Interaction laws of monads and comonads

Tarmo Uustalu
joint work with Shin-ya Katsumata and Exequiel Rivas

OWLS seminar, 29 July 2020

Effects happen in interaction

- To run,

  an effectful program behaving as
  a computation

  needs to **interact** with

  a environment
  that an effect-providing machine behaves as

- E.g.,
  - a nondeterministic program needs a machine making choices;
  - a stateful program needs a machine coherently responding to fetch and store commands.

This talk

- We propose and study
  - functor-functor interaction laws,
  - monad-comonad interaction laws.

  as mathematical concepts for describing interaction protocols in this scenario.
- Functor-functor interaction laws are for unrestricted notions of computation
- Monad-comonad interaction laws are for notions of computation that are closed under
  - "doing nothing" (just returning),
  - sequential composition.

Outline

- Functor-functor and monad-comonad interaction laws
- Some examples and degeneracy theorems
- Dual—greatest interacting functor or monad;
  Sweedler dual—greatest interacting comonad
- Some examples
- Residual interaction laws (to counteract degeneracies, but not only)
- Object-object and monoid-comonoid interaction laws
  in duoidal categories

Functor-functor interaction laws

- Let $\mathcal{C}$ be a Cartesian category (symmetric monoidal will work too).
- Think $\mathcal{C} = \mathbf{Set}$.
- A *functor-functor interaction law* is given by two functors $F, G : \mathcal{C} \to \mathcal{C}$ and a family of maps

$$\phi_{X,Y} : FX \times GY \to X \times Y$$

natural in X, Y.

- Legend:
X – values, FX – computations
Y – states, GY – environments (incl an initial state)

Examples of functor-functor interaction laws

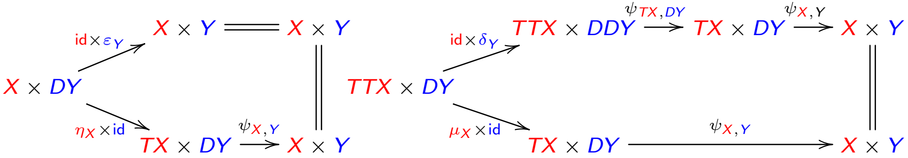
- $$F\,X = \underbrace{O \times}_{outp} ((\underbrace{I \Rightarrow}_{inp} X) \underbrace{\times}_{ext\ ch} (\underbrace{O' \times}_{outp} X)),$$
  $$G\,Y = \underbrace{O \Rightarrow}_{inp} ((\underbrace{I \times}_{outp} Y) \underbrace{+}_{int\ ch} (\underbrace{O' \Rightarrow}_{inp} Y))$$
  for some sets O, I, O'
- $\phi((o, (f, (o', x))), g) =$
  $$\text{case } g\,o \text{ of } \begin{cases} \mathsf{inl}\,(i, y) & \mapsto & (f\,i, y) \\ \mathsf{inr}\,h & \mapsto & (x, h\,o') \end{cases}$$
- We can vary ϕ, e.g., change o' to $o * o'$ in the 2nd case for some $* : O \times O' \to O'$
- We can also vary G, e.g., take
  $G'\,Y = \mathbb{N} \Rightarrow (I \times Y)$
- $\phi'(o, (f, \_)), g) = \text{let } (i, y) = g\,42 \text{ in } (f\,i, y)$
- (This is like session types, no?)

Monad-comonad interaction laws

- A *monad-comonad interaction law* is given by a monad (T, η, μ) and a comonad (D, ε, δ) and a family of maps

$$\psi_{X,Y} : TX \times DY \to X \times Y$$

natural in X, Y such that



- Legend:
  X – values, TX – computations
  Y – states, DY – environments (incl an initial state)

Some examples of mnd-cmnd int laws

- $TX = S \Rightarrow X$ (the reader monad),
  $DY = S_0 \times Y$
  for some S_0, S and $c : S_0 \to S$
- $\psi\,(f,(s_0,y)) = (f\,(c\,s_0),y)$
- Legend:
  X – values, S – “views” of store,
  Y – (control) states, S_0 – states of store

- $TX = S \Rightarrow (S \times X)$ (the state monad),
  $DY = S_0 \times (S_0 \Rightarrow Y)$
  for some S_0, S, $c : S_0 \to S$ and $d : S_0 \times S \to S_0$
  forming a *(very well-behaved) lens*
- $\psi\,(f,(s_0,g)) = \text{let}\ (s',x) = f\,(c\,s_0)\ \text{in}\ (x,g\,(d\,(s_0,s')))$

