Diegetic representation of feedback in open games

Matteo Capucci

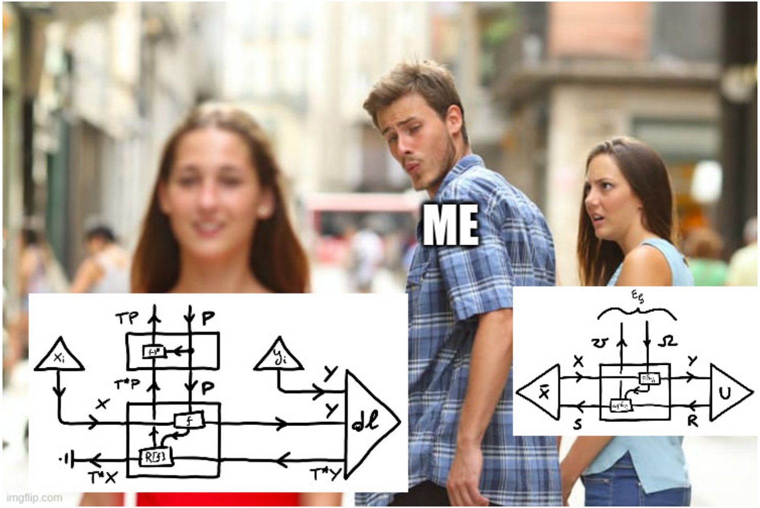
Mathematically Structured Programming group,
Department of Computer and Information Sciences,
University of Strathclyde

5th International Conference on Applied Category Theory,

Glasgow (Scotland)

July 21st, 2022

Meme abstract



What is a game?

Game theory is the mathematical study of interaction among independent, self-interested agents.

– Essentials of Game Theory, [1]

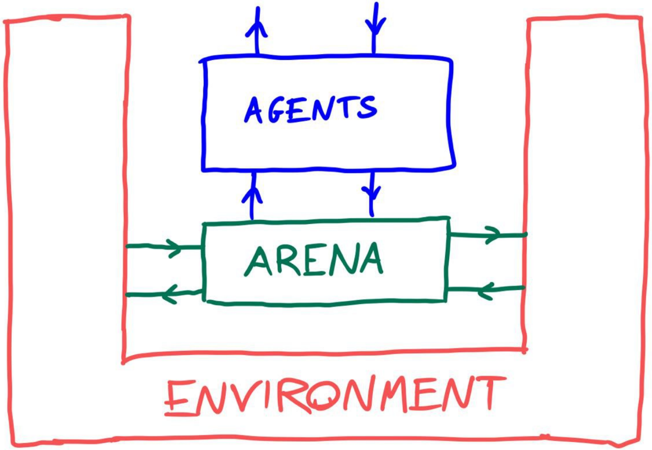
What is a game?

Game theory is the mathematical study of interaction among independent, self-interested agents.
– Essentials of Game Theory, [1]

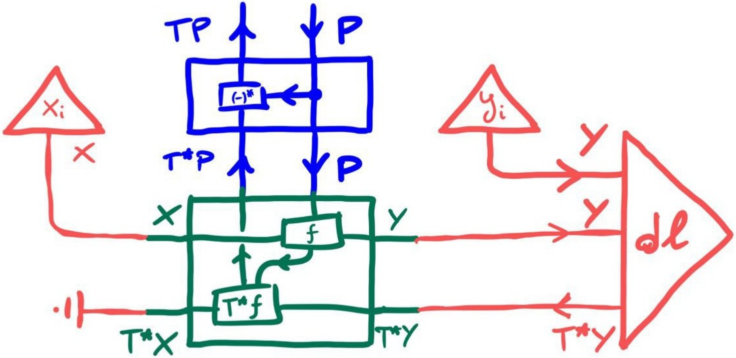
A game factors in two parts:

1. An **arena**, which models the **dynamics** of the game.
2. Some **players**, which intervene in the arena by making **decisions**.

The idea behind parametric-optics-as-cybernetic-systems [2] is 'players in arenas' is a rough description of many other kinds of systems, including learners, Bayesian reasoners, control problems, etc.

