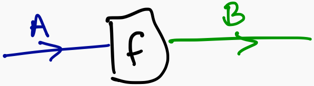
Lenses as bidirectional transformations

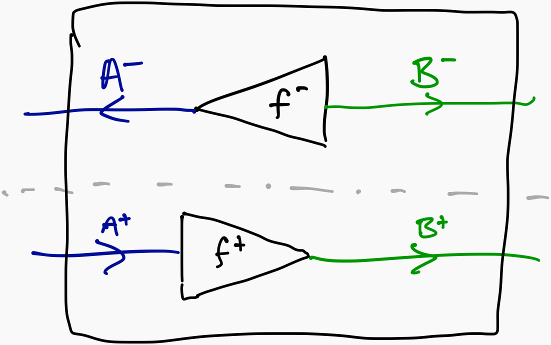
A <u>transformation</u> is a map $f : A \longrightarrow B$



Lenses as bidirectional transformations

A bidirectional transformation is a pair of maps:

$f^+ : A^+ \longrightarrow B^+ \qquad f^- : B^- \longrightarrow A^-$

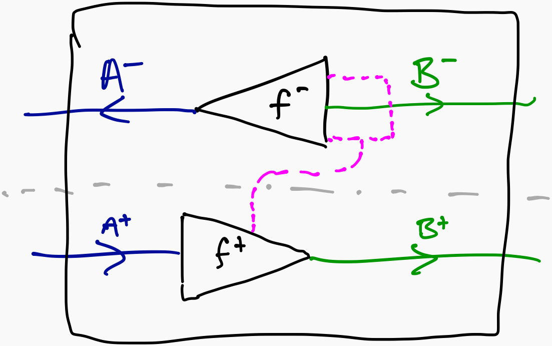


$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

Lenses as bidirectional transformations

When the domain of the backwards map depends on the forwards map, it's a Lens

$f^+ : A^+ \longrightarrow B^+ \qquad f^- : F(f^+)(B^-) \longrightarrow A^-$



$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \begin{matrix} \longleftarrow \\ \longrightarrow \end{matrix} \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

The "Lens" Family

(highly incomplete list of citations)

Lawful Lenses

- (Very) Lawful Lenses (Foster-Greenwald-Moore-Pierce-Schmidt)
- Delta Lenses (Diskin-Xiong-Czarnecki, Clarke)
- Cofunctors (Aguiar, Ahman-Uustalu, Spivak-Niu)
- Polymorphic Lenses (O'Connor, Kmett)

Lawless Lenses

- Functor Lenses (Spivak)
- Dependent Lenses
- Monadic Lenses (Pacheco-Hu-Fischer)
- Simple Lenses

The "Lens" Family (highly incomplete list of citations)

Lawful Lenses

- (Very) Lawful Lenses (Foster-Greenwald-Moore-Pierce-Schmidt)
- Delta Lenses (Diskin-Xiong-Czarnecki, Clarke)
- Cofunctors (Aguiar, Ahman-Uustalu, Spivak-Niu)
- Polymorphic Lenses (O'Connor, Kmett)

Lawless Lenses

- Functor Lenses (Spivak)
- Dependent Lenses
- Monadic Lenses (Pacheo-Hu-Fischer)
- Simple Lenses

Our focus today

The "Lens" Family — Applications

- Data accessors in functional programming
  - ↳ solving the "view/update" problem
- Database design
- Categorical approaches to
  - ↳ Machine Learning (Fong-Spivak-Tuyeras)
  - ↳ Open games (Ghani-Hedges-Winschel-Zahn)
  - ↳ (controlled) dynamical systems (Spivak, Myers, Capucci-Gavranović-Hedges-Rischel)

The "Lens" Family — Applications

- Data accessors in functional programming
- Database design
- Categorical approaches to
  - ↳ Machine Learning (Fong-Spivak-Tuyeras)
  - ↳ Open games (Ghani-Hedges-Winschel-Zahn)
  - ↳ (controlled) dynamical systems (Spivak, Myers, Capucci-Gavranović-Hedges-Rischel)

Our main focus today

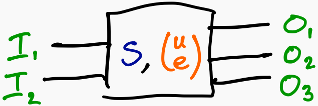
Doctrines of Systems Theory

A doctrine is a theory of what it means to be a theory of systems.