- $TX = \mu Z.\, X + Z \times Z$, $DY = \nu W.\, Y \times (W + W)$

Monad-comonad interaction laws are monoids

- A *functor-functor interaction law map* between (F, G, ϕ), (F', G', ϕ') is given by nat. transfs. $f : F \to F'$, $g : G' \to G$ such that

$$
\begin{array}{ccccc}
 & \overset{\mathrm{id} \times g_Y}{\nearrow} & FX \times GY & \xrightarrow{\phi_{X,Y}} & X \times Y \\
FX \times G'Y & & & & \Vert \\
 & \underset{f_X \times \mathrm{id}}{\searrow} & F'X \times G'Y & \xrightarrow{\phi'_{X,Y}} & X \times Y
\end{array}
$$

- Functor-functor interaction laws form a category with a composition-based monoidal structure.

- These categories are isomorphic:
  - monad-comonad interaction laws;
  - monoid objects of the category of functor-functor interaction laws.

Some degeneracy thms for func-func int laws

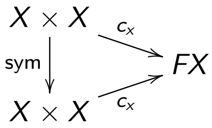
- Assume $\mathcal{C}$ is extensive ("has well-behaved coproducts").
- If F has a nullary operation, i.e., a family of maps

$$c_X : 1 \to FX$$

natural in X (eg, $F = \text{Maybe}$)
or a binary commutative operation, i.e., a family of maps

$$c_X : X \times X \to FX$$

natural in X such that



(eg, $F = \mathcal{M}^{+}_{\mathrm{fin}}$) and F interacts with G, then $GY \cong 0$.

A degeneracy thm for mnd-cmnd int laws

- If T has a binary associative operation, ie a family of maps $c_x : X \times X \to TX$ natural in X such that

$$\begin{array}{ccc} (X \times X) \times X & \xrightarrow{\ell_X} & TX \\ \downarrow{\scriptstyle \mathsf{ass}} & & \| \\ X \times (X \times X) & \xrightarrow{r_X} & TX \end{array}$$

where

$$\ell_X = (X \times X) \times X \xrightarrow{c_X \times \eta_X} TX \times TX \xrightarrow{c_{TX}} TTX \xrightarrow{\mu_X} TX$$

$$r_X = X \times (X \times X) \xrightarrow{\eta_X \times c_X} TX \times TX \xrightarrow{c_{TX}} TTX \xrightarrow{\mu_X} TX$$

(eg, $T = \mathsf{List}^+$), then any int law ψ of T and D obeys

$$\begin{array}{ccccc} (X \times X) \times X \times DY & \xrightarrow{\ell_X \times \mathsf{id}} & TX \times DY & \xrightarrow{\psi_{X,Y}} & X \times Y \\ \downarrow{\scriptstyle \mathsf{fst} \times \mathsf{id} \times \mathsf{id}} & & & & \| \\ X \times X \times DY & \xrightarrow{c_X \times \mathsf{id}} & TX \times DY & \xrightarrow{\psi_{X,Y}} & X \times Y \\ \uparrow{\scriptstyle \mathsf{id} \times \mathsf{snd} \times \mathsf{id}} & & & & \| \\ X \times (X \times X) \times DY & \xrightarrow{r_X \times \mathsf{id}} & TX \times DY & \xrightarrow{\psi_{X,Y}} & X \times Y \end{array}$$

Dual of a functor

- Assume now $\mathcal{C}$ is Cartesian closed.
- For a functor $G : \mathcal{C} \to \mathcal{C}$, its *dual* is the functor $G^\circ : \mathcal{C} \to \mathcal{C}$ is

$$G^\circ X = \int_Y GY \Rightarrow (X \times Y)$$

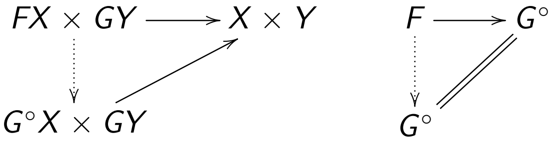
(if this end exists).

- $(-)^\circ$ is a functor $[\mathcal{C}, \mathcal{C}]^{\mathrm{op}} \to [\mathcal{C}, \mathcal{C}]$
(if all functors $\mathcal{C} \to \mathcal{C}$ are dualizable;
if not, restrict to some full subcategory of $[\mathcal{C}, \mathcal{C}]$ closed under dualization).

