we make sense of trace semantics in games?
- If so, what are the trace semantics at a state in a game?

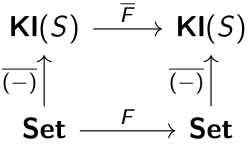
Goal: phrase synthesis coalgebraically to obtain general theory.

Outline

1. Choose a monad S and functor F to model two-player games, giving us a finite trace semantics map [Hasuo et al., 2007]:

$$\mathrm{tr} : \mathrm{States} \to S(\mathrm{Initial}_F)$$

2. Give a coalgebraic/categorical definition of strategies.
3. Characterising the trace map.

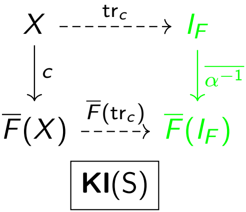


Theorem ([Hasuo et al., 2007])

- *A monad S : **Set** $\to$ **Set** such that*
  1. $\mathbf{Kl}(S)$ *is* ω-**cpo** *enriched*
  2. $S(0) \cong 1$
  3. *Zero maps* $X \to 0 \to Y$ *are bottom elements*
- *A functor F : **Set** $\to$ **Set** such that*
  1. *F lifts to* $\overline{F} : \mathbf{Kl}(S) \to \mathbf{Kl}(S)$
  2. $\overline{F}$ *is locally monotone*
  3. *The initial F-algebra* $\alpha : F(I) \to I$ *is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ *is the final* $\overline{F}$*-coalgebra.*

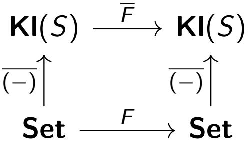


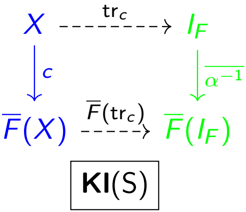
Theorem ([Hasuo et al., 2007])

- *A monad S : **Set** $\to$ **Set** such that*
  1. **Kl**(S) *is ω-**cpo** enriched*
  2. $S(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor F : **Set** $\to$ **Set** such that*
  1. *F lifts to $\overline{F}$:* **Kl**$(S) \to$ **Kl**(S)
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

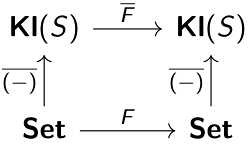
  $$\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.





This is Theorem 3.3 of [Hasuo et al., 2007], presented with slightly weaker assumptions like in [Jacobs, 2016]

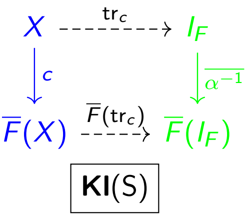


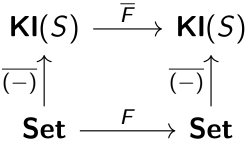
Theorem ([Hasuo et al., 2007])

- *A monad S : **Set** $\to$ **Set** such that*
  1. **Kl**(S) *is* ω-**cpo** *enriched*
  2. $S(0) \cong 1$
  3. *Zero maps* $X \to 0 \to Y$ *are bottom elements*
- *A functor F : **Set** $\to$ **Set** such that*
  1. *F lifts to* $\overline{F}$: **Kl**$(S) \to$ **Kl**(S)
  2. $\overline{F}$ *is locally monotone*
  3. *The initial F-algebra* $\alpha : F(I) \to I$ *is*

$$\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ *is the final* $\overline{F}$*-coalgebra.*



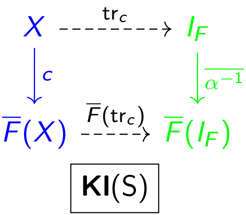


Theorem ([Hasuo et al., 2007])

- *A monad S :* **Set** $\to$ **Set** *such that*
  1. *$\mathbf{Kl}(S)$ is ω-**cpo** enriched*
  2. $S(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor F :* **Set** $\to$ **Set** *such that*
  1. *F lifts to $\overline{F} : \mathbf{Kl}(S) \to \mathbf{Kl}(S)$*
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

     $$\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.

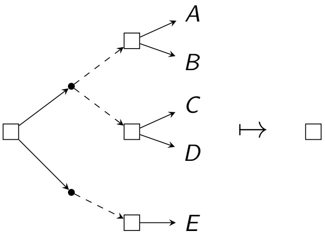


Finding a monad

- Can we combine two powerset monads PP : **Set** $\rightarrow$ **Set** for the controller and environment players in a game?

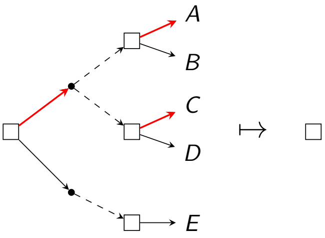
Finding a monad

- Can we combine two powerset monads PP : **Set** $\to$ **Set** for the controller and environment players in a game?
- The multiplication $\mu : PPPP(X) \to PP(X)$ should look something like:



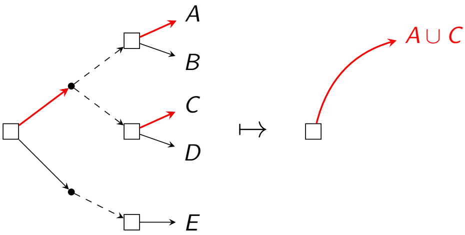
Finding a monad

- Can we combine two powerset monads PP : **Set** $\to$ **Set** for the controller and environment players in a game?
- The multiplication $\mu : PPPP(X) \to PP(X)$ should look something like:



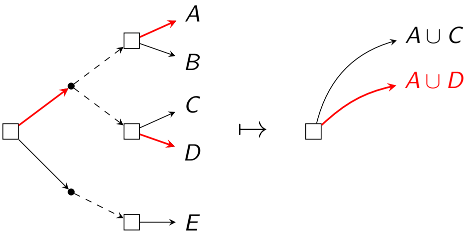
Finding a monad

- Can we combine two powerset monads $PP : \mathbf{Set} \to \mathbf{Set}$ for the controller and environment players in a game?
- The multiplication $\mu : PPPP(X) \to PP(X)$ should look something like:



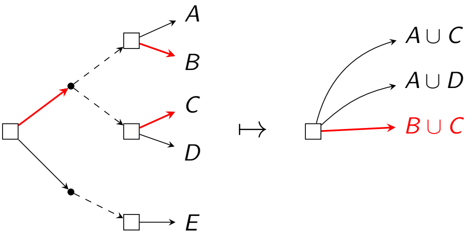
Finding a monad

- Can we combine two powerset monads PP : **Set** $\to$ **Set** for the controller and environment players in a game?
- The multiplication $\mu : PPPP(X) \to PP(X)$ should look something like:



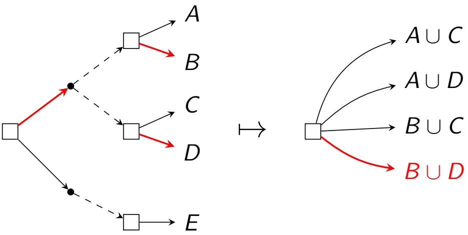
Finding a monad

- Can we combine two powerset monads PP : **Set** $\rightarrow$ **Set** for the controller and environment players in a game?
- The multiplication $\mu : PPPP(X) \rightarrow PP(X)$ should look something like:



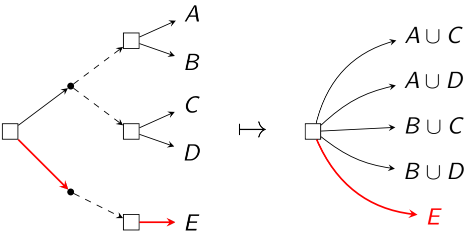
Finding a monad

- Can we combine two powerset monads PP : **Set** $\rightarrow$ **Set** for the controller and environment players in a game?
- The multiplication $\mu : PPPP(X) \rightarrow PP(X)$ should look something like:

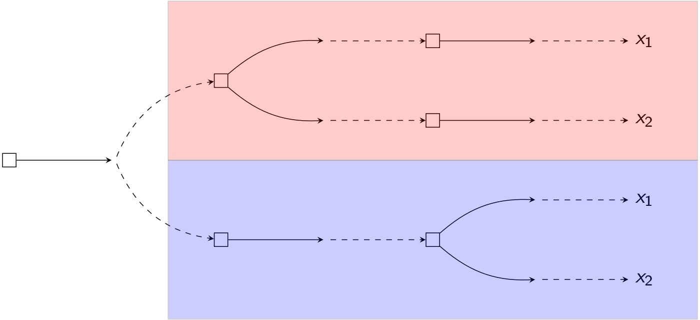


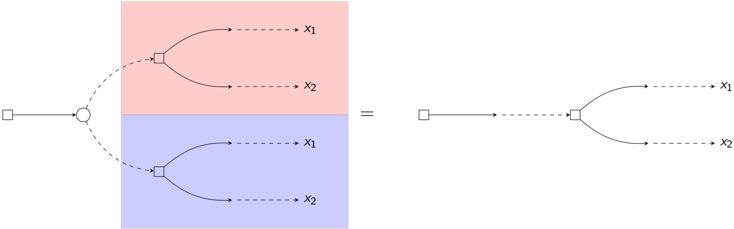
Finding a monad

- Can we combine two powerset monads $PP : \mathbf{Set} \to \mathbf{Set}$ for the controller and environment players in a game?
- The multiplication $\mu : PPPP(X) \to PP(X)$ should look something like:

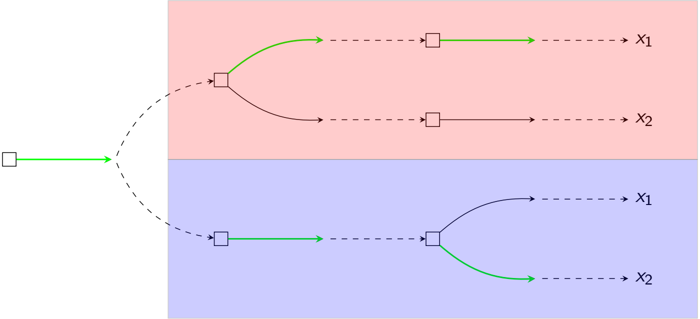


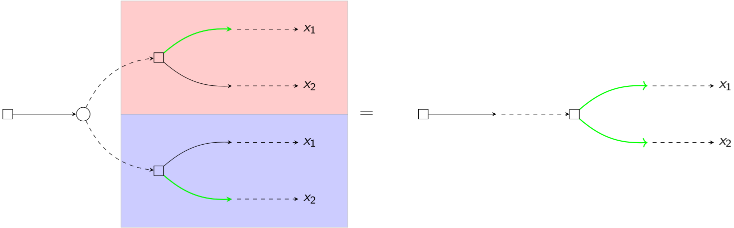
μ is not associative





μ is not associative





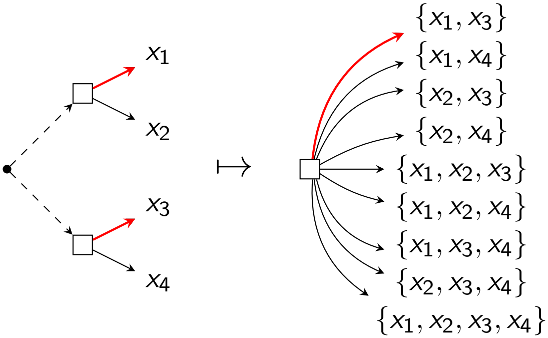
The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

- Allowing "convex choices" can be phrased in terms of a weak distributive law

This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

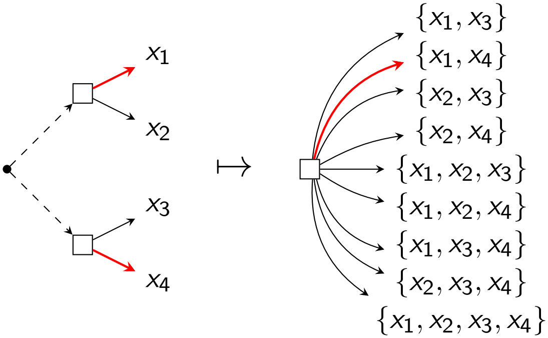
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

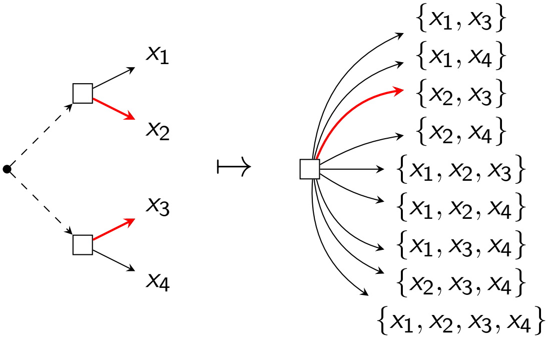
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

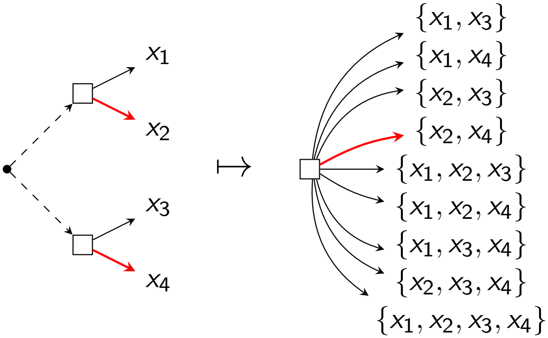
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