Eg:

A system is...	Systems are composed by...	Categorical tool
A diagram or graph	Gluing together features	Cospans + Pushout
A relationship amongst variable quantities	Setting exogenous variables equal	Spans + Pullback
A notion of how things can be (states), and a notion of how things can change, given how they are (dynamics)	Setting parameters of the dynamics according to the exposed variables of other systems	Lenses + Lens Composition

Moore Machines

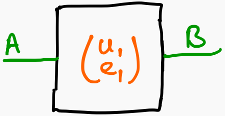


A **Moore Machine** with input alphabets $I_1, \ldots, I_n$ and output alphabets $O_1, \ldots, O_m$ is a set S of states and a pair of functions

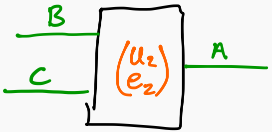
$$\begin{cases} u : S \times I_1 \times \cdots \times I_n \longrightarrow S \\ e : S \longrightarrow O_1 \times \cdots \times O_m \end{cases}$$

The state s outputs symbols $e(s)$ and transitions to state $u(s, \vec{\imath})$ when reading input symbols $\vec{\imath}$.

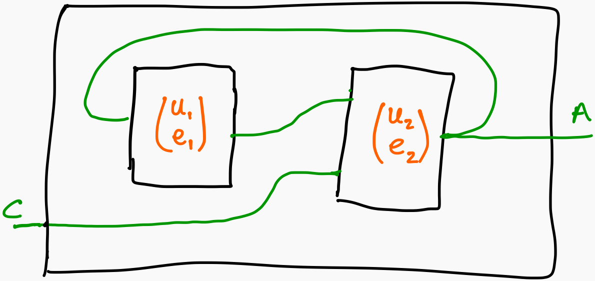
Wiring together Moore machines



$$\begin{cases} u_1 : S_1 \times A \longrightarrow S_1 \\ e_1 : S_1 \longrightarrow B \end{cases}$$



$$\begin{cases} u_2 : S_2 \times B \times C \longrightarrow S_2 \\ e_2 : S_2 \longrightarrow A \end{cases}$$



$$\begin{cases} ((s_1, s_2), c) \longmapsto \big(u_1(s_1, e_2(s_2)),\ u_2(s_2, (e_1(s_1), c))\big) \\ (s_1, s_2) \longmapsto e_2(s_2) \end{cases}$$

Moore Machines as Lenses

1

Def: A (simple) lens

$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

is a pair of functions

$$\begin{cases} f^- : A^+ \times B^- \longrightarrow A^- \\ f^+ : A^+ \longrightarrow B^+ \end{cases}$$

So, a Moore Machine is a lens

$$\begin{pmatrix} u \\ e \end{pmatrix} : \begin{pmatrix} S \\ S \end{pmatrix} \leftrightarrows \begin{pmatrix} I_1 \times \cdots \times I_n \\ O_1 \times \cdots \times O_m \end{pmatrix}$$

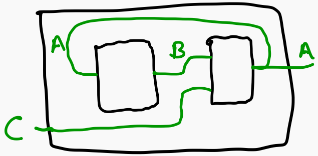
Def: A (simple) lens

$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

is a pair of functions

$$\begin{cases} f^- : A^+ \times B^- \longrightarrow A^- \\ f^+ : A^+ \longrightarrow B^+ \end{cases}$$