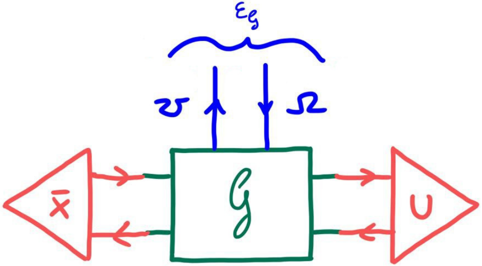


Indeed, looking at gradient-based learners:

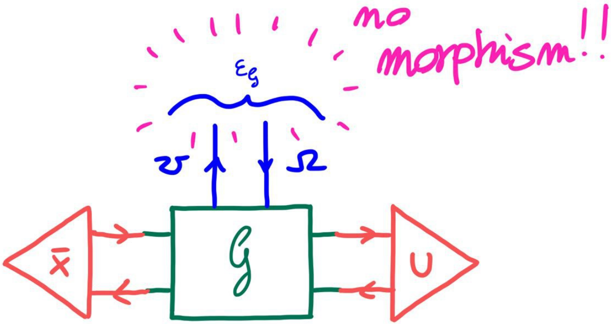


This is further corroborated by how these things compose, actually.

Contrast this with **'open games with agency'**:



Contrast this with **'open games with agency'**:



Players are **extra-diegetic**: a player's counterfactual analysis of the game happens *outside* of the system.

Outline

Goal of the work: try to understand and fix this situation.

1. Why do learners exhibit 'diegetic agency'?
2. Can we imitate this in games?

Outline

Goal of the work: try to understand and fix this situation.

1. Why do learners exhibit 'diegetic agency'?
2. Can we imitate this in games?

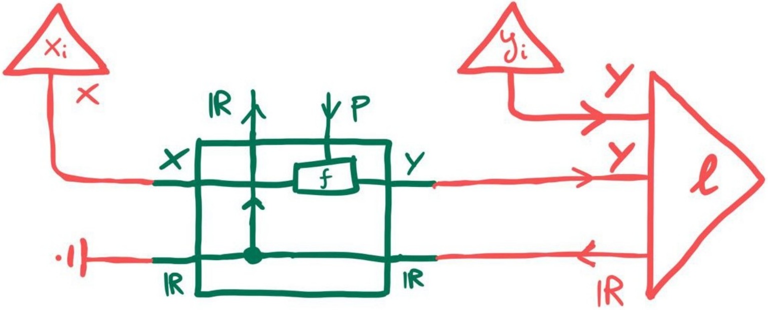
Results:

1. An understanding of **learners and games as second-order cybernetic systems**, whose mathematical structure naturally leads to backprop/backward induction
2. A **diegetic**, dynamical **view of players** obtained by reverse-mode differentiation of general parametric lenses
3. An important lax monoidal structure, the **Nashator**, shining light on game-theoretical phenomena

Learners

Model vs training dynamics

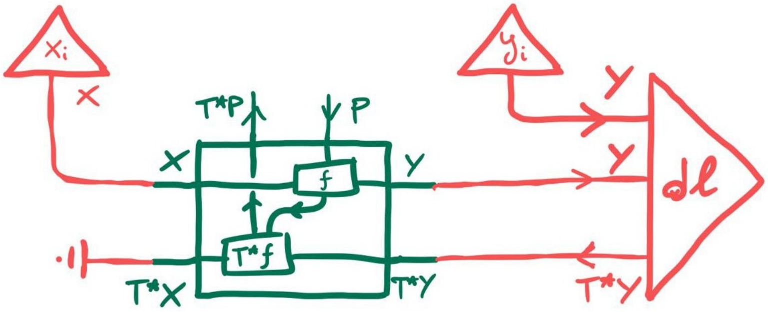
Question: Why don't we consider this to be a learner's arena?



This is the **model dynamics**: inputs and parameters go in, losses come out.

Model vs training dynamics

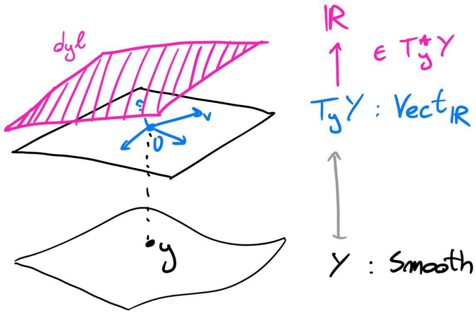
Answer: we actually want to describe the 'counterfactual' dynamics.



This is the **training dynamics**: inputs and parameters go in, *loss differentials* come out.