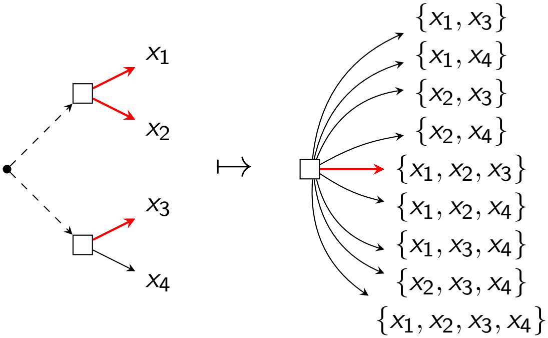
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

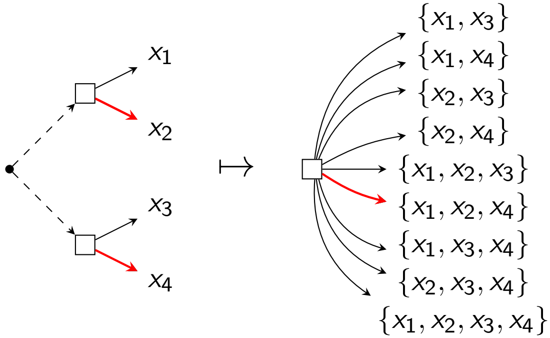
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

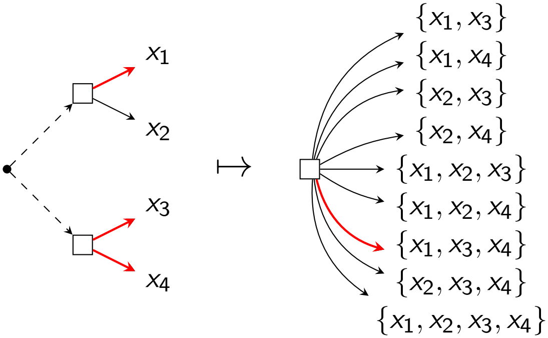
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

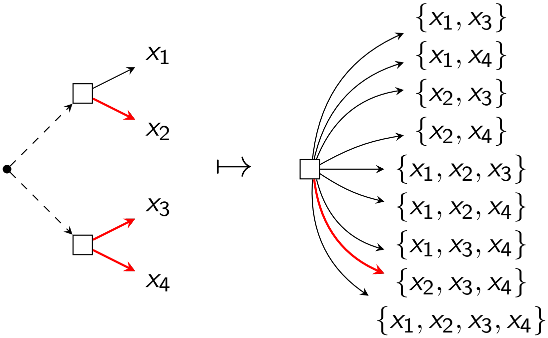
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

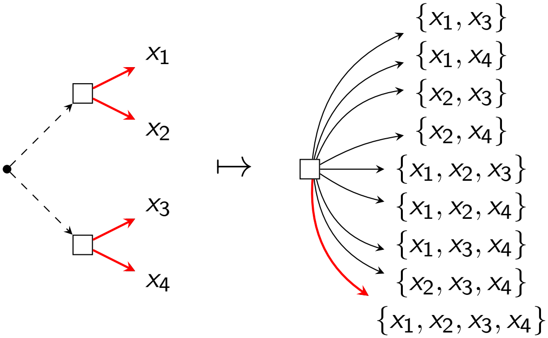
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $PP \to PP$ gives a monad $\widetilde{PP} : \mathbf{Set} \to \mathbf{Set}$

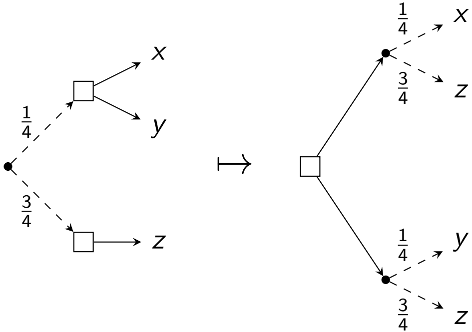
- Allowing "convex choices" can be phrased in terms of a weak distributive law
- Gives us a way of swapping environment-then-controller branching into controller-then-environment.



This law was used in [Goy, 2021] to model alternating automata, see [Garner, 2020] for composite monad construction

The law $DP \to PD$ gives a monad $\widetilde{PD} : \mathbf{Set} \to \mathbf{Set}$

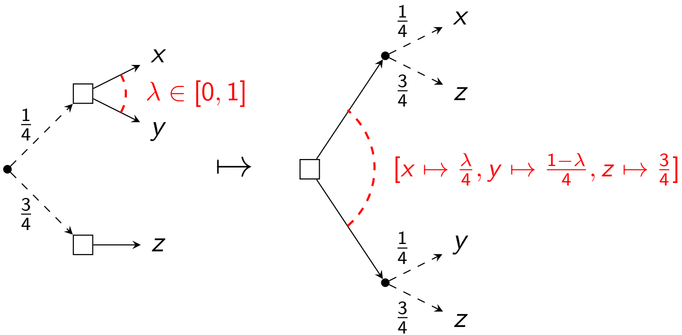
- Let D be the finite distribution monad.
- We also have a weak distributive law $DP \to PD$



Featured in [Goy and Petrişan, 2020], see [Bonchi and Santamaria, 2022] for discussions on convexity

The law $DP \to PD$ gives a monad $\widetilde{PD} : \mathbf{Set} \to \mathbf{Set}$

- Let D be the finite distribution monad.
- We also have a weak distributive law $DP \to PD$



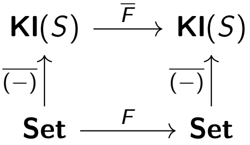
Featured in [Goy and Petrișan, 2020], see [Bonchi and Santamaria, 2022] for discussions on convexity

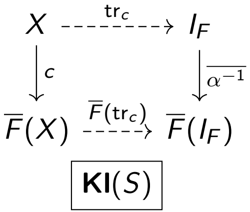
Theorem ([Hasuo et al., 2007])

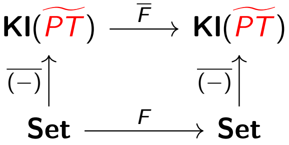
- *A monad $S : \mathbf{Set} \to \mathbf{Set}$ such that*
  1. $\mathbf{Kl}(S)$ *is* ω-**cpo** *enriched*
  2. $S(0) \cong 1$
  3. *Zero maps* $X \to 0 \to Y$ *are bottom elements*
- *A functor* $F : \mathbf{Set} \to \mathbf{Set}$ *such that*
  1. F *lifts to* $\overline{F} : \mathbf{Kl}(S) \to \mathbf{Kl}(S)$
  2. $\overline{F}$ *is locally monotone*
  3. *The initial* F*-algebra* $\alpha : F(I) \to I$ *is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ *is the final* $\overline{F}$*-coalgebra.*