Dual of a functor ctd

- The dual G° is the "greatest" functor interacting with G.
- These categories are isomorphic:
  - functor-functor interaction laws;
  - pairs of functors F, G with nat. transfs. $F \to G^\circ$;
  - pairs of functors F, G with nat. transfs. $G \to F^\circ$.

$$\frac{FX \times GY \to X \times Y}{FX \to \underbrace{\int_Y GY \Rightarrow (X \times Y)}_{G^\circ X}}$$



Some examples of dual

- Let $GY = 1$. Then $G^\circ X \cong 0$.
- Let $GY = \Sigma a : A.G'aY$, then $G^\circ X \cong \Pi a : A.(G'a)^\circ X$.
- In particular,
  for $GY = 0$, we have $G^\circ X \cong 1$
  and, for $GY = G_0 Y + G_1 Y$, we have $G^\circ X \cong G_0^\circ X \times G_1^\circ X$.
- Let $GY = A \Rightarrow Y$. We have $G^\circ X \cong A \times X$.
- But: Let $GY = \Pi a : A.G'a\,Y$. We only have
  $\Sigma a : A.(G'a)^\circ X \to G^\circ X$.
- $\mathrm{Id}^\circ \cong \mathrm{Id}$.
- But we only have $G_0^\circ \cdot G_1^\circ \to (G_0 \cdot G_1)^\circ$.
- For any G with a nullary or a binary commutative operation, we have $G^\circ X \cong 0$.

Dual of a comonad / Sweedler dual a monad

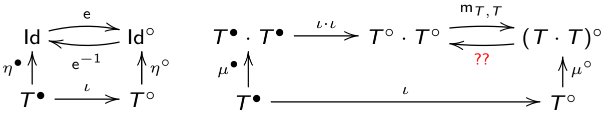
- The dual D° of a comonad D is a monad.
- This is because $(-)^\circ : [\mathcal{C}, \mathcal{C}]^{\mathrm{op}} \to [\mathcal{C}, \mathcal{C}]$ is lax monoidal, so send monoids to monoids.

- But $(-)^\circ$ is not oplax monoidal, does not send comonoids to comonoids.
- So the dual T° of a monad T is generally not a comonad.

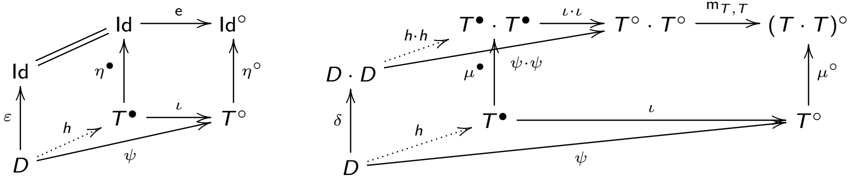
- However we can talk about the *Sweedler dual* $T^\bullet$ of T.
- Informally, it is defined as the greatest functor D that is smaller than the functor T° and carries a comonad structure $\eta^\bullet$, $\mu^\bullet$ agreeing with η°, μ°.

Dual of a comonad / Sweedler dual of a monad ctd

- Formally, the *Sweedler dual* of the monad T is the comonad $(T^\bullet, \eta^\bullet, \mu^\bullet)$ together with a natural transformation $\iota : T^\bullet \to T^\circ$ such that



  and such that, for any comonad (D, ε, δ) together with a natural transformation ψ satisfying the same conditions, there is a unique comonad map $h : D \to T^\bullet$ satisfying



Some examples of dual and Sweedler dual

- Let $TX = \mathrm{List}^{+}X \cong \Sigma n : \mathbb{N}. ([0..n] \Rightarrow X)$ (the nonempty list monad) .
- We have $T^{\circ}Y \cong \Pi n : \mathbb{N}. ([0..n] \times Y)$ but $T^{\bullet}Y \cong Y \times (Y + Y)$.
- Let $TX = S \Rightarrow (S \times X) \cong (S \Rightarrow S) \times (S \Rightarrow X)$ (the state monad).
- We have $T^{\circ}Y = (S \Rightarrow S) \Rightarrow (S \times Y)$ but $T^{\bullet}Y = S \times (S \Rightarrow Y)$.

Residual interaction laws

- Given a monad (R, η^R, μ^R) on $\mathcal{C}$.
- Eg, $R =$ Maybe, $\mathcal{M}^+$ or $\mathcal{M}$.