Backpropagation, conceptually

Feedback about $y \in Y$ is given **to its entire 'infinitesimal neighbourhood'** $T_y Y$

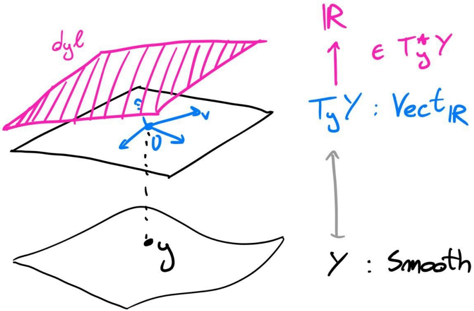


It is a counterfactual evaluation:

How much would ℓ change if I were to move away from y in a given direction $v \in T_y Y$?

Backpropagation, conceptually

Feedback about $y \in Y$ is given **to its entire 'infinitesimal neighbourhood'** $T_y Y$

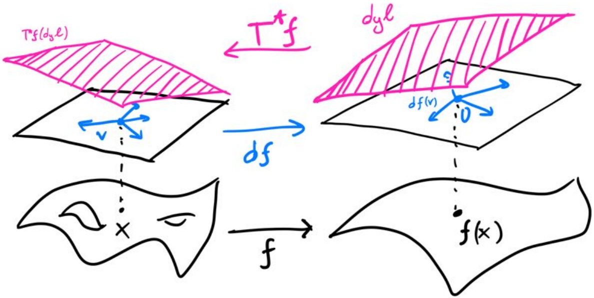


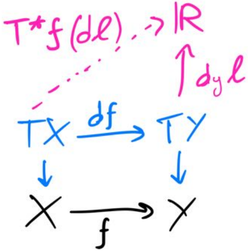
$d\ell \in T^*Y$ is a **covector over** Y:

$$\begin{matrix} T^*Y \\ \downarrow \\ Y \end{matrix} = \left\{ \begin{matrix} TY & \xrightarrow{\xi} & \mathbb{R} \times Y \\ & \searrow \quad \swarrow_{\pi_Y} & \\ & Y & \end{matrix} \;\middle|\; \xi_y : T_y Y \to \mathbb{R} \text{ is linear for all } y \in Y \right\}$$

Backpropagation, conceptually

T^* also acts *contravariantly* on maps, yielding reverse derivatives:



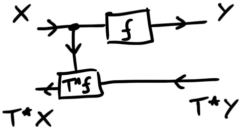


Backpropagation, functorially

The latter assignment yields a functor

$$\mathbf{Smooth} \xrightarrow{\quad T^* \quad} \mathbf{DLens}(\mathbf{Vect}_{\mathbb{R}})$$

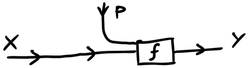
$\longmapsto$



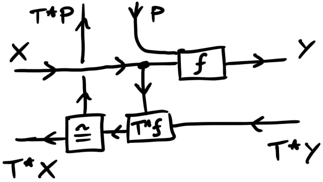
Backpropagation, functorially

However, we are interested in *parametric* smooth functions:

$$\mathbf{Para}(\mathbf{Smooth}) \xrightarrow{\mathbf{Para}(T^*)} \mathbf{Para}(\mathbf{DLens}(\mathbf{Vect}_{\mathbb{R}}))$$



$\longmapsto$

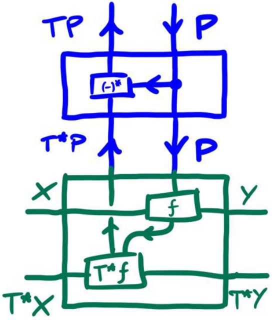


where $\cong$ is the **strong monoidal stucture of** T^*

$$\cong : T^*(P \times X) \longrightarrow T^*P \oplus T^*X$$

Gradients from covectors

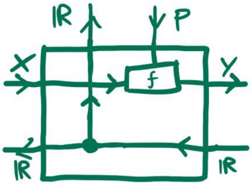
Once an arena is given, a learner is a system that converts a covector to a vector:



A morphism like $(-)^\sharp : T^*X \to TX$ (a 2-form) is induced by **Riemannian metrics, symplectic and Poisson structures** (relevant in optimization theory & physics)

