Interaction laws of monads and comonads

Shin-ya Katsumata   Exequiel Rivas   Tarmo Uustalu

LICS 2020
do a ← read (
  if a then
    return 42
  else
    write C
    return 41
)

Requests:
\[ F_Z = (B \Rightarrow Z) + (\{A, B, C\} \times Z) \]

Responses:
\[ G_W = (B \times W) \times (\{A, B, C\} \Rightarrow W) \]

▶ Computations are wellfounded trees leaf-labelled by return values:
\[ F^* X = \mu Z.X + F_Z \]

▶ Environments are nonwellfounded trees node-labelled by states:
\[ G^* Y = \nu W.Y \times G_W \]

Monad-comonad interaction as a nat. transf.
\[ \psi_{X,Y}: F^* X \times G^* Y \rightarrow X \times Y \]

comes from functor-functor interaction as a nat. transf.
\[ \phi_{Z,W}: F_Z \times G_W \rightarrow Z \times W \]
Motivation

do a ← read ()
if a then
  return 42
else
  write C
  return 41
do a ← \texttt{read} ()
  if a then
    return 42
  else
    write C
    return 41
Motivation

do a ← read ()
if a then
  return 42
else
  write C
  return 41
read ()
Motivation

\[
\text{do } a \leftarrow \text{read} () \\
\text{if } a \text{ then} \\
\quad \text{return } 42 \\
\text{else} \\
\quad \text{write } C \\
\quad \text{return } 41
\]

Execution stops with result \((41, y_2)\)

Requests:
\[
\text{FZ} = (B \Rightarrow Z) + \{A, B, C\} \times Z
\]

Responses:
\[
\text{GW} = (B \times W) \times (\{A, B, C\} \Rightarrow W)
\]

\[\text{Computations are wellfounded trees leaf-labelled by return values:}\]
\[
\mathcal{F}^* X = \mu Z. X + \mathcal{F}Z
\]

\[\text{Environments are nonwellfounded trees node-labelled by states:}\]
\[
\mathcal{G}^\dagger Y = \nu W. Y \times \mathcal{G}W
\]

\[\text{Monad-comonad interaction as a nat. transf.}\]
\[
\psi_{X,Y}: \mathcal{F}^* X \times \mathcal{G}^\dagger Y \rightarrow X \times Y
\]

\[\text{comes from functor-functor interaction as a nat. transf.}\]
\[
\phi_{Z,W}: \mathcal{F}Z \times \mathcal{G}W \rightarrow Z \times W
\]
do a ← read ()
  if a then
    return 42
  else
    write C
    return 41

read ()
false

Execution stops with result (41, y₁)

Requests:
FZ = (B ⇒ Z) + ({A, B, C} × Z)

Responses:
GW = (B × W) × ({A, B, C} ⇒ W)

▶ Computations are wellfounded trees leaf-labelled by return values:
F∗X = µZ.X + FZ

▶ Environments are nonwellfounded trees node-labelled by states:
G†Y = νW.Y × GW

Monad-comonad interaction as a nat. transf.
ψX,Y: F∗X × G†Y → X × Y

comes from functor-functor interaction as a nat. transf.
φZ,W: FZ × GW → Z × W
Motivation

do a ← read ()
if a then
  return 42
else
  write C
  return 41

Execution stops with result (41, y2)

Requests:
FZ = (B ⇒ Z) + ({A, B, C} × Z)

Responses:
GW = (B × W) × ({A, B, C} ⇒ W)

▶ Computations are wellfounded trees leaf-labelled by return values:
F∗X = µZ.X + FZ

▶ Environments are nonwellfounded trees node-labelled by states:
G†Y = νW.Y × GW

Monad-comonad interaction as a nat. transf.
ψX,Y: F∗X × G†Y → X × Y
comes from functor-functor interaction as a nat. transf.
φZ,W: FZ × GW → Z × W
Motivation

do a ← read ()
if a then
  return 42
else
  write C
  return 41

Execution stops with result (41, y1)

Requests:
FZ = (B ⇒ Z) + ({A, B, C}× Z)

Responses:
GW = (B× W)× ({A, B, C}⇒ W)

▶ Computations are wellfounded trees leaf-labelled by return values:
F ∗ X = µZ.X + FZ

▶ Environments are nonwellfounded trees node-labelled by states:
G † Y = νW.Y × GW

Monad-comonad interaction as a nat. transf.
ψ X, Y : F ∗ X × G † Y → X × Y
comes from functor-functor interaction as a nat. transf.
φ Z, W : FZ × GW → Z × W
do a ← read ()
  if a then
    return 42
  else
    write C
    return 41

Execution stops with result (41, y2)

Requests:
\[ F_{Z} = (B \Rightarrow Z) + (\{A, B, C\} \times Z) \]

Responses:
\[ G_{W} = (B \times W) \times (\{A, B, C\} \Rightarrow W) \]

\[ \psi_{X,Y}: F_{X} \times G_{Y} \rightarrow X \times Y \]
\[ \phi_{Z,W}: F_{Z} \times G_{W} \rightarrow Z \times W \]
Motivation

```plaintext
do a ← read ()
  if a then
    return 42
  else
    write C
    return 41
```

Requests:
\[ F_Z = (B \Rightarrow Z) + (\{A, B, C\} \times Z) \]

Responses:
\[ G_W = (B \times W) \times (\{A, B, C\} \Rightarrow W) \]

Computations are wellfounded trees leaf-labelled by return values:
\[ F^*_X = \mu Z.X + F_Z \]

Environments are nonwellfounded trees node-labelled by states:
\[ G^*_Y = \nu W.Y \times G_W \]

Monad-comonad interaction as a nat. transf.
\[ \psi_{X,Y}: F^*_X \times G^*_Y \rightarrow X \times Y \]

comes from functor-functor interaction as a nat. transf.
\[ \phi_{Z,W}: F_Z \times G_W \rightarrow Z \times W \]
do a ← read ()
  if a then
    return 42
  else
    write C
    return 41
Motivation

{\begin{tabular}{l}
\textbf{do} \ a \leftarrow \text{read}() \\
\quad \textbf{if} \ a \ \text{then} \\
\quad \quad \text{return} \ 42 \\
\quad \textbf{else} \\
\quad \quad \textbf{write} \ C \\
\quad \quad \text{return} \ 41 \\
\end{tabular}}

\text{Execution stops with result } (41, y_2)
do a ← read ()
  if a then
    return 42
  else
    write C
    return 41
write C

Execution stops with result (41, y_2)
do a ← read ()
if a then
  return 42
else
  write C
  return 41
write C

Execution stops with result (41, y_2)

Requests:
FZ = (B ⇒ Z) + (A, B, C) × Z

Responses:
GW = (B × W) × (A, B, C) ⇒ W

Computations are wellfounded trees leaf-labelled by return values:
F∗X = µZ.X + FZ

Environments are nonwellfounded trees node-labelled by states:
G†Y = νW.Y × GW

Monad-comonad interaction as a nat. transf.
ψX,Y: F∗X × G†Y → X × Y

comes from functor-functor interaction as a nat. transf.
φZ,W: FZ × GW → Z × W
Motivation

```
do a ← read ()
   if a then
      return 42
   else
      write C
      return 41
```
Motivation

```plaintext
do a ← read ()
  if a then
    return 42
  else
    write C
    return 41
```

Execution stops with result $(41, y_2)$

Requests: $FZ = (B \Rightarrow Z) + \{A, B, C\} \times Z$

Responses: $GW = (B \times W) \times (\{A, B, C\} \Rightarrow W)$

▶ Computations are wellfounded trees leaf-labelled by return values: $F^\ast X = \mu Z. X + FZ$

▶ Environments are nonwellfounded trees node-labelled by states: $G\dagger Y = \nu W. Y \times GW$