A wiring diagram is also a lens:



$$\begin{pmatrix} w^- \\ w^+ \end{pmatrix} : \begin{pmatrix} A \times B \times C \\ B \times A \end{pmatrix} \leftrightarrows \begin{pmatrix} C \\ A \end{pmatrix}$$

$$\begin{cases} w^-((b,a),c) = (a,b,c) \\ w^+(b,a) = a \end{cases}$$

Examples of Lenses

A

- Data access and update:

sometimes "put_x"

$$\begin{pmatrix} \mathrm{set}_x \\ \mathrm{get}_x \end{pmatrix} : \begin{pmatrix} \mathbb{R}^3 \\ \mathbb{R}^3 \end{pmatrix} \leftrightarrows \begin{pmatrix} \mathbb{R} \\ \mathbb{R} \end{pmatrix} \qquad \begin{cases} \mathrm{set}_x : \mathbb{R}^3 \times \mathbb{R} \longrightarrow \mathbb{R}^3 \\ \mathrm{get}_x : \mathbb{R}^3 \longrightarrow \mathbb{R} \end{cases}$$

$$\mathrm{get}_x(x,y,z) := x, \qquad \mathrm{set}_x((x,y,z),x') := (x',y,z)$$

This is an example of a (very) lawful lens

- (get set) $\mathrm{get}_x(\mathrm{set}_x((x,y,z),x')) = x'$
- (set get) $\mathrm{set}_x((x,y,z),\mathrm{get}_x(x,y,z)) = (x,y,z)$
- (set set) $\mathrm{set}_x(\mathrm{set}_x((x,y,z),x'),x'') = \mathrm{set}_x((x,y,z),x'')$

Lens Composition:

2

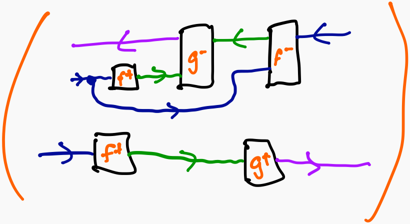
Given

$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix} \text{ and } \begin{pmatrix} g^- \\ g^+ \end{pmatrix} : \begin{pmatrix} B^- \\ B^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} C^- \\ C^+ \end{pmatrix}$$

Their composite

$$\begin{pmatrix} g^- \\ g^+ \end{pmatrix} \circ \begin{pmatrix} f^- \\ f^+ \end{pmatrix} = \begin{pmatrix} (a^+, c^-) \mapsto f^-(a^+, g^-(f^+(a^+), c)) \\ a^+ \mapsto g^+(f^+(a^+)) \end{pmatrix}$$

in string diagrams



Monoidal Product of Lenses

3

Given

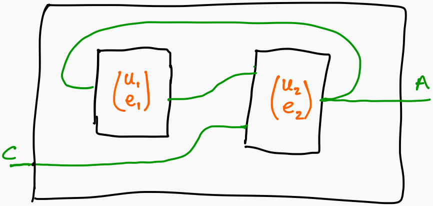
$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} g^- \\ g^+ \end{pmatrix} : \begin{pmatrix} C^- \\ C^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} D^- \\ D^+ \end{pmatrix}$$

Their product is

$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} \otimes \begin{pmatrix} g^- \\ g^+ \end{pmatrix} : \begin{pmatrix} A^- \times C^- \\ A^+ \times C^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} B^- \times D^- \\ B^+ \times D^+ \end{pmatrix}$$

$$= \begin{pmatrix} ((a^+, c^+), (b^-, d^-)) \longmapsto (f^-(a^+, b^-), g^-(c^+, d^-)) \\ (a^+, c^+) \longmapsto (f^+(a^+), g^+(c^+)) \end{pmatrix}$$

The composite system



$$\begin{cases} ((s_1, s_2), c) \longmapsto \big(u_1(s_1, e_2(s_2)),\ u_2(s_2, (e_1(s_1), c))\big) \\ (s_1, s_2) \longmapsto e_2(s_2) \end{cases}$$

is

$$\binom{w^-}{w^+} \circ \left(\binom{u_1}{e_1} \otimes \binom{u_2}{e_2} \right)$$

where

$$\binom{w^-}{w^+} : \binom{A \times B \times C}{B \times A} \leftrightarrows \binom{C}{A}$$

$$\begin{cases} w^-((b,a), c) = (a, b, c) \\ w^+(b,a) = a \end{cases}$$

Categories of Lenses:

4

The construction of the category of lenses only used the cartesian product, so...

