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 $\mathcal{C} = \textbf{Set}$.
- A \textit{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

- \( F \mathcal{X} = O \times ((I \Rightarrow X) \times (O' \times X)), \)
  \( \begin{array}{l}
  \text{outp} \\
  \text{inp} \\
  \text{ext ch} \\
  \text{outp}
  \end{array} \)

- \( G \mathcal{Y} = O \Rightarrow ((I \times Y) + (O' \Rightarrow Y)), \)
  \( \begin{array}{l}
  \text{inp} \\
  \text{outp} \\
  \text{int ch} \\
  \text{inp}
  \end{array} \)

for some sets \( O, I, O' \)

- \( \phi((o, (f, (o', x))), g) = \)
  \begin{cases}
    \text{inl} (i, y) & \mapsto (f \ i, y) \\
    \text{inr} \ h & \mapsto (x, h \ o')
  \end{cases}

We can vary \( \phi \), e.g., change \( o' \) to \( o \ast o' \) in the 2nd case
for some \( \ast : O \times O' \rightarrow O' \)

We can also vary \( G \), e.g., take
\( G' \mathcal{Y} = \mathbb{N} \Rightarrow (I \times Y) \)

- \( \phi'((o, (f, _)), g) = \) let \((i, y) = g \text{ 42}\) in \((f \ i, y)\)

(This is like session types, no?)
Monad-comonad interaction laws

A *monad-comonad interaction law* is given by a monad \((T, \eta, \mu)\) and a comonad \((D, \varepsilon, \delta)\) and a family of maps

\[ \psi_{X,Y} : TX \times DY \rightarrow X \times Y \]

natural in \(X, Y\) such that

![Diagram of monad-comonad interaction laws]

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 \( c : S_0 \rightarrow S \)
- \( \psi (f, (s_0, y)) = (f (c s_0), y) \)
- **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 \( S_0, S, c : S_0 \rightarrow S \) and \( d : S_0 \times S \rightarrow S_0 \)
  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 \((F, G, \phi)\), \((F', G', \phi')\) is given by nat. transf. \(f : F \to F'\), \(g : G' \to G\) such that

\[
\begin{align*}
F X \times G' Y & \xrightarrow{id \times g Y} FX \times G Y \\
& \xrightarrow{\phi X, Y} X \times Y \\
& \xrightarrow{f X \times id} F' X \times G' Y \\
& \xrightarrow{\phi' X, Y} X \times Y
\end{align*}
\]

- 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

\[
\begin{array}{ccc}
X \times X & \xrightarrow{c_x} & FX \\
\downarrow\text{sym} & & \downarrow c_x \\
X \times X & \xleftarrow{c_x} & FX
\end{array}
\]

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

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

$$
\begin{array}{c}
(X \times X) \times X \\
\downarrow \text{ass} \\
X \times (X \times X)
\end{array} \xrightarrow{\ell_X} TX \\
\begin{array}{c}
X \times (X \times X)
\end{array} \xrightarrow{r_X} TX
$$

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, $T = \text{List}^+$), then any int law $\psi$ of $T$ and $D$ obeys

$$
\begin{array}{c}
(X \times X) \times X \times DY \\
\downarrow \text{fst} \times \text{id} \times \text{id} \\
X \times X \times DY \\
\downarrow \text{id} \times \text{snd} \times \text{id} \\
X \times (X \times X) \times DY
\end{array} \xrightarrow{\ell_X \times \text{id}} TX \times DY \xrightarrow{\psi_{X,Y}} TX \times DY \xrightarrow{\psi_{X,Y}} TX \times DY \\
\begin{array}{c}
X \times X \times DY \\
\downarrow \text{c_X} \times \text{id} \\
TX \times DY \\
\downarrow \psi_{X,Y} \\
TX \times DY \\
\downarrow \psi_{X,Y} \\
TX \times DY
\end{array} \xrightarrow{\text{id} \times \text{snd} \times \text{id}} TX \times DY \\
\begin{array}{c}
X \times (X \times X) \times DY \\
\downarrow \text{r_X} \times \text{id} \\
TX \times DY
\end{array}
$$
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 $[\mathcal{C}, \mathcal{C}]^{\text{op}} \to [\mathcal{C}, \mathcal{C}]$
  (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 $G^\circ$ 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. $G \to F^\circ$.

\[
\begin{align*}
  FX \times GY & \to X \times Y \\
  FX & \to \int_Y GY \Rightarrow (X \times Y)
\end{align*}
\]

\[
\begin{array}{c}
  FX \times GY \to X \times Y \\
  G^\circ X \times GY \\
  G^\circ X \to G^\circ
\end{array}
\]

\[
\begin{array}{c}
  FX \times GY \\
  F \to G^\circ
\end{array}
\]
Some examples of dual

- Let $GY = 1$. Then $G^\circ X \cong 0$.