Monad-comonad interaction as a nat.

```
\psi_{X,Y}: F^\ast X \times G\dagger Y \to X \times Y
```

comes from functor-functor interaction as a nat.

```
\phi_{Z,W}: FZ \times GW \to Z \times W
```
Motivation

\[
\text{do } a \leftarrow \text{read ()} \\
\text{if } a \text{ then} \\
\quad \text{return } 42 \\
\text{else} \\
\quad \text{write } C \\
\quad \text{return } 41
\]

* Execution stops with result \((41, y_2)\)
Motivation

```plaintext
do a ← read ()
    if a then
        return 42
    else
        write C
        return 41
```

Execution stops with result \((41, y_2)\)

Requests: \(FZ = (\mathbb{B} \Rightarrow Z) + (\{A, B, C\} \times Z)\)

Responses: \(GW = (\mathbb{B} \times W) \times (\{A, B, C\} \Rightarrow W)\)

- Computations are wellfounded trees leaf-labelled by return values: \(F^* X = \mu Z. X + FZ\)
- Environments are nonwellfounded trees node-labelled by states: \(G^\dagger Y = \nu W. Y \times GW\)
Motivation

Execution stops with result \((41, y_2)\)

Requests: \(FZ = (B \Rightarrow Z) + (\{A, B, C\} \times Z)\)

Responses: \(GW = (B \times W) \times (\{A, B, C\} \Rightarrow W)\)

- Computations are wellfounded trees leaf-labelled by return values: \(F^* X = \mu Z. X + FZ\)
- Environments are nonwellfounded trees node-labelled by states: \(G^\uparrow Y = \nu W. Y \times GW\)
Motivation

Execution stops with result \((41, y_2)\)

Requests: \(FZ = (B \Rightarrow Z) + (\{A, B, C\} \times Z)\)

Responses: \(GW = (B \times W) \times (\{A, B, C\} \Rightarrow W)\)

- Computations are wellfounded trees leaf-labelled by return values: \(F^* X = \mu Z. X + FZ\)
- Environments are nonwellfounded trees node-labelled by states: \(G^\dagger Y = \nu W. Y \times GW\)

Monad-comonad interaction as a nat. transf. \(\psi_{X,Y} : F^* X \times G^\dagger Y \rightarrow X \times Y\)
comes from functor-functor interaction as a nat. transf. \(\phi_{Z,W} : FZ \times GW \rightarrow Z \times W\)
Overview of the talk

Motivation.
Overview of the talk

- Motivation.
- Functor-functor interaction laws.

practically self-motivated is provided by residual functor-functor interaction laws: monad-comonad interaction laws and residual remains on generalization when the machine need not be able to perform all effects of the computation, or used only partially and remain after interaction or running (see the end of Section 2).

Functor-functor interaction laws. We begin with functor-functor interaction, to then proceed with one fixed base category $C$ given by two endofunctors $F, G$ and a natural transformation $\eta : G \to F$. Let $X \xrightarrow{f} Y$ be some fixed object. But we can also take, e.g., $X = A \times X$, $Y = C \times Y$, and $G(Y) = Y \times Y$, $F(X) = X \times X$, $\eta : Y \to X$ some fixed map $A \to X$.

### Example 2.1.
The archetypical example of a functor-functor interaction law is defined by $F \cong X \to X$, $G \cong A \times X$, and $\eta : A \times X \to X$. This example is basically available for any $X$ and $A$. But we can also take, e.g., $X = A \times X$, $Y = C \times Y$, and $G(Y) = Y \times Y$, $F(X) = X \times X$, $\eta : Y \to X$ some fixed map $A \to X$.

### Example 2.2.
This example may be obtained by taking $F \cong X \to X$ and $G \cong Y \to Y$. Then $\eta(j) \circ f = f_\circ j_0 \circ \eta(\cdot)$.

### Example 2.3.

A more interesting example is obtained by taking $F \cong A \times (X \to X)$, $G \cong A \times (X \to X)$, $\eta(j) \circ f = f_\circ j_0 \circ \eta(\cdot)$.

### Example 2.4.

In a nutshell, extensive categories are categories with finite products. For some constructions (the dual of a functor), we also need that $\mathcal{E}$ is extensive with finite products. For some constructions (the dual of a functor), we also need that $\mathcal{E}$ is extensive with finite products. For some constructions (the dual of a functor), we also need that $\mathcal{E}$ is extensive with finite products.

### Example 2.5.

A more interesting example is obtained by taking $F \cong A \times (X \to X)$, $G \cong A \times (X \to X)$, $\eta(j) \circ f = f_\circ j_0 \circ \eta(\cdot)$.

### Example 2.6.

The archetypical example of a functor-functor interaction law is defined by $F \cong X \to X$, $G \cong A \times X$, and $\eta : A \times X \to X$. This example is basically available for any $X$ and $A$. But we can also take, e.g., $X = A \times X$, $Y = C \times Y$, and $G(Y) = Y \times Y$, $F(X) = X \times X$, $\eta : Y \to X$ some fixed map $A \to X$.

### Example 2.7.

A more interesting example is obtained by taking $F \cong A \times (X \to X)$, $G \cong A \times (X \to X)$, $\eta(j) \circ f = f_\circ j_0 \circ \eta(\cdot)$.

### Example 2.8.

In a nutshell, extensive categories are categories with finite products. For some constructions (the dual of a functor), we also need that $\mathcal{E}$ is extensive with finite products. For some constructions (the dual of a functor), we also need that $\mathcal{E}$ is extensive with finite products. For some constructions (the dual of a functor), we also need that $\mathcal{E}$ is extensive with finite products.
Overview of the talk

- Motivation.
- Functor-functor interaction laws.
- Monad-comonad interaction laws.

Sometimes only our ‘lenses bound’, at the back of a functor constructed from some given functors can be expressed in terms of their duals. This holds for the composition of two general functors, i.e., for compositions of a general functor

Dual of exponents of a general functor: Let \( GT = A \otimes GT' \). For a general \( G \), we only have a canonical natural transformation with components \( A \otimes GT') \to GT'.

Dual of compositions of two general functors: For general \( G, G' \), we only have the canonical natural transformation \( (G', G) \bullet (G', G') \to (G', G') \). This implies that \((G', G) \bullet (G', G') = (G', G) \) is not monoidal, but only has the structure of a (monoidal) category.

Example 2.2. Let \( G, G', Y \), \( A \times X \otimes Y \), \( A \times (X \otimes Y) \). The dual of \( (G', G) \) is \( G' \bullet (G', G') \) which is isomorphic to \( (G', G) \). The canonical natural transformation \( \rho_{G', G} : (G', G) \to (G', G') \) is isomorphic to \( \rho_{G, G} \).

3 Monad-comonad interaction

3.1 Monad-comonad interaction laws

In a monad-comonad interaction law, the allowed computations (the chosen notions of computation) are finite, i.e., they are defined by a monad rather than a functor. In such a case, the allowed machine behaviors (the notion of machine behavior) are defined by a monad. The idea is that interaction of a “just returning” computation should remain immutable in the initial state of the given machine behavior whereas interaction of a sequence of computations should amount to a sequence of interactions.

We define a monad-comonad interaction law on \( C \) to be given by a monad \( T \) and a comonad \( D \) (\( C \)-indexed) together with a family \( \mu \) of maps

\[ \mu_{A, X} : T(A \times X) \to \mu A \times \mu X \]

natural in \( X \) and \( A \) (i.e., a functor-functor interaction law of \( T, D \) defined on the monad-
comonad structure) such that also

\[ \eta_A : A \to \epsilon A \]

and

\[ \sigma_B : \mu B \times \mu B = \mu B \times B \]

natural in \( B \) (i.e., a functor-functor interaction law of \( T, D \) defined on the monad-
comonad structure).