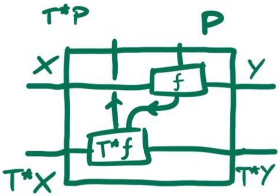
Recap

1. Start with a model



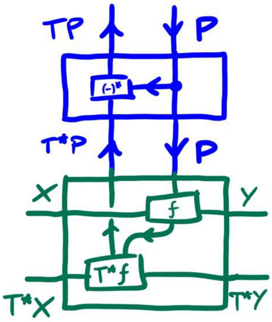
Recap

2. Get the arena by applying $\mathbf{Para}(T^*)$



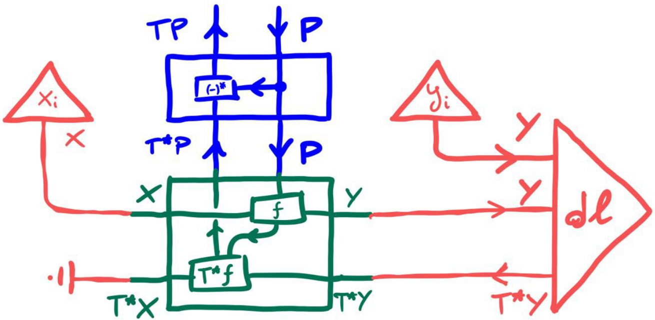
Recap

3. Add the learner



Recap

4. Surround with a training apparatus (training data + d(loss))



States, changes, values and valuations

What did we use to do this?

States, changes, values and valuations

What did we use to do this? Remember how $\mathrm{d}\ell$ arisen:

States, changes, values and valuations

What did we use to do this? Remember how $\mathrm{d}\ell$ arisen:

it's a **valuation in** $\mathbb{R}$

States, changes, values and valuations

What did we use to do this? Remember how $\mathrm{d}\ell$ arisen:

it's a **valuation in** $\mathbb{R}$
of possible changes $T_y Y$

States, changes, values and valuations

What did we use to do this? Remember how $\mathrm{d}\ell$ arisen:

it's a **valuation in** $\mathbb{R}$
of possible changes $T_y Y$
from a given state $y \in Y$

States, changes, values and valuations

	learners	games
states	space $Y : \mathbf{Smooth}$	

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{DLens}(\Delta_{\mathbb{R}})$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$		

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$		

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoffs	$R : \mathbf{Set}/_{\mathrm{prj}}Y$

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoffs	$R : \mathbf{Set}/_{\mathrm{prj}}Y$
	loss map	$\ell : Y \to \mathbb{R}$		

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoffs	$R : \mathbf{Set}/_{\mathrm{prj}}Y$
	loss map	$\ell : Y \to \mathbb{R}$	payoff map	$u : Y \to R$

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoffs	$R : \mathbf{Set}/_{\mathrm{prj}}Y$
	loss map	$\ell : Y \to \mathbb{R}$	payoff map	$u : Y \to R$
	loss diff.	$\mathrm{d}\ell : (y : Y) \to (T_y Y \to \mathbb{R})$		

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoffs	$R : \mathbf{Set}/_{\mathrm{prj}}Y$
	loss map	$\ell : Y \to \mathbb{R}$	payoff map	$u : Y \to R$
	loss diff.	$\mathrm{d}\ell : (y : Y) \to (T_y Y \to \mathbb{R})$		$\Delta u : Y \to (Y \to R)$
			const	$\bar{y} \mapsto u$
			regret	$\bar{y} \mapsto \lambda y \,.\, u(y) - u(\bar{y})$

States, changes, values and valuations

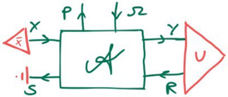
	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoffs	$R : \mathbf{Set}/_{\mathrm{prj}}Y$
	loss map	$\ell : Y \to \mathbb{R}$	payoff map	$u : Y \to R$
	loss diff.	$\mathrm{d}\ell : (y : Y) \to (T_yY \to \mathbb{R})$		$\Delta u : Y \to (Y \to R)$
			const	$\bar{y} \mapsto u$
			regret	$\bar{y} \mapsto \lambda y \,.\, u(y) - u(\bar{y})$
valuations	covectors	$T^*Y : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$		

States, changes, values and valuations

	learners		games	
states	space	$Y : \mathbf{Smooth}$	moves & payoffs	$(Y, R) : \mathbf{Lens}(\mathbf{Set})$
changes	tangent vectors	$TY : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	deviations	$Y : \mathbf{Set}$
values	losses	$\mathbb{R} : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoffs	$R : \mathbf{Set}/_{\mathrm{prj}}Y$
	loss map	$\ell : Y \to \mathbb{R}$	payoff map	$u : Y \to R$
	loss diff.	$\mathrm{d}\ell : (y : Y) \to (T_y Y \to \mathbb{R})$		$\Delta u : Y \to (Y \to R)$
			const	$\bar{y} \mapsto u$
			regret	$\bar{y} \mapsto \lambda y \,.\, u(y) - u(\bar{y})$
valuations	covectors	$T^*Y : \mathbf{Vect}_R(\mathbf{Smooth}/Y)$	payoff maps	$R^Y : \mathbf{Set}/_{\mathrm{prj}}Y$