Thm: For any category $\mathcal{C}$ with finite products, we have a symmetric monoidal category $\mathbf{Lens}_{\mathcal{C}}$ of lenses in $\mathcal{C}$,

↳ objects $\binom{A^-}{A^+}$, morphisms $\binom{f^-}{f^+} : \binom{A^-}{A^+} \leftrightarrows \binom{B^-}{B^+}$

For a product-preserving functor $F : \mathcal{C} \to \mathcal{D}$, get $\binom{F}{F} : \mathbf{Lens}_{\mathcal{C}} \to \mathbf{Lens}_{\mathcal{D}}$

Wiring diagrams as free lenses

Thm: The free cartesian category on an object X, "Arity" has objects: X^F for finite sets F,

morphisms: $f^*: X^F \to X^{F'}$ for $f: F' \to F$

ie $\text{Arity} \simeq \text{Fin}^{op}$.

Def: A wiring diagram is a lens in Arity.

input ports   input ports

$$\begin{pmatrix} W^{*}_{-} \\ W^{*}_{+} \end{pmatrix} : \underbrace{\begin{pmatrix} X^{B_1^-} \\ X^{B_1^+} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} X^{B_n^-} \\ X^{B_n^+} \end{pmatrix}}_{\text{inner boxes}} \leftrightarrows \underbrace{\begin{pmatrix} X^{B^-} \\ X^{B^+} \end{pmatrix}}_{\text{outer box}}$$

output ports   output ports

Def: A wiring diagram is a lens in Arity.

input ports   input ports

$$\begin{pmatrix} w_-^* \\ w_+^* \end{pmatrix} : \underbrace{\begin{pmatrix} x^{B_1^-} \\ x^{B_1^+} \end{pmatrix} \otimes \cdots \otimes \begin{pmatrix} x^{B_n^-} \\ x^{B_n^+} \end{pmatrix}}_{\text{inner boxes}} \leftrightarrows \underbrace{\begin{pmatrix} x^{B^-} \\ x^{B^+} \end{pmatrix}}_{\text{outer box}}$$

output ports   output ports

That is:

$$\begin{cases} w_- : B_1^- + \cdots + B_n^- \longrightarrow B_1^+ + \cdots + B_n^+ + B^- \\ w_+ : B^+ \longrightarrow B_1^+ + \cdots + B_n^+ \end{cases}$$

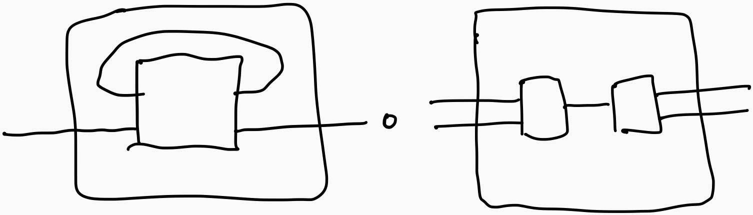
i.e.

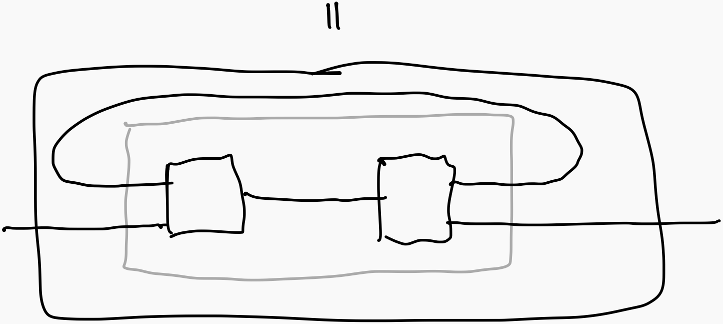
1. every outer output b^+ comes from exactly one inner output $w_+(b^+)$.