Example 3.1. Take \( TX = A \Rightarrow A \times X \), \( DY = A \Rightarrow Y \) and \( \psi(f, (a, b)) = (\psi_a(a), f(b)) \) for a fixed monad \( D \). The functors

\[ D = \text{a monad is called monad map if a comonad and \( \psi \) meets the condition (i)} \]

Example 3.2. Take \( TX = A \Rightarrow X \times (A \otimes X) \), \( DY = A \Rightarrow Y \), \( \eta_A(a) : \epsilon A \Rightarrow A \) for a fixed monad \( D \) acting on a fixed object \( A \). The functor \( D \) is called monad map if a comonad and \( \psi \) meets the condition (i).

Example 3.3. Take \( TX = A \Rightarrow (A \otimes X) \times (A \otimes Y) \), \( DY = A \Rightarrow Y \), \( \eta_A(a) : \epsilon A \Rightarrow A \) for a fixed monad \( D \) acting on a fixed object \( A \). The functors \( T = \text{a monad is called monad map if a comonad and \( \psi \) meets the condition (i)} \]

Example 3.4. Take \( TX = A \Rightarrow (A \otimes X) \times (A \otimes Y) \), \( DY = A \Rightarrow Y \), \( \eta_A(a) : \epsilon A \Rightarrow A \) for a fixed monad \( D \) acting on a fixed object \( A \). The functors \( T = \text{a monad is called monad map if a comonad and \( \psi \) meets the condition (i)} \]

Monad-comonad interaction laws are essentially the same as monad objects in the monoidal category \( B(\mathcal{C}) \) of functor-functor interaction laws. To be precise, monad-comonad interaction law \( (\psi, \mu, \epsilon) \) yields a monad \((\psi T, \mu T, \epsilon T)\) in \( B(\mathcal{C}) \) and vice versa.

Monad-comonad interaction laws \( (\psi, \mu, \epsilon) \) form a category \( B(\mathcal{C}) \) monomorphic to the category \( \text{Mon} \text{B}(\mathcal{C}) \).

3.2 A degeneracy result

Monads with an associative operation. Here is a degeneracy theorem for monad-comonad interaction laws: If a normal \( T \) has an associative binary operation, i.e., family of maps \( \cdot : A \times A \Rightarrow A \times A \) satisfying

\[ [x,y] \cdot [z,w] = [x,z] \cdot [y,w] \]

where

\[ \cdot_A = (A \otimes A) \times (A \otimes A) \Rightarrow A \otimes A \]

\[ \cdot_A = (A \otimes A) \times (A \otimes A) \Rightarrow A \otimes A \]

then \( \mu_A : (\cdot_A) \Rightarrow (\cdot_A) \) is a good.
Overview of the talk

- Motivation.
- Functor-functor interaction laws.
- Monad-comonad interaction laws.
- Sweedler dual.

Functor-functor interaction laws.

Overview of the talk

▶

Functor-functor interaction laws.

Now, if $\mathcal{T}$ has the Sweedler dual, there is a bijection between monad-comonad interaction laws of $\mathcal{T}$, $D$, i.e., natural

Moreover, if $\mathcal{T}$ has the Sweedler dual, there is a bijection between monad-comonad interaction laws of $\mathcal{T}$, $D$, i.e., natural

...
Overview of the talk

- Motivation.
- Functor-functor interaction laws.
- Monad-comonad interaction laws.
- Sweedler dual.
- Monoid-comonoid interaction laws.

Section 4, a runner of the monad $T$ in also essentially the same thing as a coalgebra $(f, y)$ of the functor $T^*$ satisfying the conditions:

\[
\begin{align*}
\Delta(X) & \rightarrow T \Delta(X) \\
\eta & \rightarrow T \eta
\end{align*}
\]

This is because of the bijection

\[
\int C = \int C(T(X) \times X) \xrightarrow{\cong} C(TX) \times X \xrightarrow{\varepsilon \times 1} C(TX)
\]

Recall that the functor $T^*$ in general not a comonad in $\mathbf{C}$, is not invertible, so we cannot generally speak of functor coalgebras satisfying conditions (i) as comonad coalgebras. Lastly, recall that the comonad $\text{Cost}_Y$ for an object $Y$ is defined by $\text{Cost}^Y = (\text{Cost}_Y Y, \phi, \psi)$, where $\phi$ is a natural transformation $\text{Cost}^Y \rightarrow \text{Cost}_Y Y$.

This gives us a third characterization; a monad essentially the same as an object $Y$ together with a natural transformation $\phi$ between the underlying functor of the comonad $\text{Cost}_Y$ and the functor $T^*$, satisfying:

\[
\begin{align*}
\int C & = \int C(TX) \\
\phi & = \int \phi
\end{align*}
\]

This is because of the bijection

\[
\begin{align*}
\varepsilon & \equiv \int \phi = \varepsilon \\
\int C & \xrightarrow{\phi} \int C(TX)
\end{align*}
\]

(The first bijection is an internal version of the Yoneda lemma, which applies $\phi^* \equiv \phi$ in necessity strong if the Sweedler dual $\text{Cost}^Y \times T^*$ of the monad $T$ exists, then we can continue this reasoning. We see that a monad is essentially the same as an object $Y$ with a comonad morphism between $\text{Cost}_Y Y$ and $\text{Cost}^Y$.

Summing up, we have established that the following categories are isomorphic:

- (i) runners of $T^*$.
- (ii) objects $Z$ with a monad map from $T$ to $T^*$.
- (iii) functor coalgebras of $T^*$ subject to conditions (5).
- (iv) objects $Y$ with a natural transformation from $\text{Cost}_Y Y$ to $\text{Cost}^Y$.
- (v) comonad coalgebras of $\text{Cost}_Y$.
- (vi) monad-comonad interaction laws.

4.2 Runners vs. monad-comonad interaction laws

Monad-comonad interaction laws of $T$, $D$ are in a bijection with $D$-coalgebra $T$-runner specs by which we mean concrete-factoring fractions between Codeg$(D)$ and Run$(T)$, i.e., functions $\Psi : \text{Codeg}(D) \rightarrow \text{Run}(T)$ such that

\[
\text{Codeg}(D) \xrightarrow{\Psi} \text{Run}(T)
\]

Indeed, given a monad-comonad interaction law $\phi$, we can define a runner $\Psi$ by

\[
\Psi_{\phi} = TX \xrightarrow{\phi} TY \xrightarrow{\Phi} TX \xrightarrow{\phi} TY \xrightarrow{\Phi} TX
\]

In the opposite direction, given a runner specs $\Psi$, we build an interaction law from the cofree coalgebra of $D$. For any $D$, we define a monad-comonad interaction law $\phi$ by $\Psi_{\phi} = TX \xrightarrow{\Phi} TY \xrightarrow{\phi} TX \xrightarrow{\Phi} TY \xrightarrow{\phi} TX$.

A pair of a monad map $f : T \rightarrow T'$ and a coalgebra map $\gamma : TX \rightarrow TX$ is an interaction map between $(T, \eta)$ and $(T', \zeta)$ if the corresponding cofree-cofree runners satisfy

\[
\text{Codeg}(D) \xrightarrow{\psi} \text{Run}(T'), \text{Codeg}(D) \xrightarrow{\lambda} \text{Run}(T)
\]

(Notice that $\text{Codeg}(D)$, $\text{Run}(T)$ : $\text{CAT} \rightarrow \mathbf{C}$; for the category of monad-comonad interaction laws and cofree-cofree runner specs are monomorphisms.

Most modularity, but assuming that all Sweedler duals exist, the isomorphism of the categories of monad-comonad interaction laws and cofree-cofree runner specs follow from the following sequence of monomorphisms of categories, using that Run$(T) \cong \text{Codeg}(\zeta^T)$:

- (a) monad-comonad interaction laws.
- (b) triples of a monad $T$, a comonad $D$ and a monad-comonad interaction law $\phi : TX \rightarrow TY$.
- (c) triples of a monad $T$, a comonad $D$ and a cofree-cofree fraction map from $\text{Codeg}(D)$ to $\text{Run}(T)$.