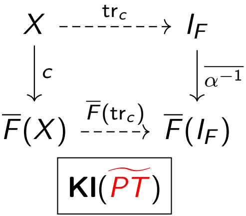


Theorem ([Hasuo et al., 2007])

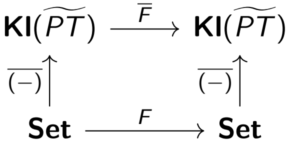
- *Let T : **Set** $\to$ **Set** be P or D*
  1. **Kl**$(\widetilde{PT})$ *is ω-**cpo** enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor F : **Set** $\to$ **Set** such that*
  1. *F lifts to $\overline{F}$:* **Kl**$(\widetilde{PT}) \to$ **Kl**$(\widetilde{PT})$
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

     $\operatorname{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.



The appropriate laws powerset over powerset laws exists by restricting $PP \to PP$, see [Goy, 2021, Section 3.1.2]

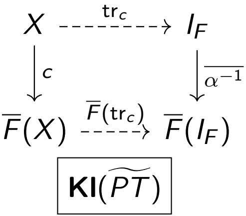


Theorem ([Hasuo et al., 2007])

- *Let $T : \mathbf{Set} \to \mathbf{Set}$ be P or D*
  1. $\mathbf{Kl}(\widetilde{PT})$ *is ω-**cpo** enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor $F : \mathbf{Set} \to \mathbf{Set}$ such that*
  1. *F lifts to $\overline{F} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$*
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.



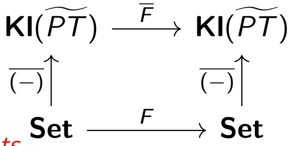
The appropriate laws powerset over powerset laws exists by restricting $PP \to PP$, see [Goy, 2021, Section 3.1.2]

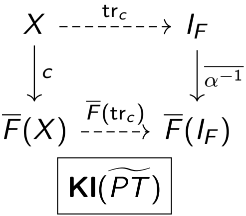
Theorem ([Hasuo et al., 2007])

- *Let* $T : \mathbf{Set} \to \mathbf{Set}$ *be* P *or* D
  1. $\mathbf{Kl}(\widetilde{PT})$ *is* ω-**cpo** *enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps* $X \to 0 \to Y$ *are bottom elements*
- *A functor* $F : \mathbf{Set} \to \mathbf{Set}$ *such that*
  1. F *lifts to* $\overline{F} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$
  2. $\overline{F}$ *is locally monotone*
  3. *The initial* F*-algebra* $\alpha : F(I) \to I$ *is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ *is the final* $\overline{F}$*-coalgebra.*





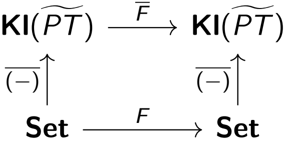
$PP(0) \cong P(1) \cong 2$

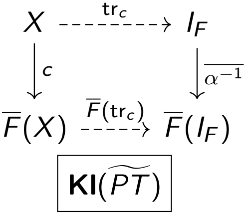
Theorem ([Hasuo et al., 2007])

- *Let T : **Set** $\to$ **Set** be P^+ or D*
  1. ***Kl***$(\widetilde{PT})$ *is ω-**cpo** enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor F : **Set** $\to$ **Set** such that*
  1. *F lifts to $\overline{F}$:* ***Kl***$(\widetilde{PT}) \to$ ***Kl***$(\widetilde{PT})$
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

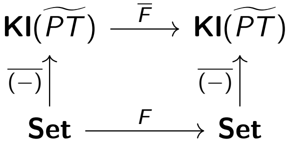
$$\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.





$PP^+(0) \cong P(0) \cong 1$

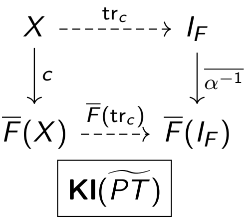


Theorem ([Hasuo et al., 2007])

- *Let $T : \mathbf{Set} \to \mathbf{Set}$ be P^+ or D*
  1. $\mathbf{Kl}(\widetilde{PT})$ *is ω-**cpo** enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor $F : \mathbf{Set} \to \mathbf{Set}$ such that*
  1. *F lifts to $\overline{F} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$*
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.

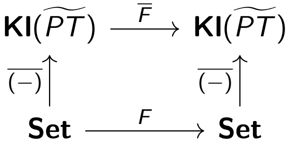


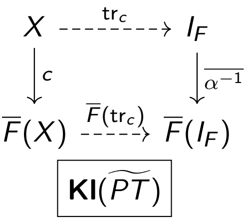
Theorem ([Hasuo et al., 2007])

- *Let* $T : \mathbf{Set} \to \mathbf{Set}$ *be* P_f^+ *or* D
  1. $\mathbf{Kl}(\widetilde{PT})$ *is* ω-**cpo** *enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps* $X \to 0 \to Y$ *are bottom elements*
- *A functor* $F : \mathbf{Set} \to \mathbf{Set}$ *such that*
  1. F *lifts to* $\overline{F} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$
  2. $\overline{F}$ *is locally monotone*
  3. *The initial* F*-algebra* $\alpha : F(I) \to I$ *is*

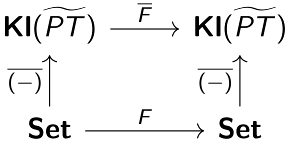
     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ *is the final* $\overline{F}$*-coalgebra.*





The appropriate laws powerset over powerset laws exists by restricting $PP \to PP$, see [Goy, 2021, Section 3.1.2]

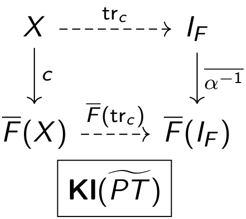


Theorem ([Hasuo et al., 2007])

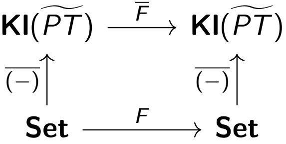
- *Let T :* **Set** $\to$ **Set** *be P_f^+ or D*
  1. $\mathbf{Kl}(\widetilde{PT})$ *is ω-**cpo** enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor F :* **Set** $\to$ **Set** *such that*
  1. *F lifts to $\overline{F} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$*
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.