2. every inner input b_i^- comes from either an inner output or an outer input $w_-(b_i^-)$

Composition of wiring diagrams is by nesting:

6





The universal property of $\mathbf{Arity}$ says that for any $C \in \mathcal{C}$ there is a unique cartesian functor $X \mapsto C : \mathbf{Arity} \to \mathcal{C}$

This gives us $\begin{pmatrix} X \mapsto C \\ X \mapsto C \end{pmatrix} : \mathbf{Lens}_{\mathbf{Arity}} \to \mathbf{Lens}_{\mathcal{C}}$ which interprets every wiring diagram as a lens in $\mathcal{C}$!

The Parameter-Setting Doctrine of Systems

A system consists of
- a notion of how things may be (State)
- a notion of how things change, given how they are (dynamics)

A system exposes variables of state, and the dynamics may depend on parameters that can be set by the exposed variables of other systems.

E.g.:
- A Moore Machine with input alphabet I and output alphabet O is a set S of states and a pair of maps

$$\begin{cases} u : S \times I \longrightarrow S \\ e : S \longrightarrow O \end{cases}$$

The state s outputs symbol $e(s)$ and transitions to state $u(s,i)$ when reading input symbol i.

The Parameter-Setting Doctrine of Systems

A system consists of
- a notion of how things may be (State)
- a notion of how things change, given how they are (dynamics)

A system exposes variables of state, and the dynamics may depend on parameters that can be set by the exposed variables of other systems.

E.g.:

- A Markov Decision Process with action menu A and outputs O is a set S of states and a pair of maps

$$\begin{cases} u : S \times A \longrightarrow D(\mathbb{R} \times S) \\ e : S \longrightarrow O \end{cases}$$

If $u(s,a) = \sum_{(r,s')} p_{(r,s')}(r,s')$, then $p_{(r,s')}$ is the probability that action a causes the transition $s \rightsquigarrow s'$ with reward r.

The Parameter-Setting Doctrine of Systems

A system consists of
- a notion of how things may be (State)
- a notion of how things change, given how they are (dynamics)

A system exposes variables of state, and the dynamics may depend on parameters that can be set by the exposed variables of other systems.

E.g.:
- A system of ODEs $\left\{ \frac{ds}{dt} = u(s, p) \right.$ with k parameters p is a manifold S of states and a pair of maps

$$\left\{ \begin{tikzcd} TS \arrow[d, "\pi"'] & e^*I \arrow[l, "u"'] \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & I \arrow[d, "\pi"] \\ S \arrow[r, equal] & S \arrow[r, "e"'] & O \end{tikzcd} \right.$$

Moore Machines
$$\begin{cases} u : S \times I \longrightarrow S \\ e : S \longrightarrow O \end{cases}$$
Simple

Markov Decision Processes
$$\begin{cases} u : S \times I \longrightarrow D(\mathbb{R} \times S) \\ e : S \longrightarrow O \end{cases}$$
Monadic

Systems of ODEs
$$\begin{tikzcd} TS \arrow[d, "\pi"'] & e^*I \arrow[l, "u"'] \arrow[r] \arrow[d] \arrow[dr, phantom, very near start, "\lrcorner"] & I \arrow[d, "\pi"] \\ S \arrow[r, equal] & S \arrow[r, "e"'] & O \end{tikzcd}$$
dependent

are all Lenses of different kinds

$$\begin{pmatrix} u \\ e \end{pmatrix} : \begin{pmatrix} TS \\ S \end{pmatrix} \leftrightarrows \begin{pmatrix} I \\ O \end{pmatrix}$$