5 Monomorph-comonoid interaction

Exploring that monads and monoidal-like objects like arrows or their monoidal facets (applicative functors) are monoidal has turned out to be very fruitful in categorical semantics (e.g. e.g. [1, 3, 12]). We now explore this perspective by abstracting monoid-comonoid interaction laws into monoid-comonoid interaction laws. This leads to further known concepts and methods from category theory.
Overview of the talk

Motivation.

Functor-functor interaction laws.

Monad-comonad interaction laws.

Sweedler dual.

Monoid-comonoid interaction laws.

Conclusion.
Functor-functor interactions

A functor-functor interaction law on a Cartesian cat. $C$ is given by two endofunctors $F, G$ with

$$\phi_{X,Y} : F_{X} \times G_{Y} \rightarrow X \times Y$$

a family of maps natural in $X$ and $Y$. 

---

Examples and no-go situations:

The archetypical example is $F_{X} = S \Rightarrow X$, $G_{Y} = S \times Y$, $\phi_{X,Y}(f, (s,y)) = (fs,y)$.

A more interesting example is $F_{X} = S \Rightarrow (S \times X)$, $G_{Y} = S \times (S \Rightarrow Y)$, $\phi_{X,Y}(f, (s,g)) =$ let $(s',x) \leftarrow fs$ in $(x,gs')$.

We can vary this last example by taking $G_{Y} = (S \Rightarrow S) \Rightarrow (S \times Y)$ and $\phi_{X,Y}(f,h) =$ let $\langle f_0, f_1 \rangle \leftarrow f$; $(s,y) \leftarrow hf_0$ in $(f_1 s, y)$.

If $F$ has a nullary operation $c_{X} : 1 \rightarrow F_{X}$, then $G_{X} \sim = 0$.

If $F$ has a comm. operation $c_{X} : X \times X \rightarrow F_{X}$, then $G_{X} \sim = 0$.

---

When $C$ is extensive then $(F,G,\phi) \Rightarrow (G,F,\phi^\ast)$ an interaction law.
A functor-functor interaction law on a Cartesian cat. $\mathcal{C}$ is given by two endofunctors $F$, $G$ with

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

a family of maps natural in $X$ and $Y$.

Examples and no-go situations:

- The archetypical example is $FX = S \Rightarrow X, GY = S \times Y, \phi(f, (s, y)) = (f \ s, y)$. 
- If $F$ has a nullary operation $c_X : 1 \to FX$, then $GX \simeq 0$.
- If $F$ has a comm. operation $c_X : X \times X \to FX$, then $GX \simeq 0$.
Functor-functor interactions

A functor-functor interaction law on a Cartesian cat. $\mathcal{C}$ is given by two endofunctors $F$, $G$ with

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

a family of maps natural in $X$ and $Y$.

Examples and no-go situations:

- The archetypical example is $FX = S \Rightarrow X$, $GY = S \times Y$, $\phi(f, (s, y)) = (f \circ s, y)$.
- A more interesting example is $FX = S \Rightarrow (S \times X)$, $GY = S \times (S \Rightarrow Y)$, $\phi(f, (s, g)) = \text{let } (s', x) \leftarrow f \circ s \text{ in } (x, g \circ s')$. 
Functor-functor interactions

A functor-functor interaction law on a Cartesian cat. $C$ is given by two endofunctors $F$, $G$ with

$$
\phi_{X,Y} : FX \times GY \to X \times Y \quad \text{a family of maps natural in } X \text{ and } Y.
$$

Examples and no-go situations:

- The archetypical example is $FX = S \Rightarrow X$, $GY = S \times Y$, $\phi(f, (s, y)) = (f \cdot s, y)$.
- A more interesting example is $FX = S \Rightarrow (S \times X)$, $GY = S \times (S \Rightarrow Y)$, $\phi(f, (s, g)) = \text{let } (s', x) \leftarrow f \cdot s \text{ in } (x, g \cdot s')$.
- We can vary this last example by taking $GY = (S \Rightarrow S) \Rightarrow (S \times Y)$ and $\phi(f, h) = \text{let } (f_0, f_1) \leftarrow f; (s, y) \leftarrow h \cdot f_0 \text{ in } (f_1 \cdot s, y)$.
Functor-functor interactions

A functor-functor interaction law on a Cartesian cat. $\mathcal{C}$ is given by two endofunctors $F, G$ with

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

a family of maps natural in $X$ and $Y$.

Examples and no-go situations:

- ✓ The archetypical example is $FX = S \Rightarrow X, GY = S \times Y, \phi(f, (s, y)) = (fs, y)$.
- ✓ A more interesting example is $FX = S \Rightarrow (S \times X), GY = S \times (S \Rightarrow Y)$,
  $\phi(f, (s, g)) = \text{let } (s', x) \leftarrow fs \text{ in } (x, gs')$.
- ✓ We can vary this last example by taking $GY = (S \Rightarrow S) \Rightarrow (S \times Y)$ and
  $\phi(f, h) = \text{let } \langle f_0, f_1 \rangle \leftarrow f; (s, y) \leftarrow h f_0 \text{ in } (f_1 s, y)$.

- ✗ If $F$ has a nullary operation $c_X : 1 \to FX$, then $GX \cong 0$. {when $\mathcal{C}$ is extensive
  - ✗ If $F$ has a comm. operation $c_X : X \times X \to FX$, then $GX \cong 0$.}
**Functor-functor interactions**

A functor-functor interaction law on a Cartesian cat. $\mathcal{C}$ is given by two endofunctors $F$, $G$ with

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

a family of maps natural in $X$ and $Y$.

Examples and no-go situations:

- The archetypical example is $FX = S \Rightarrow X$, $GY = S \times Y$, $\phi(f, (s, y)) = (fs, y)$.
- A more interesting example is $FX = S \Rightarrow (S \times X)$, $GY = S \times (S \Rightarrow Y)$,
  $$\phi(f, (s, g)) = \text{let } (s', x) \leftarrow fs \text{ in } (x, gs').$$
- We can vary this last example by taking $GY = (S \Rightarrow S) \Rightarrow (S \times Y)$ and
  $$\phi(f, h) = \text{let } \langle f_0, f_1 \rangle \leftarrow f; (s, y) \leftarrow h f_0 \text{ in } (f_1 s, y).$$
- If $F$ has a nullary operation $c_X : 1 \to FX$, then $GX \cong 0$.
- If $F$ has a comm. operation $c_X : X \times X \to FX$, then $GX \cong 0$. \{ when $\mathcal{C}$ is extensive \}

$(F, G, \phi)$ an interaction law $\rightsquigarrow (G, F, \phi^{\text{rev}})$ an interaction law (where $\phi^{\text{rev}}_{X,Y} = \sigma \circ \phi_{Y,X} \circ \sigma$).
Functor-functor interactions: the category

An functor-functor interaction law map between \((F, G, \phi), (F', G', \phi')\) is given by nat. transf. \(f: F \to F', g: G' \to G\) s.t.

\[
\begin{align*}
FX \times G'Y & \xrightarrow{f_X \times \text{id}} F'X \times G'Y \\
& \quad \xrightarrow{\phi'_{X,Y}} X \times Y \\
& \quad \xrightarrow{id \times g_Y} FX \times GY \\
& \quad \xrightarrow{\phi_{X,Y}} X \times Y 
\end{align*}
\]
Functor-functor interactions: the category

An functor-functor interaction law map between \((F, G, \phi), (F', G', \phi')\) is given by \(f : F \to F', \ g : G' \to G\) s.t.

\[
\begin{align*}
&FX \times G'Y \\
\xrightarrow{id \times gY} &FX \times GY \xrightarrow{\phi_{X,Y}} X \times Y \\
\xrightarrow{fx \times id} &F'X \times G'Y \xrightarrow{\phi'_{X,Y}} X \times Y
\end{align*}
\]

Functor-functor interaction laws form a category \(\text{IL}(\mathcal{C})\).