The appropriate laws powerset over powerset laws exists by restricting $PP \to PP$, see [Goy, 2021, Section 3.1.2]

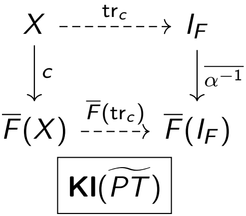


Theorem ([Hasuo et al., 2007])

- *Let T : **Set** $\to$ **Set** be P_f^+ or D*
  1. $\mathbf{Kl}(\widetilde{PT})$ *is ω-**cpo** enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *A functor F : **Set** $\to$ **Set** such that*
  1. *F lifts to $\overline{F}$: $\mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$*
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \to I$ is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.



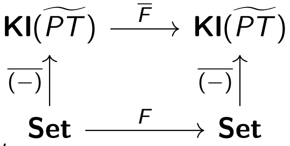
The appropriate laws powerset over powerset laws exists by restricting $PP \to PP$, see [Goy, 2021, Section 3.1.2]

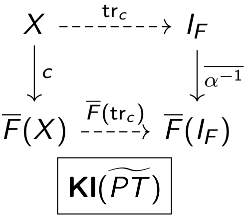
Theorem ([Hasuo et al., 2007])

- *Let T : **Set** $\rightarrow$ **Set** be P_f^+ or D*
  1. **Kl**$(\widetilde{PT})$ *is ω-**cpo** enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \rightarrow 0 \rightarrow Y$ are bottom elements*
- *A functor F : **Set** $\rightarrow$ **Set** such that*
  1. *F lifts to $\overline{F}$:* **Kl**$(\widetilde{PT}) \rightarrow$ **Kl**$(\widetilde{PT})$
  2. *$\overline{F}$ is locally monotone*
  3. *The initial F-algebra $\alpha : F(I) \rightarrow I$ is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \rightarrow \overline{F}(I_F)$ is the final $\overline{F}$-coalgebra.



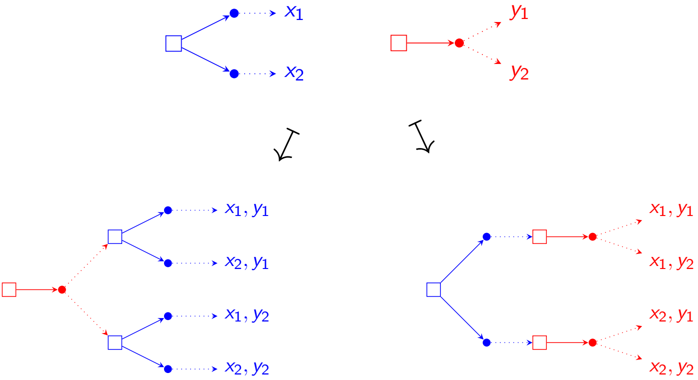


$\widetilde{PT}$ is commutative $\implies$ F lifts to $\overline{F}$

The appropriate laws powerset over powerset laws exists by restricting $PP \rightarrow PP$, see [Goy, 2021, Section 3.1.2]

$\widetilde{PT}$ is not commutative

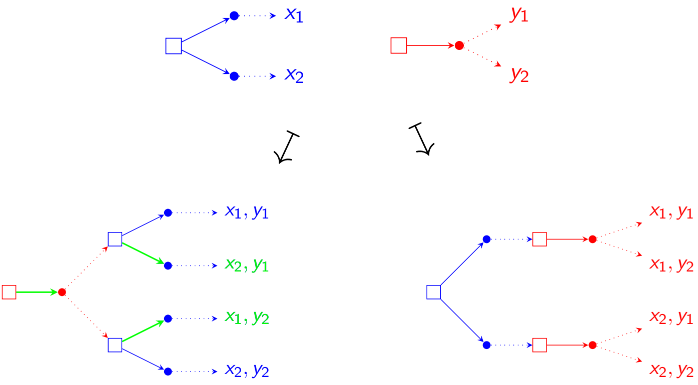
$$
\begin{array}{ccccc}
S(SX \times Y) & \xleftarrow{\text{stl}} & SX \times SY & \xrightarrow{\text{str}} & S(X \times SY) \\
\downarrow{\scriptstyle S(\text{str})} & & & & \downarrow{\scriptstyle S(\text{stl})} \\
SS(X \times Y) & \xrightarrow{\mu} & S(X \times Y) & \xleftarrow{\mu} & SS(X \times Y)
\end{array}
$$



Sets which arise from convexity are not shown on this slide (they make no difference to the counterexample).

$\widetilde{PT}$ is not commutative

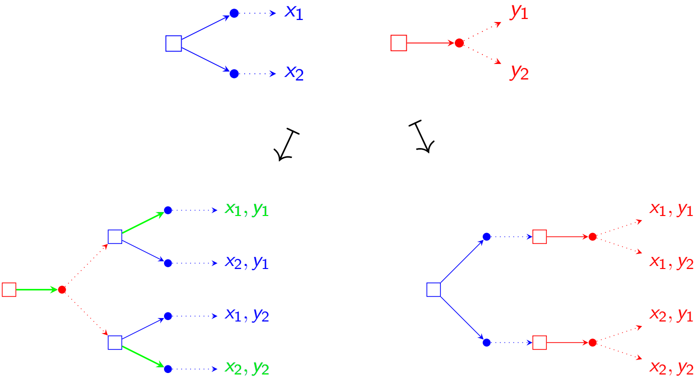
$$
\begin{array}{ccccc}
S(SX \times Y) & \xleftarrow{\mathsf{stl}} & SX \times SY & \xrightarrow{\mathsf{str}} & S(X \times SY) \\
\downarrow{\scriptstyle S(\mathsf{str})} & & & & \downarrow{\scriptstyle S(\mathsf{stl})} \\
SS(X \times Y) & \xrightarrow{\mu} & S(X \times Y) & \xleftarrow{\mu} & SS(X \times Y)
\end{array}
$$



Sets which arise from convexity are not shown on this slide (they make no difference to the counterexample).

$\widetilde{PT}$ is not commutative

$$\begin{array}{ccccc} S(SX \times Y) & \xleftarrow{\text{stl}} & SX \times SY & \xrightarrow{\text{str}} & S(X \times SY) \\ \downarrow S(\text{str}) & & & & \downarrow S(\text{stl}) \\ SS(X \times Y) & \xrightarrow{\mu} & S(X \times Y) & \xleftarrow{\mu} & SS(X \times Y) \end{array}$$



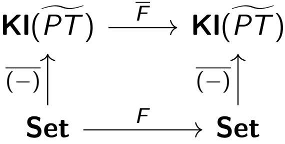
Sets which arise from convexity are not shown on this slide (they make no difference to the counterexample).