Lenses as Spans

We can rearrange the data of a lens as follows

$$\begin{pmatrix} f^- \\ f^+ \end{pmatrix} : \begin{pmatrix} A^- \\ A^+ \end{pmatrix} \leftrightarrows \begin{pmatrix} B^- \\ B^+ \end{pmatrix}$$

$$\begin{cases} f^- : A^+ \times B^- \longrightarrow A^- \\ f^+ : A^+ \longrightarrow B^+ \end{cases}
\qquad
\left\{ \begin{tikzcd}
A^+ \times A^- \arrow[d] & A^+ \times B^- \arrow[l, "{(A^+; f^-)}"'] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & B^+ \times B^- \arrow[d] \\
A^+ \arrow[r, equal] & A^+ \arrow[r, "f^+"'] & B^+
\end{tikzcd} \right.$$

$A \leftarrow \bullet \rightarrow B$

The right hand side is a span in the category $\mathcal{C}^{\downarrow}$ of arrows in $\mathcal{C}$, whose

1. left leg is vertical — the bottom is iso
2. right leg is cartesian — a pullback

This gives us a more general way to think of lenses...

Indexed Categories and the Grothendieck Construction

Def: An indexed category is a pseudofunctor $\mathcal{F} : \mathcal{C}^{op} \to \mathrm{Cat}$.

- For every $c \in \mathcal{C}$, a category $\mathcal{F}(c)$
- For every $f : c' \to c$, a functor $\mathcal{F}(f) : \mathcal{F}(c) \to \mathcal{F}(c')$
- Natural isomorphisms

$$\eta_c : \mathrm{id}_{\mathcal{F}(c)} \xrightarrow{\sim} \mathcal{F}(\mathrm{id}_c)$$
$$\mu_c : \mathcal{F}(f) \circ \mathcal{F}(g) \xrightarrow{\sim} \mathcal{F}(g \circ f)$$

Satisfying unit and associativity laws (like those of a lax monoidal functor).

Indexed Categories and the Grothendieck Construction

Def: An **indexed category** is a **pseudofunctor** $\mathcal{F} : \mathcal{C}^{op} \to \mathrm{Cat}$.

- For every $c \in \mathcal{C}$, a category $\mathcal{F}(c)$
- For every $f : c' \to c$, a functor $\mathcal{F}(f) : \mathcal{F}(c) \to \mathcal{F}(c')$
- Natural isomorphisms

$$\eta_c : \mathrm{id}_{\mathcal{F}(c)} \xrightarrow{\sim} \mathcal{F}(\mathrm{id}_c), \qquad \mu_c : \mathcal{F}(f) \circ \mathcal{F}(g) \xrightarrow{\sim} \mathcal{F}(g \circ f)$$

Def: The **Grothendieck Construction** $\int^{c:\mathcal{C}} \mathcal{F}(c)$ of an indexed cat has

- objects $\begin{pmatrix} F \\ c \end{pmatrix}$ with $c \in \mathcal{C}$ and $F \in \mathcal{F}(c)$
- maps $\begin{pmatrix} \bar{f} \\ f \end{pmatrix} : \begin{pmatrix} F \\ c \end{pmatrix} \rightrightarrows \begin{pmatrix} F' \\ c' \end{pmatrix}$ are pairs $\begin{cases} \bar{f} : F \to \mathcal{F}(f)(F') \\ f : c \to c' \end{cases}$
- Composition is by

$$\begin{pmatrix} \bar{g} \\ g \end{pmatrix} \circ \begin{pmatrix} \bar{f} \\ f \end{pmatrix} = \begin{pmatrix} F \xrightarrow{\bar{f}} \mathcal{F}(f)(F') \xrightarrow{\mathcal{F}(f)(\bar{g})} \mathcal{F}(f)(\mathcal{F}(g)(F'')) \xrightarrow[\sim]{\mu} \mathcal{F}(g \circ f)(F'') \\ c \xrightarrow{f} c' \xrightarrow{g} c'' \end{pmatrix}$$

Def: The **Grothendieck Construction** $\int^{c:\mathcal{C}} \mathcal{F}(c)$ of an indexed cat has