- Let $GY = \Sigma a : A.G'aiY$, then $G^\circ X \cong \Pi a : A.(G'ai)^\circ X$.

- 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 $GY = \Pi a : A.G'ai Y$. We only have $\Sigma a : A.(G'ai)^\circ X \rightarrow G^\circ X$.

- $\text{Id}^\circ \cong \text{Id}$.

- But we only have $G_0^\circ \cdot G_1^\circ \rightarrow (G_0 \cdot G_1)^\circ$.

- For any $G$ with a nullary or a binary commutative operation, we have $G^\circ X \cong 0$. 
The dual $D^\circ$ of a comonad $D$ is a monad.

This is because $(−)^\circ : [C,C]^{op} \to [C,C]$ is lax monoidal, so send monoids to monoids.

But $(−)^\circ$ is not oplax monoidal, does not send comonoids to comonoids.

So the dual $T^\circ$ 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 $T^\circ$ and carries a comonad structure $\eta^\bullet$, $\mu^\bullet$ agreeing with $\eta^\circ$, $\mu^\circ$. 
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

\begin{align*}
\text{Id} \quad \xrightarrow{e} \quad \text{Id}^\circ \\
\eta^\bullet \quad \xleftarrow{e^{-1}} \quad \eta^\circ \\
T^\bullet \quad \xrightarrow{\iota} \quad T^\circ
\end{align*}

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

\begin{align*}
\text{Id} \quad \xrightarrow{e} \quad \text{Id}^\circ \\
\eta^\bullet \quad \xleftarrow{e^{-1}} \quad \eta^\circ \\
D \quad \xrightarrow{\varepsilon} \quad T^\bullet \quad \xrightarrow{\iota} \quad T^\circ
\end{align*}
Some examples of dual and Sweedler dual

- Let \( TX = \text{List}^+ X \cong \Sigma n : \mathbb{N}. ([0..n] \Rightarrow X) \)
  (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 $(R, \eta^R, \mu^R)$ on $\mathcal{C}$.
- Eg, $R = \text{Maybe}$, $\mathcal{M}^+$ 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 $X$, $Y$. 
A residual monad-comonad interaction law is given by a monad \((T, \eta, \mu)\), a comonad \((D, \varepsilon, \delta)\) and a family of maps

\[\psi_{X,Y} : TX \times DY \rightarrow R(X \times Y)\]

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 $([\mathcal{C}, \mathcal{C}], \text{Id}, \star)$ with vertex $\text{Id}$, ie, triples of two functors $F$, $G$ with a nat transf $F \star G \to \text{Id}$.

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

\[
\begin{align*}
FX \times GY &\to X \times Y \\
\mathcal{C}(X \times Y, Z) &\to \mathcal{C}(FX \times GY, Z) \\
\int^{X,Y} \mathcal{C}(X \times Y, Z) \bullet (FX \times GY) &\to Z
\end{align*}
\]
We do not immediately get another characterization 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 \((R, \eta^R, \mu^R)\) in \((\mathcal{F}, I, \cdot)\).
- Define \((-)^\circ : \mathcal{F}^{\text{op}} \to \mathcal{F}\) by \(G^\circ = G \to \star R\).
- \((-)^\circ\) 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 \((J, \star, \to \star)\) its Day convolution SMC structure (if \(\star\) and \(\to \star\) are defined for all functors).

Then these categories are isomorphic:
- \(R\)-residual monad-comonad interaction laws;
- monoids in the monoidal category \([\mathcal{C}, \mathcal{C}] \downarrow (-)^\circ\).
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 $(\text{Coalg}(D))^\text{op} \times \text{Alg}(R) \to \text{Alg}(T)$. 
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