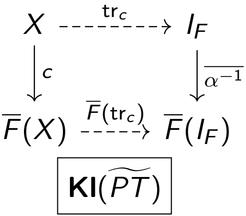
Theorem ([Hasuo et al., 2007])

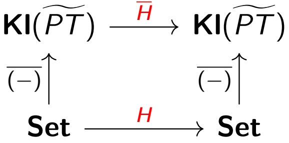
- *Let* $T : \mathbf{Set} \to \mathbf{Set}$ *be* P_f^+ *or* D
  1. $\mathbf{Kl}(\widetilde{PT})$ *is* ω-**cpo** *enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps* $X \to 0 \to Y$ *are bottom elements*
- *A functor* $F : \mathbf{Set} \to \mathbf{Set}$ *such that*
  1. F *lifts to* $\overline{F} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$
  2. $\overline{F}$ *is locally monotone*
  3. *The initial* F*-algebra* $\alpha : F(I_F) \to I_F$ *is*

     $\mathrm{colim}(0 \xrightarrow{!} F(0) \xrightarrow{F(!)} F^2(0) \xrightarrow{F^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : I_F \to \overline{F}(I_F)$ *is the final* $\overline{F}$*-coalgebra.*

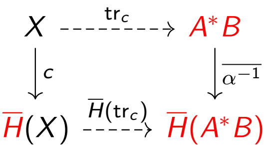






Theorem ([Hasuo et al., 2007])

- *Let $T : \mathbf{Set} \to \mathbf{Set}$ be P_f^+ or D*
  1. $\mathbf{Kl}(\widetilde{PT})$ *is* ω-**cpo** *enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps $X \to 0 \to Y$ are bottom elements*
- *Let $H : \mathbf{Set} \to \mathbf{Set}$ be $B + A \times (-)$*
  1. *H lifts to $\overline{H} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$*
  2. *$\overline{H}$ is locally monotone*
  3. *The initial H-algebra $\alpha : H(A^*B) \to A^*B$ is* $\mathrm{colim}(0 \xrightarrow{!} B \xrightarrow{H(!)} B + AB \xrightarrow{H^2(!)} \cdots)$



Then $\overline{\alpha^{-1}} : A^*B \to \overline{H}(A^*B)$ is the final $\overline{H}$-coalgebra.

$\mathbf{Kl}(\widetilde{PT})$

Theorem ([Hasuo et al., 2007])

- *Let* $T : \mathbf{Set} \to \mathbf{Set}$ *be* P_f^+ *or* D
  1. $\mathbf{Kl}(\widetilde{PT})$ *is* ω-**cpo** *enriched*
  2. $\widetilde{PT}(0) \cong 1$
  3. *Zero maps* $X \to 0 \to Y$ *are bottom elements*
- *Let* $H : \mathbf{Set} \to \mathbf{Set}$ *be* $B + A \times (-)$
  1. H *lifts to* $\overline{H} : \mathbf{Kl}(\widetilde{PT}) \to \mathbf{Kl}(\widetilde{PT})$
  2. $\overline{H}$ *is locally monotone*
  3. *The initial* H*-algebra* $\alpha : H(A^*B) \to A^*B$ *is* $\mathrm{colim}(0 \xrightarrow{!} B \xrightarrow{H(!)} B + AB \xrightarrow{H^2(!)} \cdots)$

Then $\overline{\alpha^{-1}} : A^*B \to \overline{H}(A^*B)$ *is the final* $\overline{H}$*-coalgebra.*

$$\begin{array}{ccc} \mathbf{Kl}(\widetilde{PT}) & \xrightarrow{\overline{H}} & \mathbf{Kl}(\widetilde{PT}) \\ \uparrow \overline{(-)} & & \uparrow \overline{(-)} \\ \mathbf{Set} & \xrightarrow{H} & \mathbf{Set} \end{array}$$

$$\begin{array}{ccc} X & \overset{\mathrm{tr}_c}{\dashrightarrow} & A^*B \\ \downarrow c & & \downarrow \overline{\alpha^{-1}} \\ \overline{H}(X) & \overset{\overline{H}(\mathrm{tr}_c)}{\dashrightarrow} & \overline{H}(A^*B) \end{array}$$

$\boxed{\mathbf{Kl}(\widetilde{PT})}$

Games live in $\mathbf{Kl}(\widetilde{PT})$

$$c : X \to \widetilde{PT}H(X)$$

- For $T = P_f^+$, these are games with a nondeterministic environment
- For $T = D$, these are games with a stochastic environment
- Both are morphisms $X \to \overline{H}(X)$ in $\mathbf{Kl}(\widetilde{PT})$

Games live in $\mathbf{Kl}(\widetilde{PT})$

$$c : X \to \widetilde{PT}H(X)$$

- For $T = P_f^+$, these are games with a nondeterministic environment
- For $T = D$, these are games with a stochastic environment
- Both are morphisms $X \to \overline{H}(X)$ in $\mathbf{Kl}(\widetilde{PT})$
- Where do strategies live?
  - A strategy picks some $TH(X)$ in $c(x)$
  - Strategies live in $\mathbf{Kl}(T)$

Categorical definition of a strategy

- Strategies depend on past states, as well as behaviour.

Categorical definition of a strategy

- Strategies depend on past states, as well as behaviour.
- Define $H_X(-) := X \times (B + A \times -)$,

Categorical definition of a strategy

- Strategies depend on past states, as well as behaviour.
- Define $H_X(-) := X \times (B + A \times -)$,
- $H_X^n(X)$ contains partial plays of length n and completed plays (ending in a b) of length less than n.

Categorical definition of a strategy

- Strategies depend on past states, as well as behaviour.
- Define $H_X(-) := X \times (B + A \times -)$,
- $H_X^n(X)$ contains partial plays of length n and completed plays (ending in a b) of length less than n.

Definition

A strategy is a chain of maps $\{\sigma_n\}_{n \in \omega}$

- $\sigma_0 : 1 \to TX$ picks an initial state
- $\sigma_{n+1} : \mathrm{Im}(\sigma_n) \to TH_X^{n+1}(X)$ extends plays

such that:

- We only extend partial plays
- Successors are chosen from the game

Categorical definition of a strategy

- Strategies depend on past states, as well as behaviour.
- Define $H_X(-) := X \times (B + A \times -)$,
- $H_X^n(X)$ contains partial plays of length n and completed plays (ending in a b) of length less than n.