- objects $\binom{F}{c}$ with $c \in \mathcal{C}$ and $F \in \mathcal{F}(c)$
- maps $\binom{\bar f}{f} : \binom{F}{c} \rightrightarrows \binom{F'}{c'}$ are pairs $\begin{cases} \bar f : F \to \mathcal{F}(f)(F') \\ f : c \to c' \end{cases}$
- composition is by

$$\binom{\bar g}{g} \circ \binom{\bar f}{f} = \begin{pmatrix} F \xrightarrow{\bar f} \mathcal{F}(f)(F') \xrightarrow{\mathcal{F}(f)(\bar g)} \mathcal{F}(f)(\mathcal{F}(g)(F'')) \xrightarrow[\sim]{\mu} \mathcal{F}(g \circ f)(F'') \\ c \xrightarrow{f} c' \xrightarrow{g} c'' \end{pmatrix}$$

Def: A map $\binom{\bar f}{f}$ is **vertical** if f is iso and **Cartesian** if $\bar f$ is iso.

Thm: Every map $\binom{\bar f}{f}$ factors uniquely as a vertical followed by a cartesian map:

$$\binom{\bar f}{f} = \binom{\mathrm{id}_{\mathcal{F}(f)(F')}}{f} \circ \binom{\eta_c \circ \bar f}{\mathrm{id}}$$

Lenses via the Grothendieck Construction

Def (Spivak): Let $\mathcal{F} : \mathcal{C}^{op} \to \mathbf{Cat}$ be an indexed category. The category of $\mathcal{F}$-lenses is

$$\mathbf{Lens}_{\mathcal{F}} := \int^{C:\mathcal{C}} \mathcal{F}(C)^{op}$$

ie objects $\binom{A^-}{A^+}$ with $A^+ \in \mathcal{C}$ and $A^- \in \mathcal{F}(A^+)$

morphisms $\binom{f^-}{f^+} : \binom{A^-}{A^+} \leftrightarrows \binom{B^-}{B^+}$

$$\begin{cases} f^- : \mathcal{F}(f^+)(B^-) \longrightarrow A^- \\ f^+ : A^+ \longrightarrow B^+ \end{cases}$$

with composition

$$\left(\begin{array}{c} \mathcal{F}(g^+ \circ f^+)(C^-) \xrightarrow{\sim} \mathcal{F}(f^+)\mathcal{F}(g^+)(C^-) \xrightarrow{\mathcal{F}(f^+)(g^-)} \mathcal{F}(f^+)(B^-) \xrightarrow{f^-} A^- \\ A^+ \xrightarrow{f^+} B^+ \xrightarrow{g^+} C^+ \end{array} \right)$$

Type of System | **Indexed category** $F : \mathcal{C}^{op} \to Cat$

Moore Machines

$$\begin{cases} u : S \times I \to S \\ e : S \to O \end{cases}$$

$$C \mapsto \left\{ \begin{tikzcd} C \times X \arrow[rr] \arrow[dr] & & C \times Y \arrow[dl] \\ & C & \end{tikzcd} \right\}$$

Markov Decision Processes

$$\begin{cases} u : S \times I \to \mathbb{D}(\mathbb{R} \times S) \\ e : S \to O \end{cases}$$

$$C \mapsto \mathrm{Bikl}(C \times -, \mathbb{D}(\mathbb{R} \times -))$$

Systems of ODEs

$$\left\{ \begin{tikzcd} TS \arrow[d, "\pi"'] & e^{*}I \arrow[l, "u"'] \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & I \arrow[d, "\pi"] \\ S \arrow[r, equal] & S \arrow[r, "e"'] & O \end{tikzcd} \right.$$

$$C \mapsto \left\{ \begin{tikzcd} M \arrow[rr] \arrow[dr, "\text{subm}"'] & & N \arrow[dl, "\text{subm}"] \\ & C & \end{tikzcd} \right\}$$