- The final functor-functor interaction law is \((1, 0, \phi_1), \ \ \phi_{1X,Y} = 1 \times 0 \xrightarrow{\text{snd}} 0 \xrightarrow{?} X \times Y\).
- The initial functor-functor interaction law is \((0, 1, \phi_1^{\text{rev}})\).
- Given \((F_0, G_0, \phi_0)\) and \((F_1, G_1, \phi_1)\), their product is \((F_0 \times F_1, G_0 + G_1, \phi_\times)\).

\[
\phi_{\times X,Y} = (F_0X \times F_1X) \times (G_0Y + G_1Y) \xrightarrow{\phi_{0X,Y}, \phi_{1X,Y}} (F_0X \times G_0Y) + (F_1X \times G_1Y) \xrightarrow{X + Y}
\]

- Their coproduct is \((G_0 + G_1, F_0 \times F_1, \phi_\times^{\text{rev}})\).
Functor-functor interactions: the category

An functor-functor interaction law map between \((F, G, \phi), (F', G', \phi')\) is given by nat. transfs. \(f : F \rightarrow F', g : G' \rightarrow G\) s.t.

\[
\begin{align*}
F \times G Y & \xrightarrow{\text{id} \times g_Y} X \times Y \\
F' \times G' Y & \xrightarrow{f \times \text{id}} X \times Y
\end{align*}
\]

Functor-functor interaction laws form a category \(\text{IL}(\mathcal{C})\).

- The final functor-functor interaction law is \((1, 0, \phi_1), \phi_{1X,Y} = 1 \times 0 \xrightarrow{\text{snd}} 0 \xrightarrow{?} X \times Y\).
- The initial functor-functor interaction law is \((0, 1, \phi_1^{\text{rev}})\).
- Given \((F_0, G_0, \phi_0)\) and \((F_1, G_1, \phi_1)\), their product is \((F_0 \times F_1, G_0 + G_1, \phi_\times)\).
  \[
  \phi_{\times X,Y} = (F_0 X \times F_1 X) \times (G_0 Y + G_1 Y) \xrightarrow{\phi_{0X,Y}, \phi_{1X,Y}} (F_0 X \times G_0 Y) + (F_1 X \times G_1 Y) \xrightarrow{X + Y}
  \]
- Their coproduct is \((G_0 + G_1, F_0 \times F_1, \phi_\times^{\text{rev}})\).

\(\text{IL}(\mathcal{C})\) has a composition-based monoidal structure inherited from \([\mathcal{C}, \mathcal{C}]\).
Functor-functor interactions: the dual

If $\mathcal{C}$ is Cartesian closed, then we define the dual $G^\circ$ of an endofunctor $G$ on $\mathcal{C}$ by

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

(provided that this end exists).

The dual $G^\circ$ canonically interacts with $G$: the “greatest” functor interacting with $G$. 
Functor-functor interactions: the dual

If \( \mathcal{C} \) is Cartesian closed, then we define the dual \( G^\circ \) of an endofunctor \( G \) on \( \mathcal{C} \) by

\[
G^\circ X = \int_Y GY \Rightarrow (X \times Y)
\]
(provided that this end exists).

The dual \( G^\circ \) canonically interacts with \( G \): the “greatest” functor interacting with \( G \).

We obtain alternative characterizations:

\[
\int_{X,Y} \mathcal{C}(FX \times GY, X \times Y) \cong \int_X \mathcal{C}(FX, \int_Y GY \Rightarrow (X \times Y))
\]

We only have

\[
\begin{align*}
G^\circ X &\Rightarrow \phi_{X,Y} : FX \times GY \rightarrow X \times Y \\
\phi_{X,Y} : F &\rightarrow G^\circ X \\
\phi_{X,Y} : G &\rightarrow F \circ X
\end{align*}
\]

\( \Rightarrow \) for

\( GY = 1 \)

\( GY = A \times G'Y \)

\( GY = A \Rightarrow Y \)

\( GY = A \Rightarrow G'Y \)

\( GY = G_0(G_1Y) \)
If $\mathcal{C}$ is Cartesian closed, then we define the dual $G^\circ$ of an endofunctor $G$ on $\mathcal{C}$ by

$$G^\circ X = \int_Y GY \Rightarrow (X \times Y)$$  (provided that this end exists).

The dual $G^\circ$ canonically interacts with $G$: the “greatest” functor interacting with $G$.

We obtain alternative characterizations:

$$\int_{X,Y} \mathcal{C}(FX \times GY, X \times Y) \cong \int_X \mathcal{C}(FX, \int_Y GY \Rightarrow (X \times Y)) \leadsto \left\{ \begin{array}{l} 
\phi_{X,Y} : FX \times GY \to X \times Y \\
\phi : F \to G^\circ \\
\phi : G \to F^\circ 
\end{array} \right.$$
Functor-functor interactions: the dual

If \( \mathcal{C} \) is Cartesian closed, then we define the dual \( G^\circ \) of an endofunctor \( G \) on \( \mathcal{C} \) by

\[
G^\circ X = \int_Y GY \Rightarrow (X \times Y) \quad \text{(provided that this end exists)}.
\]

The dual \( G^\circ \) canonically interacts with \( G \): the “greatest” functor interacting with \( G \).

We obtain alternative characterizations:

\[
\int_{X,Y} \mathcal{C}(FX \times GY, X \times Y) \cong \int_X \mathcal{C}(FX, \int_Y GY \Rightarrow (X \times Y)) \cong \begin{cases}
\phi_{X,Y} : FX \times GY \to X \times Y \\
\phi : F \to G^\circ \\
\phi : G \to F^\circ
\end{cases}
\]

- \( \text{Id}^\circ \cong \text{Id} \).
- For \( GY = 1 \), we have \( G^\circ X \cong 0 \).
- For \( GY = A \times G'Y \), we have \( G^\circ X \cong A \Rightarrow G'^{\circ} X \).
- For \( GY = A \Rightarrow Y \), we have \( G^\circ X \cong A \times X \).
- For \( GY = A \Rightarrow G'Y \), we only have \( A \times G'^{\circ} X \to G^\circ X \).
- For \( GY = G_0(G_1Y) \), we only have \( G_0^\circ \cdot G_1^\circ \to G^\circ \).
A monad-comonad interaction law on \( C \) is given by a monad \( T = (T, \eta, \mu) \) and a comonad \( D = (D, \varepsilon, \delta) \) together with a family \( \psi \) of maps

\[
\psi_{X,Y} : TX \times DY \rightarrow X \times Y, \quad \text{natural in } X \text{ and } Y \text{ such that}
\]

\[
\begin{align*}
& \text{id} \times \varepsilon_Y : X \times Y \\
& \eta_X \times \text{id} : TX \times DY \\
& \mu_X \times \text{id} : TX \times DY \\
& \psi_{X,Y} : TX \times DY
\end{align*}
\]

With a suitable notion of map, monad-comonad interaction laws form a category \( \text{MCIL}(C) \).
Monad-comonad interactions

A monad-comonad interaction law on $C$ is given by a monad $T = (T, \eta, \mu)$ and a comonad $D = (D, \varepsilon, \delta)$ together with a family $\psi$ of maps

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

natural in $X$ and $Y$ such that

$$\begin{align*}
\text{id} \times \varepsilon_Y &: X \times Y \\
\eta_X \times \text{id} &: TX \times DY \\
\mu_X \times \text{id} &: TX \times DY \\
\psi_{X,Y} &: X \times Y
\end{align*}$$

With a suitable notion of map, monad-comonad interaction laws form a category $\text{MCIL}(C)$.