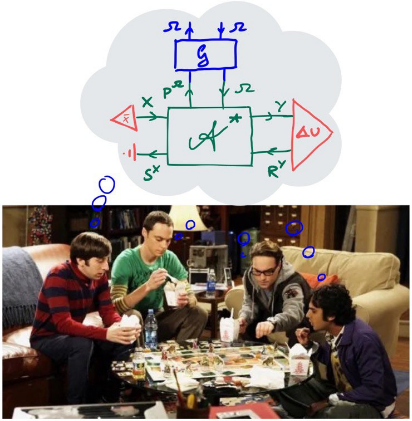
Idea

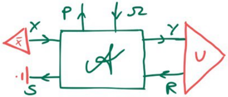
Like learners, games have a ‘model dynamics’ and a ‘counterfactual dynamics’.
We specify the first, but players live in the latter and that’s what we are interested with:



Idea

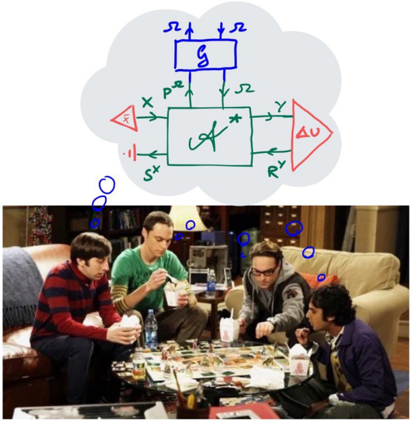
Like learners, games have a ‘model dynamics’ and a ‘counterfactual dynamics’.
We specify the first, but players live in the latter and that’s what we are interested with:

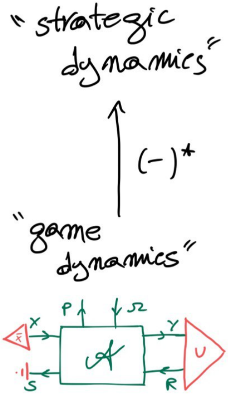




Idea

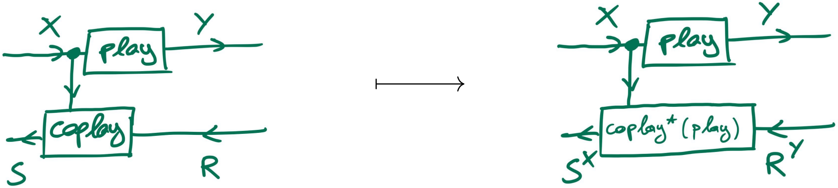
Like learners, games have a ‘model dynamics’ and a ‘counterfactual dynamics’.
We specify the first, but players live in the latter and that’s what we are interested with:



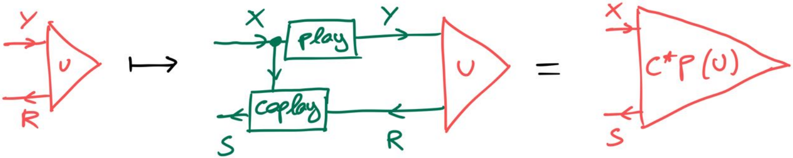


As before, valuations give a 'reverse-mode differentiation', sending a map of game states to its counterfactual dynamics:

$$\mathbf{Lens}(\mathbf{Set}) \xrightarrow{\quad (-)^* \quad} \mathbf{Lens}(\mathbf{Set})$$



where $\text{coplay}^*(\text{play})(u) = \lambda x\,.\,\text{coplay}(x, u(\text{play}(x)))$, i.e.:



Nashator

Crucially, the functor $(-)^*$ can be equipped with a lax monoidal structure.
Given $\mathbf{X} = (X, S)$, $\mathbf{Y} = (Y, R)$:

$$(1_{\mathbf{X}\times\mathbf{Y}}, \mathbf{n}_{\mathbf{X},\mathbf{Y}}) : (X \times Y, S^X \times R^Y) \rightleftarrows (X \times Y, (S \times R)^{X \times Y})$$

where the backward part is the **Nashator**:

$$\begin{aligned} \mathbf{n}_{\mathbf{X},\mathbf{Y}} : X \times Y \times (S \times R)^{X \times Y} &\longrightarrow S^X \times R^Y \\ (\bar{x},\ \bar{y},\ u) \quad &\longmapsto \langle\ \lambda x \,.\, u_S(x, \bar{y}),\ \ \lambda y \,.\, u_R(\bar{x}, y)\ \rangle \end{aligned}$$