- A *residual functor-functor interaction law* is given by two functors $F, G : \mathcal{C} \to \mathcal{C}$ and a family of maps

$$\phi_{X,Y} : FX \times GY \to R(X \times Y)$$

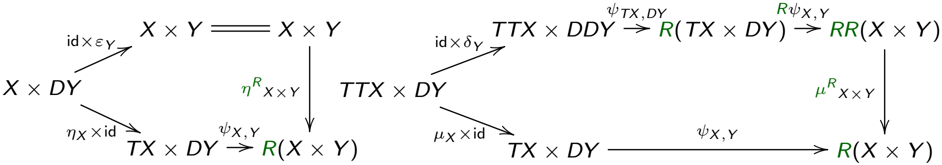
natural in X, Y.

Residual interaction laws ctd

- A *residual monad-comonad interaction law* is given by a monad (T, η, μ), a comonad (D, ε, δ) and a family of maps

$$\psi_{X,Y} : TX \times DY \to R(X \times Y)$$

natural in X, Y such that



- R-residual functor-functor interaction laws form a monoidal category with R-residual monad-comonad interaction laws as monoids.

Interaction laws and Chu spaces

- The *Day convolution* of F, G is

$$(F \star G)Z = \int^{X,Y} \mathcal{C}(X \times Y, Z) \bullet (FX \times GY)$$

(if this coend exists).

- These categories are isomorphic:
  - functor-functor interaction laws;
  - Chu spaces on $([\mathcal{C}, \mathcal{C}], \mathrm{Id}, \star)$ with vertex Id, ie, triples of two functors F, G with a nat transf $F \star G \to \mathrm{Id}$.

(if $\star$ is defined for all functors).

$$\dfrac{\dfrac{FX \times GY \to X \times Y}{\mathcal{C}(X \times Y, Z) \to \mathcal{C}(FX \times GY, Z)}}{\underbrace{\int^{X,Y} \mathcal{C}(X \times Y, Z) \bullet (FX \times GY)}_{(F \star G)Z} \to Z}$$

Interaction laws and Chu spaces ctd

- We do not immediately get another chacterization of the category of monad-comonad interaction laws.
- That's because the standard monoidal structure on the above category of Chu spaces is constructed from the Day convolution.
- But we want a monoidal structure from composition.

Interaction laws and Hasegawa's glueing

- Given a duoidal category $(\mathcal{F}, I, \cdot, J, \star)$ closed wrt. $(J, \star)$.
- Given also a monoid (R, η^R, μ^R) in $(\mathcal{F}, I, \cdot)$.
- Define $(-)^\circ : \mathcal{F}^{\mathrm{op}} \to \mathcal{F}$ by $G^\circ = G \multimap_\star R$.
- $(-)^\circ$ is lax monoidal.
- By an argument by Hasegawa, the comma category $\mathcal{F} \downarrow (-)^\circ$ has a $(I, \cdot)$ based monoidal structure.
- Now take $\mathcal{F} = [\mathcal{C}, \mathcal{C}]$ with $(I, \cdot)$ its composition monoidal and $(J, \star, \multimap_\star)$ its Day convolution SMC structure (if $\star$ and $\multimap_\star$ are defined for all functors).
- Then these categories are isomorphic:
  - R-residual monad-comonad interaction laws;
  - monoids in the monoidal category $[\mathcal{C}, \mathcal{C}] \downarrow (-)^\circ$.

Relation to effect handling (jww Niels Voorneveld)

- An R-residual mnd-cmnd int law of T, D explains how some of the effects of a computation are dealt with by the environment, some are left alone or transformed.
- Given
  - an int law $\psi_{Y,Z} : T(Y \Rightarrow Z) \to DY \Rightarrow RZ$,
  - a coalgebra $(B, \beta : B \to DB)$ of D (a coeffect producer) and
  - an algebra $(C, \gamma : RC \to C)$ of R (a residual effect handler)

  we get an algebra $(B \Rightarrow C, (\beta \Rightarrow \gamma) \circ \psi_{B,C} : T(B \Rightarrow C) \to B \Rightarrow C)$ of T (an effect handler).
- In fact, mnd-mnd interaction laws are in a bijection with carrier-exponentiating functors from $(\mathbf{Coalg}(D))^{\mathrm{op}} \times \mathbf{Alg}(R) \to \mathbf{Alg}(T)$.

Takeaway

- A single framework for talking about computations, environments and interaction
- Lots of mathematical structure around, a lot can be stated very generally

- What are some recipes for calculating the Sweedler dual?
- Sweedler dual in the residual case
- Relationship of interaction laws to session types