▶ Reader monad $TX = S \Rightarrow X$ interacts with $DY = S \times Y$. 
Monad-comonad interactions

A monad-comonad interaction law on $\mathcal{C}$ is given by a monad $T = (T, \eta, \mu)$ and a comonad $D = (D, \varepsilon, \delta)$ together with a family $\psi$ of maps

$$\psi_{X,Y} : TX \times DY \to X \times Y, \quad \text{natural in } X \text{ and } Y \text{ such that}$$

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

With a suitable notion of map, monad-comonad interaction laws form a category $\text{MCIL}(\mathcal{C})$.

- Reader monad $TX = S \Rightarrow X$ interacts with $DY = S \times Y$.
- State monad $TX = S \Rightarrow (S \times X)$ interacts with $DY = S \times (S \Rightarrow Y)$. 
Monad-comonad interactions: the category

Monad-comonad interaction laws are monoid objects \(\mathbf{IL}(C)\) wrt. composition:

\[
\mathbf{MCIL}(C) \cong \mathbf{Mon}(\mathbf{IL}(C))
\]
Monad-comonad interactions: the category

Monad-comonad interaction laws are monoid objects $\text{IL}(C)$ wrt. composition:

\[ \text{MCIL}(C) \cong \text{Mon} (\text{IL}(C)) \]

Some structure on this category:

- The final monad-comonad interaction law is $(1, 0, \psi_1)$.
- The initial monad-comonad interaction law is $(\text{id}, \text{id}, \text{id})$.
- Given $(T_0, D_0, \psi_0)$ and $(T_1, D_1, \psi_1)$, their product is $(T_0 \times T_1, D_0 + D_1, \psi \times)$. 

Categories by fixing the monad $T$ or the comonad $D$:

\[ \text{MCIL}(C) | _T, - \text{ and } \text{MCIL}(C) | -, D \]

- The final object of $\text{MCIL}(C) | _T, -$ is $(T, 0, ...)$.
- The initial object of $\text{MCIL}(C) | -, D$ is $(\text{id}, D, ...)$.
Monad-comonad interactions: the category

Monad-comonad interaction laws are monoid objects $\text{IL}(C)$ wrt. composition:

$$\text{MCIL}(C) \cong \text{Mon}(\text{IL}(C))$$

Some structure on this category:

- The final monad-comonad interaction law is $(1, 0, \psi_1)$.
- The initial monad-comonad interaction law is $(\text{Id}, \text{Id}, \text{id})$.
- Given $(T_0, D_0, \psi_0)$ and $(T_1, D_1, \psi_1)$, their product is $(T_0 \times T_1, D_0 + D_1, \psi_\times)$.

Categories by fixing the monad $T$ or the comonad $D$: $\text{MCIL}(C)|_{T,-}$ and $\text{MCIL}(C)|_{-,D}$.

- The final object of $\text{MCIL}(C)|_{T,-}$ is $(T, 0, \ldots)$.
- The initial object of $\text{MCIL}(C)|_{-,D}$ is $(\text{Id}, D, \ldots)$. 
The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf.

$$\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W$$

satisfying two equations induce a monad-comonad interaction law

$$\psi'_{X,Y} : T'X \times D'Y \to X \times Y$$

of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$. 

▶ The free monad is given by $F^*X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$.

▶ The cofree comonad is given by $G^\dagger Y = \nu W. D(Y, W)$ where $D(Y, W) = Y \times GW$.

▶ Li/btw $\phi$ to a $\psi$: $(X + FZ) \times (Y \times GW) \to X \times Y + Z \times W$.

▶ Apply the previous theorem and obtain $\psi'_{X,Y}$:

$$\psi'_{X,Y} : F^*X \times G^\dagger Y \to X \times Y$$

$\psi'_{X,Y}$ is the free monad-comonad interaction law on the functor-functor interaction law $\phi$.

▶ This construction provides the interaction law we saw in the motivation!
A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf. $\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W$

satisfying two equations induce a monad-comonad interaction law

$\psi'_{X,Y} : T'X \times D'Y \to X \times Y$

of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$. 

This construction provides the interaction law we saw in the motivation!
The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf.

$$\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W$$

satisfying two equations induce a monad-comonad interaction law

$$\psi'_{X,Y} : T'X \times D'Y \to X \times Y$$

of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$.

- The free monad is given by $F^* X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$. 
The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf.

$$\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W$$

satisfying two equations induce a monad-comonad interaction law

$$\psi'_{X,Y} : T'X \times D'Y \to X \times Y$$

of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$.

- The free monad is given by $F^*X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$.
- The cofree comonad is given by $G^\dagger Y = \nu W. D(Y, W)$ where $D(Y, W) = Y \times GW$. 
The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf. 

$$\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W$$

satisfying two equations induce a monad-comonad interaction law

$$\psi'_{X,Y} : T'X \times D'Y \to X \times Y$$

of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$.

- The free monad is given by $F^* X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$.
- The cofree comonad is given by $G^\dagger Y = \nu W. D(Y, W)$ where $D(Y, W) = Y \times GW$.
- Lift $\phi$ to a $\psi : (X + FZ) \times (Y \times GW) \to X \times Y + Z \times W$. 
## The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf.

$$\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W$$

satisfying two equations induce a monad-comonad interaction law

$$\psi'_{X,Y} : T'X \times D'Y \to X \times Y$$

of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$.

- The free monad is given by $F^* X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$.
- The cofree comonad is given by $G^\dagger Y = \nu W. D(Y, W)$ where $D(Y, W) = Y \times GW$.
- Lift $\phi$ to a $\psi : (X + FZ) \times (Y \times GW) \to X \times Y + Z \times W$.
- Apply the previous theorem and obtain $\psi' : F^* X \times G^\dagger Y \to X \times Y$. 
The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf.

$$\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W$$

satisfying two equations induce a monad-comonad interaction law

$$\psi'_{X,Y} : T'X \times D'Y \to X \times Y$$

of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$.

- The free monad is given by $F^* X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$.
- The cofree comonad is given by $G^\dagger Y = \nu W. D(Y, W)$ where $D(Y, W) = Y \times GW$.
- Lift $\phi$ to a $\psi : (X + FZ) \times (Y \times GW) \to X \times Y + Z \times W$.
- Apply the previous theorem and obtain $\psi' : F^* X \times G^\dagger Y \to X \times Y$.
- $\psi'$ is the free monad-comonad interaction law on the functor-functor interaction law $\phi$. 
The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf. 
\[ \psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W \]
satisfying two equations induce a monad-comonad interaction law 
\[ \psi'_{X,Y} : T'X \times D'Y \to X \times Y \]
of $T'X = \mu Z. T(X, Z)$ and $D'Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$.

- The free monad is given by $F^* X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$.
- The cofree comonad is given by $G^\dagger Y = \nu W. D(Y, W)$ where $D(Y, W) = Y \times GW$.
- Lift $\phi$ to a $\psi : (X + FZ) \times (Y \times GW) \to X \times Y + Z \times W$.
- Apply the previous theorem and obtain $\psi' : F^* X \times G^\dagger Y \to X \times Y$.
- $\psi'$ is the free monad-comonad interaction law on the functor-functor interaction law $\phi$. 
The free monad-comonad interaction law

A parameterized monad $T$, a parameterized comonad $D$ and a nat. transf.

$$
\psi_{X,Y,W,Z} : T(X, Z) \times D(Y, W) \to X \times Y + Z \times W
$$

satisfying two equations induce a monad-comonad interaction law

$$
\psi'_{X,Y} : T'X \times D'Y \to X \times Y
$$
of $T' X = \mu Z. T(X, Z)$ and $D' Y = \nu W. D(Y, W)$.

Suppose that there is a functor-functor interaction law $\phi$ between $F$ and $G$.