Nashator

Crucially, the functor $(-)^*$ can be equipped with a lax monoidal structure.
Given $\mathbf{X} = (X, S)$, $\mathbf{Y} = (Y, R)$:

$$(1_{\mathbf{X}\times\mathbf{Y}}, \mathbf{n}_{\mathbf{X},\mathbf{Y}}) : (X \times Y, S^X \times R^Y) \rightleftarrows (X \times Y, (S \times R)^{X\times Y})$$

where the backward part is the **Nashator**:

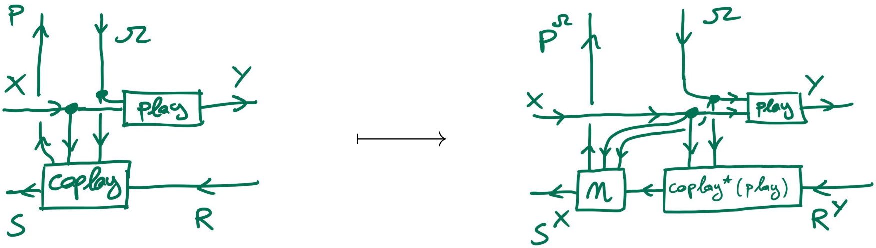
$$\begin{aligned} \mathbf{n}_{\mathbf{X},\mathbf{Y}} : X \times Y \times (S \times R)^{X\times Y} &\longrightarrow S^X \times R^Y \\ (\bar{x},\ \bar{y},\ u) &\longmapsto \langle\ \lambda x\,.\,u_S(x, \bar{y}),\ \lambda y\,.\,u_R(\bar{x}, y)\ \rangle \end{aligned}$$

Notice this monoidal structure can only be defined because $(-)^*$ lands in lenses!

Nashator

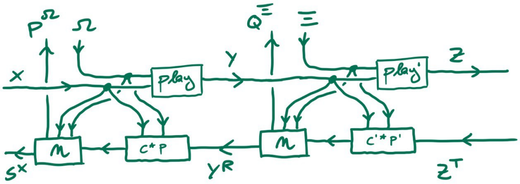
We can use the Nashator to make $\mathbf{Para}((-)^*)$ into a *lax 2-functor*:

$$\mathbf{Para}(\mathbf{Lens}(\mathbf{Set})) \xrightarrow{\mathbf{Para}((-)^*)} \mathbf{Para}(\mathbf{Lens}(\mathbf{Set}))$$



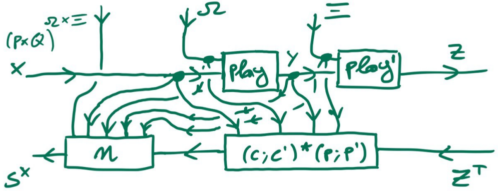
Nashator

The laxness of this functor represents the difference between two players competing



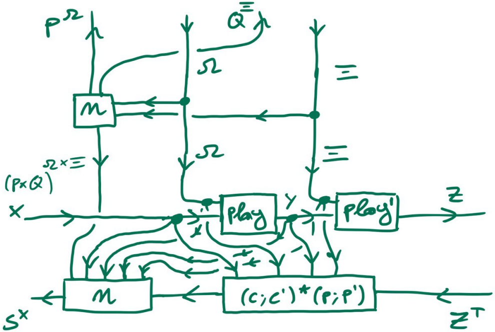
Nashator

The laxness of this functor represents the difference between two players competing and two players together:



Nashator

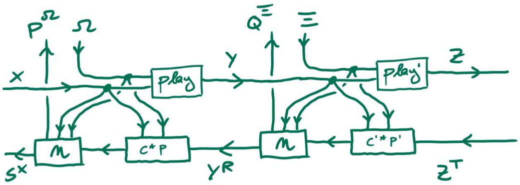
The laxness of this functor represents the difference between two players competing and two players together:



The Nashator breaks coalitions canonically

Nashator

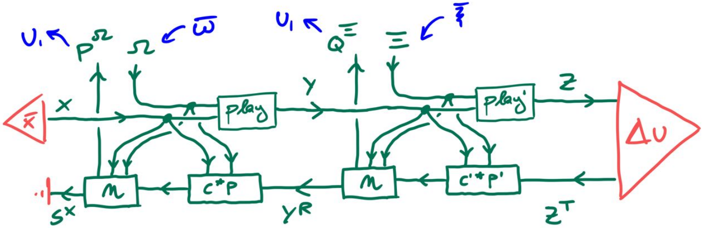
The laxness of this functor represents the difference between two players competing and two players together:



The Nashator breaks coalitions naturally

Nashator

Nashators + **lens composition** reproduce precisely the propagation of feedback from the regret function to players (aka performs **backward induction**)



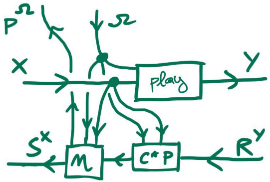
If players play the profile $(\bar{\omega}, \bar{\xi}) \in \Omega \times \Xi$, they respectively get back

$$u_1 = \lambda\omega\,.\ \underbrace{u(\mathsf{play}'(\bar{\xi}, \mathsf{play}(\omega, \bar{x}))) - u(\mathsf{play}'(\bar{\xi}, \mathsf{play}(\bar{\omega}, \bar{x})))}_{\text{regret expected by unilaterally deviating from } \bar{\omega} \text{ to } \omega}$$

$$u_2 = \lambda\xi\,.\ \underbrace{u(\mathsf{play}'(\xi, \mathsf{play}(\bar{\omega}, \bar{x}))) - u(\mathsf{play}'(\bar{\xi}, \mathsf{play}(\bar{\omega}, \bar{x})))}_{\text{regret expected by unilaterally deviating from } \bar{\xi} \text{ to } \xi}$$

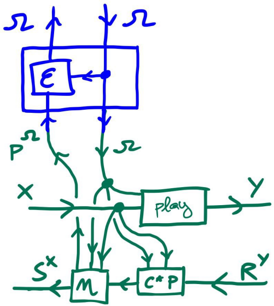
Players as a system

Now arenas provide enough information to allow players to be embodied by a system over it:



Players as a system

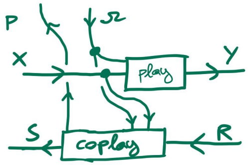
Now arenas provide enough information to allow players to be embodied by a system over it:



A morphism $\varepsilon : (\Omega \to P) \to \mathscr{P}\Omega$ is known as **selection function** and does indeed encode an agent's preference (see [3, 4, 5]). The extra dependency on Ω can be interpreted as a hint towards **incomplete information games** [6].

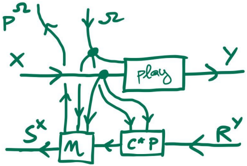
Recap

1. Start with a game dynamics



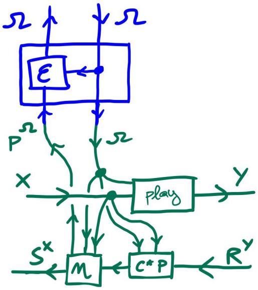
Recap

2. Get the arena by applying $\mathbf{Para}((-)^*)$



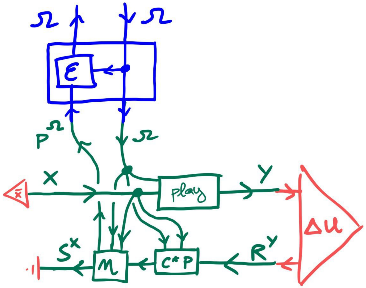
Recap

3. Add the players



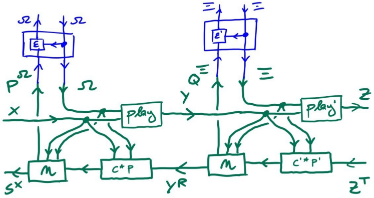
Recap

4. Surround with a context (Nature's move (initial state) + Δ(payoff function))



Recap

4. Or don't, and compose it with other games!



Conclusions

Conclusions

In this talk, we've seen...

Conclusions

In this talk, we've seen...

1. how to think accurately about the geometry of learners

Conclusions

In this talk, we've seen...

1. how to think accurately about the geometry of learners
2. the general motif behind their workings: states, changes, values and valuations

Conclusions

In this talk, we've seen...