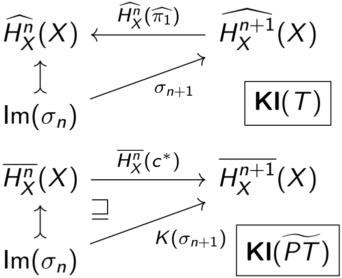
Definition

A strategy is a chain of maps $\{\sigma_n\}_{n \in \omega}$

- $\sigma_0 : 1 \to TX$ picks an initial state
- $\sigma_{n+1} : \mathrm{Im}(\sigma_n) \to TH_X^{n+1}(X)$ extends plays

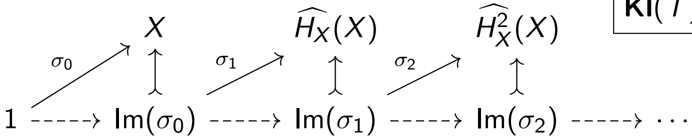
such that:

- We only extend partial plays
- Successors are chosen from the game

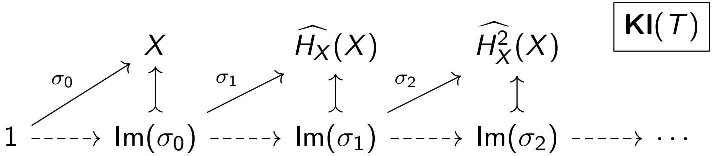


Strategy outcomes

KI(T)

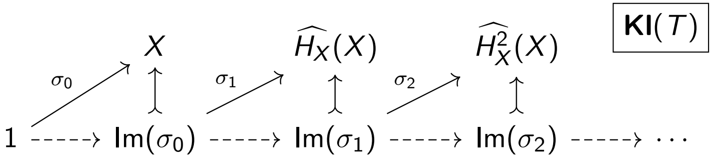


Strategy outcomes



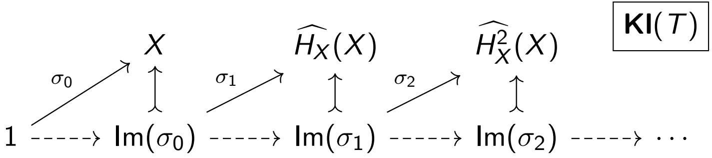
- Composing along the bottom and then up we can obtain an *n-step partial outcome* $\mathrm{plays}_n^\sigma : 1 \to TH_X^n(X)$

Strategy outcomes

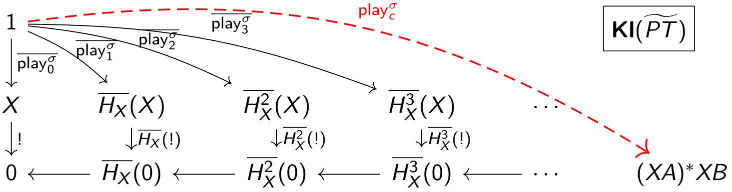


- Composing along the bottom and then up we can obtain an *n-step partial outcome* $\text{plays}^{\sigma}_{n} : 1 \to TH^{n}_{X}(X)$
- For a full outcome, we want map into $(XA)^{*}XB$

Strategy outcomes



- Composing along the bottom and then up we can obtain an *n-step partial outcome* $\mathrm{plays}^\sigma_n : 1 \to TH^n_X(X)$
- For a full outcome, we want map into $(XA)^*XB$
- We know $(XA)^*XB$ is the limit of the final sequence in $\mathbf{Kl}(\widetilde{PT})$



What is in the trace map?

$$\begin{array}{ccc} X & \overset{\mathrm{tr}_c}{\dashrightarrow} & A^*B \\ \downarrow c & & \downarrow \overline{\alpha^{-1}} \\ \overline{H}(X) & \overset{\overline{H}(\mathrm{tr}_c)}{\dashrightarrow} & \overline{H}(A^*B) \end{array}$$

- $\mathrm{tr}_c : X \to \widetilde{PT}(A^*B)$
- $\mathrm{plays}_c^\sigma : 1 \to \widetilde{PT}((XA)^*XB)$
- Let tr_c^σ be plays_c^σ, followed by forgetting state information

What is in the trace map?

$$\begin{array}{ccc} X & \overset{\mathrm{tr}_c}{\dashrightarrow} & A^*B \\ \downarrow c & & \downarrow \overline{\alpha^{-1}} \\ \overline{H}(X) & \overset{\overline{H}(\mathrm{tr}_c)}{\dashrightarrow} & \overline{H}(A^*B) \end{array}$$

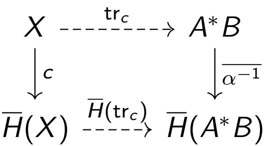
- $\mathrm{tr}_c : X \to \widetilde{PT}(A^*B)$
- $\mathrm{plays}_c^\sigma : 1 \to \widetilde{PT}((XA)^*XB)$
- Let tr_c^σ be plays_c^σ, followed by forgetting state information

Theorem

The trace semantics at x are the union of outcomes of strategies starting in x

$$\mathrm{tr}_c(x) = \bigcup_{\sigma \text{ starts in } x} \mathrm{tr}_c^\sigma$$

What is in the trace map?



- $\mathrm{tr}_c : X \to \widetilde{PT}(A^*B)$
- $\mathrm{plays}_c^\sigma : 1 \to \widetilde{PT}((XA)^*XB)$
- Let tr_c^σ be plays_c^σ, followed by forgetting state information

Theorem

The trace semantics at x are the union of outcomes of strategies starting in x

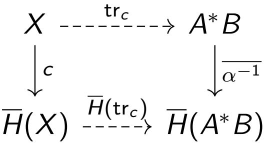
$$\mathrm{tr}_c(x) = \bigcup_{\sigma \text{ starts in } x} \mathrm{tr}_c^\sigma$$

Lemma

Unfolding the game n times from x, is equal to the n-step partial traces of n-step strategies starting in x

$$H^{n-1}(c) \circ \cdots \circ H(c) \circ c(x) = \{\mathrm{tr}_n^\sigma \mid \{\sigma_i\}_{i \leq n} \text{ starts in } x\}$$

What is in the trace map?



- $\mathrm{tr}_c : X \to \widetilde{PT}(A^*B)$
- $\mathrm{plays}_c^\sigma : 1 \to \widetilde{PT}((XA)^*XB)$
- Let tr_c^σ be plays_c^σ, followed by forgetting state information

Theorem

The trace semantics at x are the union of outcomes of strategies starting in x For any $U \in T(A^*B)$:

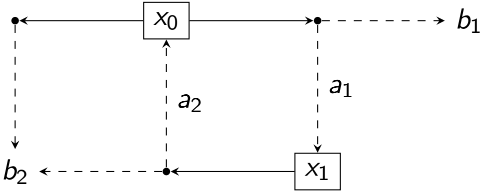
$$U \in \mathrm{tr}_c(x) \iff \text{there is a strategy which forces } U$$

Lemma

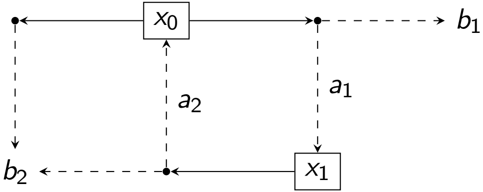
Unfolding the game n times from x, is equal to the n-step partial traces of n-step strategies starting in x

$$H^{n-1}(c) \circ \cdots \circ H(c) \circ c(x) = \{\mathrm{tr}_n^\sigma \mid \{\sigma_i\}_{i \leq n} \text{ starts in } x\}$$

Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U

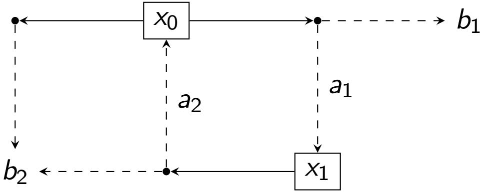


Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U



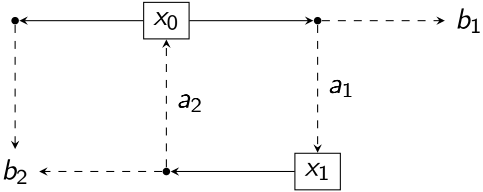
$\{b_1\}$

Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U



$\{b_1\}$ No

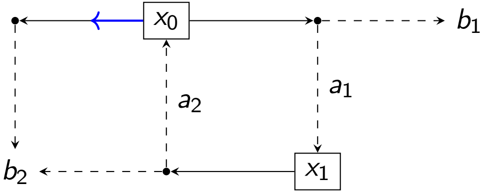
Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U



$\{b_1\}$ No

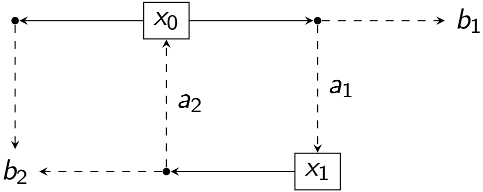
$\{b_2\}$

Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U



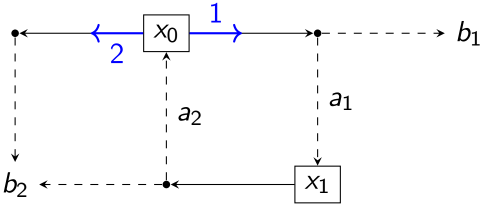
$\{b_1\}$	No
$\{b_2\}$	Yes

Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U



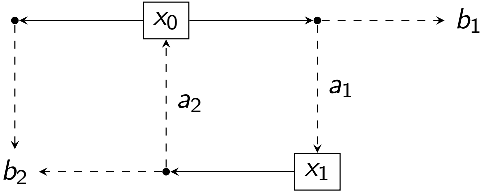
$\{b_1\}$	No
$\{b_2\}$	Yes
$\{a_1 a_2 b_2, a_1 b_2, b_1\}$	

Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U



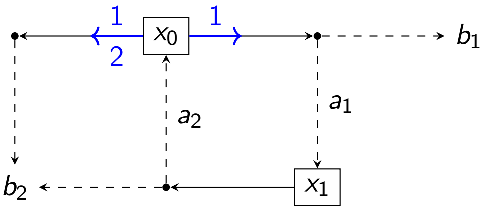
$\{b_1\}$	No
$\{b_2\}$	Yes
$\{a_1 a_2 b_2, a_1 b_2, b_1\}$	Yes

Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U

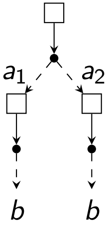


$\{b_1\}$	No
$\{b_2\}$	Yes
$\{a_1 a_2 b_2, a_1 b_2, b_1\}$	Yes
$\{a_1 a_2 b_2, a_1 b_2, b_1, b_2\}$	

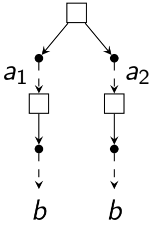
Example: $U \in \mathrm{tr}_c(x_0) \iff$ a strategy forces U



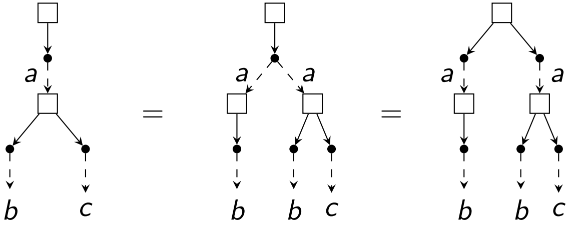
$\{b_1\}$	No
$\{b_2\}$	Yes
$\{a_1 a_2 b_2, a_1 b_2, b_1\}$	Yes
$\{a_1 a_2 b_2, a_1 b_2, b_1, b_2\}$	Yes



$\subseteq$

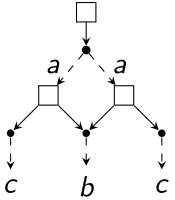


$\{\{a_1 b, a_2 b\}\}$ $\{\{a_1 b\}, \{a_2 b\}, \{a_1 b, a_2 b\}\}$

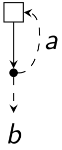


$$\{\{ab\}, \{ac\}, \{ab, ac\}\}$$

Conclusion



$\{\{ab\}, \{ac\}, \{ab, ac\}\}$



$\emptyset$

This talk:

- Obtained a finite trace semantics map
- Defined strategies in $\mathbf{Kl}(T)$
- Characterised the trace map in terms of strategies

Future work

- Product construction with an automaton
- Infinite traces

Bibliography I

Bonchi, F. and Santamaria, A. (2022).
Convexity via Weak Distributive Laws.
Logical Methods in Computer Science, Volume 18, Issue 4:8389.
arXiv:2108.10718 [cs, math].

Garner, R. (2020).
The Vietoris Monad and Weak Distributive Laws.
Applied Categorical Structures, 28(2):339–354.

Goy, A. (2021).
On the compositionality of monads via weak distributive laws.
phdthesis, Université Paris-Saclay.

Bibliography II

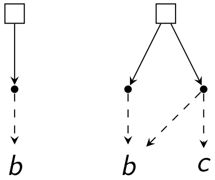
Goy, A. and Petrişan, D. (2020).
Combining probabilistic and non-deterministic choice via weak distributive laws.
In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 454–464, New York, NY, USA. Association for Computing Machinery.

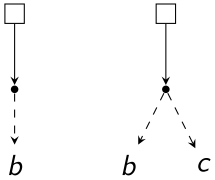
Hasuo, I., Jacobs, B., and Sokolova, A. (2007).
Generic trace semantics via coinduction.
Logical Methods in Computer Science, Volume 3, Issue 4.

Jacobs, B. (2016).
Introduction to Coalgebra: Towards Mathematics of States and Observation.
Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Bibliography III

Klin, B. and Salamanca, J. (2018).
Iterated covariant powerset is not a monad.
Electronic Notes in Theoretical Computer Science, 341:261–276.
Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).


b
b
c


b
b
c