- The free monad is given by $F^* X = \mu Z. T(X, Z)$ where $T(X, Z) = X + FZ$.
- The cofree comonad is given by $G^\dagger Y = \nu W. D(Y, W)$ where $D(Y, W) = Y \times GW$.
- Lift $\phi$ to a $\psi : (X + FZ) \times (Y \times GW) \to X \times Y + Z \times W$.
- Apply the previous theorem and obtain $\psi' : F^* X \times G^\dagger Y \to X \times Y$.
- $\psi'$ is the free monad-comonad interaction law on the functor-functor interaction law $\phi$.

This construction provides the interaction law we saw in the motivation!
Monad-comonad interactions: alternative phrasing (I)

The functor \((-)^\circ : [\mathcal{C}, \mathcal{C}]^{\text{op}} \to [\mathcal{C}, \mathcal{C}]\) is lax monoidal:

\[
e : \text{Id} \to \text{Id}^\circ \quad \text{m}_{F,G} : F^\circ \cdot G^\circ \to (F \cdot G)^\circ
\]
Monad-comonad interactions: alternative phrasing (I)

The functor \((-)\circ : [C,C]^\text{op} \rightarrow [C,C]\) is lax monoidal:

\[ e : \text{Id} \rightarrow \text{Id}^\circ \quad m_{F,G} : F^\circ \cdot G^\circ \rightarrow (F \cdot G)^\circ \]

Using \(e, m\), we can obtain an alternative characterization:
Monad-comonad interactions: alternative phrasing (I)

The functor \((-)^\circ : [C, C]^\text{op} \to [C, C]\) is lax monoidal:

\[ e : \text{Id} \to \text{Id}^\circ \quad \text{m}_{F,G} : F^\circ \cdot G^\circ \to (F \cdot G)^\circ \]

Using \(e\), \(m\), we can obtain an alternative characterization:

A monad-comonad interaction law can be given as a nat. transf. \(\psi : T \to D^\circ\) s.t.

\[
\begin{array}{ccc}
\text{Id} & \xrightarrow{e} & \text{Id}^\circ \\
\downarrow \eta & & \downarrow \varepsilon^\circ \\
T & \xrightarrow{\psi} & D^\circ \\
\end{array}
\quad \begin{array}{ccc}
T \cdot T & \xrightarrow{\psi \cdot \psi} & D^\circ \cdot D^\circ \\
\downarrow \mu & & \downarrow \delta^\circ \\
T & \xrightarrow{\psi} & D^\circ \\
\end{array}
\]

\(D^\circ\) is the greatest monad interacting with \(T\).
Monad-comonad interactions: alternative phrasing (I)

The functor \((-)^\circ : [C, C]^\text{op} \to [C, C]\) is lax monoidal:

\[
e : \text{Id} \to \text{Id}^\circ \quad m_{F,G} : F^\circ \cdot G^\circ \to (F \cdot G)^\circ
\]

Using \(e, m\), we can obtain an alternative characterization:

A monad-comonad interaction law can be given as a nat. transf. \(\psi : T \to D^\circ\) s.t.

Moreover, as lax monoidal functors turn monoids into monoids, \((-)^\circ\) turns comonads into monads, so we have:

A monad-comonad int. law is a monad map \(\psi\) from \(T\) to \(D^\circ = (D^\circ, e^\circ \circ e, \delta^\circ \circ m_{D,D})\). \(D^\circ\) is the greatest monad interacting with \(T\).
As a second alternative, a monad-comonad interaction law of $T$, $D$ can be given by a nat. transf. $\psi : D \to T^\circ$ s.t.

\[
\begin{align*}
\text{Id} & \xrightarrow{e} \text{Id}^\circ \\
\delta & \xrightarrow{\psi} T^\circ \\
D & \xrightarrow{\psi} T^\circ \\
\varepsilon & \xrightarrow{\eta^\circ} \\
\varepsilon & \xrightarrow{\eta^\circ} \\
D & \xrightarrow{\psi} T^\circ \\
\end{align*}
\]

\[
\begin{align*}
D \cdot D & \xrightarrow{\psi \cdot \psi} T^\circ \cdot T^\circ \\
& \xrightarrow{m_{T,T}} (T \cdot T)^\circ \\
\end{align*}
\]
As a second alternative, a monad-comonad interaction law of $T$, $D$ can be given by a nat. transf. $\psi : D \to T^\circ$ s.t.

\[
\begin{align*}
\text{Id} & \xrightarrow{e} \text{Id}^\circ \\
\epsilon & \xleftarrow{\psi} \text{Id} \xrightarrow{\eta^\circ} \text{Id}^\circ \\
\delta & \xleftarrow{\psi} \text{Id} \xrightarrow{\epsilon^{-1}} \text{Id} \\
D & \xrightarrow{\psi} T^\circ \\
\text{Id} & \xleftarrow{\eta^\circ} T^\circ \xrightarrow{\mu^\circ} T^\circ
\end{align*}
\]

The functor $(\ )^\circ : [C, C]^\text{op} \to [C, C]$ is not oplax monoidal.

$\Rightarrow$ $T^\circ$ does not necessarily come with a comonad structure.
As a second alternative, a monad-comonad interaction law of $T$, $D$ can be given by a nat. transf. $\psi : D \to T^\circ$ s.t.

\[
\begin{align*}
\text{Id} & \xrightarrow{e} \text{Id}^\circ \\
\varepsilon & \xleftarrow{e^{-1}} \eta^\circ \\
D & \xrightarrow{\psi} T^\circ
\end{align*}
\]

\[
\begin{align*}
D \cdot D & \xrightarrow{\psi \cdot \psi} T^\circ \cdot T^\circ \\
\delta & \xrightarrow{m_{T,T}} (T \cdot T)^\circ \\
D & \xrightarrow{\psi} T^\circ
\end{align*}
\]

The functor $(-)^\circ : [C, C]^{op} \to [C, C]$ is not oplax monoidal.

$\Rightarrow$ $T^\circ$ does not necessarily come with a comonad structure.

Approximate the solution? The greatest comonad smaller than $T^\circ$?
Sweedler dual of a monad

The *Sweedler dual* of the monad $T$ is the (unique up to isomorphism) comonad $T^\bullet = (T^\bullet, \eta^\bullet, \mu^\bullet)$ together with a nat. transf. $\iota : T^\bullet \to T^\circ$ s.t.

\[
\text{Id} \xrightarrow{\epsilon} \text{Id}^\circ \\
\eta^\bullet \downarrow \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \eta^\circ \downarrow \\
T^\bullet \xrightarrow{\iota} T^\circ
\]

\[
T^\bullet \cdot T^\bullet \xrightarrow{\iota \cdot \iota} T^\circ \cdot T^\circ \xrightarrow{m_{T,T}} (T \cdot T)^\circ \\
\mu^\bullet \downarrow \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \mu^\circ \downarrow \\
T^\bullet \xrightarrow{\iota} T^\circ
\]

and it is universal with respect to this property.
The *Sweedler dual* of the monad $T$ is the (unique up to isomorphism) comonad $T^\bullet = (T^\bullet, \eta^\bullet, \mu^\bullet)$ together with a nat. transf. $\iota : T^\bullet \rightarrow T^\circ$ s.t.

\[
\begin{align*}
\text{Id} & \xrightarrow{\varepsilon} D \\
\text{Id} & \xrightarrow{\eta^\bullet} T^\bullet \\
\text{Id} & \xrightarrow{\eta^\circ} T^\circ \\
\text{Id} & \xrightarrow{\iota} T^\bullet \cdot T^\bullet \\
\text{Id} & \xrightarrow{\mu^\bullet \cdot \iota} (T \cdot T)^\circ
\end{align*}
\]

and it is universal with respect to this property.
The *Sweedler dual* of the monad $T$ is the (unique up to isomorphism) comonad $T^\bullet = (T^\bullet, \eta^\bullet, \mu^\bullet)$ together with a nat. transf. $\iota : T^\bullet \to T^\circ$ s.t.

and it is universal with respect to this property.
The *Sweedler dual* of the monad $T$ is the (unique up to isomorphism) comonad $T^\bullet = (T^\bullet, \eta^\bullet, \mu^\bullet)$ together with a nat. transf. $\iota : T^\bullet \rightarrow T^\circ$ s.t.

and it is universal with respect to this property.