1. how to think accurately about the geometry of learners
2. the general motif behind their workings: states, changes, values and valuations
3. the way such a situation gives rise to 'reverse-mode derivation' functors, and the importance of their monoidal structures

Conclusions

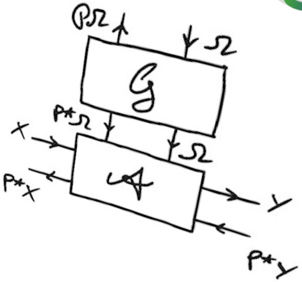
In this talk, we've seen...

1. how to think accurately about the geometry of learners
2. the general motif behind their workings: states, changes, values and valuations
3. the way such a situation gives rise to 'reverse-mode derivation' functors, and the importance of their monoidal structures
4. how to extend this to games, using the Nashator

NEW!

diegetic open games

the only ones with Nashators™



✓players ✓backprop ✓compositionality

Future directions

Wide horizon of directions from here: having pinned down the dynamical structure, we can…

Future directions

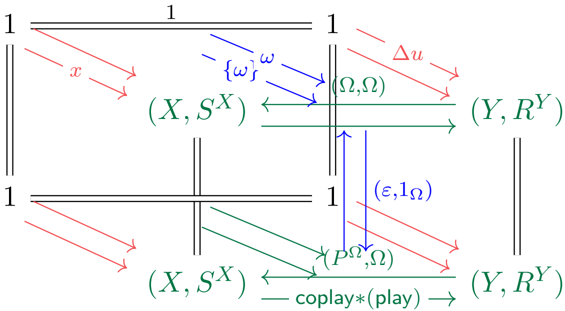
Wide horizon of directions from here: having pinned down the dynamical structure, we can...

1. vary the geometry and talk about stochastic, differential, evolutionary, etc. game theory

Future directions

Wide horizon of directions from here: having pinned down the dynamical structure, we can...

1. vary the geometry and talk about stochastic, differential, evolutionary, etc. game theory
2. separately extract equilibria and other behavioural information using machinery from (an extension of) categorical systems theory [7, 8]
   Fact: Nash equilibria are fixpoints of diegetic open games



Future directions

Wide horizon of directions from here: having pinned down the dynamical structure, we can...

1. vary the geometry and talk about stochastic, differential, evolutionary, etc. game theory
2. separately extract equilibria and other behavioural information using machinery from (an extension of) categorical systems theory [7, 8]
   Fact: Nash equilibria are fixpoints of diegetic open games
3. investigate the state-changes-values paradigm to analyze more systems, like Bayesian reasoners

Thank you!


davidad
@davidad
Replying to @Alan_Taylor_314 @Joe_DoesMath and @mattecapu
Behold, my Nashequilibriyinator!
I kept seeing people coOPerate with each other in little ways, just holding society together, y'know?
But soon, everyone in the Tri-State Area will be FORCED to play Nash equilibria, so they won't be able to coordinate without a tyrannical leader!

References I

K. Leyton-Brown and Y. Shoham, *Essentials of Game Theory: A Concise Multidisciplinary Introduction*. Morgan & Claypool, 2008.

M. Capucci, B. Gavranović, J. Hedges, and E. F. Rischel, "Towards foundations of categorical cybernetics", in *Proceedings of Applied Category Theory 2021*, EPTCS, 2021. [Online]. Available: https://arxiv.org/abs/2105.06332.

M. Capucci, N. Ghani, J. Ledent, and F. N. Forsberg, "Translating extensive form games to open games with agency", in *Proceedings of Applied Category Theory 2021*, EPTCS, 2021. [Online]. Available: https://arxiv.org/abs/2105.06763.

M. Escardó and P. Oliva, "Selection functions, bar recursion and backward induction", *Mathematical structures in computer science*, vol. 20, no. 2, pp. 127–168, 2010.

References II

- J. Hedges, P. Oliva, E. Shprits, V. Winschel, and P. Zahn, "Higher-order decision theory", in *International Conference on Algorithmic Decision Theory*, Springer, 2017, pp. 241–254.
- J. C. Harsanyi, "Games with incomplete information played by "bayesian" players, i–iii part i. the basic model", *Management science*, vol. 14, no. 3, pp. 159–182, 1967.
- D. J. Myers, "Double categories of open dynamical systems", 2020, [Online]. Available: https://arxiv.org/abs/2005.05956.
- ——, *Categorical Systems Theory*. 2022, (Work in progress). [Online]. Available: http://davidjaz.com/Papers/DynamicalBook.pdf (visited on 05/14/2022).