A monad-comonad int. law of $T$ and $D$ is the same as a comonad map from $D$ to $T^\bullet$. $T^\bullet$ is the greatest comonad interacting with $T$. 
Examples of Sweedler dual

\[TX = X^+ \cong \sum n : \mathbb{N}. ([0..n] \Rightarrow X)\]
(the nonempty list monad).

\[T \circ Y \cong \Pi n : \mathbb{N}. ([0..n] \times Y)\]
\[T \bullet Y \cong Y \times (Y + Y).\]
Examples of Sweedler dual

\[ TX = X^+ \cong \Sigma n : \mathbb{N}. ([0..n] \Rightarrow X) \]
(the nonempty list monad).

\[ TX = S \Rightarrow (S \times X) \cong (S \Rightarrow S) \times (S \Rightarrow X) \]
(the state monad).

\[ T^\circ Y \cong \Pi n : \mathbb{N}. ([0..n] \times Y) \]
\[ T^\bullet Y \cong Y \times (Y + Y). \]

\[ T^\circ Y = (S \Rightarrow S) \Rightarrow (S \times Y) \]
\[ T^\bullet Y = S \times (S \Rightarrow Y). \]
Towards monoid-comonoid interactions

Remember the Day convolution operation and its corresponding closed structure?

\[(F \star G)Z = \int^{X,Y} C(X \times Y, Z) \bullet (FX \times GY), \quad (G \rightharpoonup H)X = \int_Y GY \Rightarrow H(X \times Y)\]
Towards monoid-comonoid interactions

Remember the Day convolution operation and its corresponding closed structure?

\[(F \star G)Z = \int^{X,Y} C(X \times Y, Z) \bullet (FX \times GY), \quad (G \rightarrow H)X = \int_Y GY \Rightarrow H(X \times Y)\]

A functor-functor interaction law is a triple \((F, G, \phi : F \star G \rightarrow \text{Id}_C)\).
Towards monoid-comonoid interactions

Remember the Day convolution operation and its corresponding closed structure?

\[(F \star G)Z = \int^{X,Y} C(X \times Y, Z) \bullet (FX \times GY), \quad (G \rightarrow H)X = \int_Y GY \Rightarrow H(X \times Y)\]

A functor-functor interaction law is a triple \((F, G, \phi : F \star G \rightarrow \text{Id}_C)\).

These are Chu spaces over the object \(\text{Id}_C\) (w.r.t. \(\star\)), and there is an isomorphism

\[\text{IL}(C) \cong \text{Chu}([C, C], \text{Id}_C)\]
Towards monoid-comonoid interactions

Remember the Day convolution operation and its corresponding closed structure?

\[(F \star G)Z = \int^{X,Y} C(X \times Y, Z) \bullet (FX \times GY), \quad (G \rightharpoonup H)X = \int_Y GY \Rightarrow H(X \times Y)\]

A functor-functor interaction law is a triple \((F, G, \phi : F \star G \to \text{Id}_C)\).

These are Chu spaces over the object \(\text{Id}_C\) (w.r.t. \(\star\)), and there is an isomorphism

\[\text{IL}(C) \cong \text{Chu}([C, C], \text{Id}_C)\]

⚠️ The monoidal structure on \(\text{Chu}([C, C], \text{Id}_C)\) is constructed using pullbacks!
Replace Chu by a glueing (Hasegawa) / comma construction (Aguiar):

- Pick a duoidal category \((\mathcal{F}, I, \otimes, J, \star)\) closed wrt. \(\star\).
- Pick some monoid \(R\) wrt. \(\otimes\). We take \(R = I\) (and a general \(R\) for residual interaction).
- Let \(X^\circ = X \rightrightarrows R\). The functor \((-)^\circ : \mathcal{F}^{\text{op}} \to \mathcal{F}\) is lax monoidal wrt. \(\otimes\).
- Construct the comma category \(\mathbf{IL}(\mathcal{F}) = \mathcal{F} \downarrow (-)^\circ\).
Monoid-comonoid interactions

Replace Chu by a glueing (Hasegawa) / comma construction (Aguiar):

- Pick a duoidal category \((\mathcal{F}, I, \otimes, J, \star)\) closed wrt. \(\star\).
- Pick some monoid \(R\) wrt. \(\otimes\). We take \(R = I\) (and a general \(R\) for residual interaction).
- Let \(X^\circ = X \star R\). The functor \((-)^\circ : \mathcal{F}^{op} \to \mathcal{F}\) is lax monoidal wrt. \(\otimes\).
- Construct the comma category \(\mathbf{IL}(\mathcal{F}) = \mathcal{F} \downarrow (-)^\circ\).

- Now object-object interaction laws are \(\mathbf{IL}(\mathcal{F})\).
- \(\mathbf{IL}(\mathcal{F})\) has a monoidal structure based on \(\otimes\).
- Monoid-comonoid interaction laws are \(\mathbf{Mon}(\mathbf{IL}(\mathcal{F}))\).
Monoid-comonoid interactions

Replace Chu by a glueing (Hasegawa) / comma construction (Aguiar):

- Pick a duoidal category \((\mathcal{F}, I, \otimes, J, \star)\) closed wrt. \(\star\).
- Pick some monoid \(R\) wrt. \(\otimes\). We take \(R = I\) (and a general \(R\) for residual interaction).
- Let \(X^\circ = X \to \star R\). The functor \((-)^\circ: \mathcal{F}^{\text{op}} \to \mathcal{F}\) is lax monoidal wrt. \(\otimes\).
- Construct the comma category \(\mathbf{IL}(\mathcal{F}) = \mathcal{F} \downarrow (-)^\circ\).

- Now object-object interaction laws are \(\mathbf{IL}(\mathcal{F})\).
- \(\mathbf{IL}(\mathcal{F})\) has a monoidal structure based on \(\otimes\).
- Monoid-comonoid interaction laws are \(\text{Mon}(\mathbf{IL}(\mathcal{F}))\).

Functor-functor and monad-comonad interaction laws arise from the composition/Day convolution duoidal structure on \([\mathcal{C}, \mathcal{C}]\).
Conclusion

Summing up:

- Interaction laws describe interaction of a computation with its environment.
- Several equivalent characterizations can be given.
- The Sweedler dual approximates the dual for monads.
Conclusion

Summing up:

▶ Interaction laws describe interaction of a computation with its environment.
▶ Several equivalent characterizations can be given.
▶ The Sweedler dual approximates the dual for monads.

In the paper / preprint:

▶ Monad-comonad int. laws vs. stateful runners.
▶ The Sweedler dual for free monoids and quotients of free monoids by equations (coequalizers).
▶ Generalization to residual interaction laws to counteract degeneracies.

\[ \psi_{X,Y} : TX \times DY \to R(X \times Y) \quad \text{where } R \text{ is a monad} \]

▶ Comparison with effect handlers.
Conclusion

Summing up:

▶ Interaction laws describe interaction of a computation with its environment.
▶ Several equivalent characterizations can be given.
▶ The Sweedler dual approximates the dual for monads.

In the paper / preprint:

▶ Monad-comonad int. laws vs. stateful runners.
▶ The Sweedler dual for free monoids and quotients of free monoids by equations (coequalizers).
▶ Generalization to residual interaction laws to counteract degeneracies.

\[ \psi_{X,Y} : T X \times DY \rightarrow R(X \times Y) \quad \text{ where } R \text{ is a monad} \]

▶ Comparison with effect handlers.

Thank you for your attention!