1 Choice

1.1 10.2.

Given a monoid structure on hom-sets, i.e.,

$$\begin{aligned}
\oplus &\colon [A \to B] \times [A \to B] \to [A \to B] \\
\mathrm{fail} &\colon [A \to B] \text{ (or } [I \to B]) \\
(f \oplus g) \oplus h &= f \oplus (g \oplus h) && \text{(associativity)} \\
\mathrm{fail} \oplus f &= f = f \oplus \mathrm{fail} && \text{(unit)} \\
f\,;(g \oplus h) &= (f\,;g) \oplus (f\,;h) && \text{(naturality in } A\text{)} \\
(f \oplus g)\,;h &= (f\,;h) \oplus (g\,;h) && \text{(naturality in } B\text{)} \\
(f \oplus^{\mathcal{D}} g)^{\dagger} &= f^{\dagger} \oplus^{\mathcal{D}} g^{\dagger} && \text{(conversibility)} \\
(f \oplus^{\mathcal{D}} g)\nearrow &= f\nearrow \oplus^{\mathcal{C}} g\nearrow && \text{(conversibility)}
\end{aligned}$$

we can define semantics for `case`-expressions:

$$[\![\mathtt{case}\ e_0\ \mathtt{of}\ p_1 \Rightarrow e_1; \ldots\ p_n \Rightarrow e_n;]\!] = [\![e_0]\!]\ ;([\![p_1]\!]^{\dagger}\ ;[\![e_1]\!] \oplus \ldots \oplus [\![p_n]\!]^{\dagger}\ ;[\![e_n]\!])$$

For example use Kleisli-category over monad $A + 1$:

$$\begin{aligned}
\mathsf{sel}_A &\colon A \times A \cong [2 \to A] && (1) \\
\mathtt{if}\ a\ \mathtt{then}\ b\ \mathtt{else}\ c &= \mathsf{sel}\,(\lambda().b, \lambda().c)\,a\,() && (2) \\
- \downarrow &\colon [A \to B] \to [A \to 2] && (3) \\
\mathrm{fail} &= \mathsf{inj}_2 && (4) \\
\mathsf{rmb} &\colon A \to [I \leftrightarrows A] && (5) \\
(\mathsf{rmb}\,a)\nearrow &= \lambda().a && (6) \\
(\mathsf{rmb}\,a)\searrow &= \lambda a'.() && (7) \\
f \oplus g &= \mathrm{jV}\,(\mathsf{rmb}\,f\nearrow\downarrow)\,;\mathrm{swap}\,;\mathrm{jV}\,(\mathsf{sel}\,(f,g))\,;\mathrm{swap}\,;(\mathrm{jV}\,(\mathsf{rmb}\,f\searrow\downarrow))^{\dagger} && (8) \\
&= \begin{pmatrix} \lambda a.\mathtt{if}\ f\nearrow\downarrow\ a\ \mathtt{then}\ f\nearrow a\ \mathtt{else}\ g\nearrow a \\ \lambda a.\mathtt{if}\ f\searrow\downarrow\ a\ \mathtt{then}\ f\searrow a\ \mathtt{else}\ g\searrow a \end{pmatrix} && (9)
\end{aligned}$$

Note that $(\mathsf{rmb}\,a)\nearrow\,;(\mathsf{rmb}\,a)\searrow = \mathrm{id}_I$ for any type A.
Problem for category of partial isomorphisms (for example):

$$(f \oplus g)\nearrow\,;(f \oplus g)\searrow = f\nearrow\,;f\searrow \oplus \ f\nearrow\,;g\searrow\ \oplus\ g\nearrow\,;f\searrow \oplus g\nearrow\,;g\searrow$$

Define partial isomorphism restriction of `rmb` for types with computable equality:

$$\begin{aligned}
\mathsf{dup} &\colon A \to [I \leftrightarrows A] \\
(\mathsf{dup}\,a)\nearrow &= \lambda().a \\
(\mathsf{dup}\,a)\searrow &= \lambda a'.\mathtt{if}\ a = a'\ \mathtt{then}\ ()\ \mathtt{else}\ \mathrm{fail}
\end{aligned}$$

so $(\mathsf{dup}\, a)\nearrow ; (\mathsf{dup}\, a)\searrow = \mathrm{id}_I$ and $(\mathsf{dup}\, a)\searrow\downarrow\, a' \implies ((\mathsf{dup}\, a)\searrow ; (\mathsf{dup}\, a)\nearrow)\, a' = a'$.

So define

$$f \oplus g = \mathrm{jV}\,(\mathsf{dup}\, f\nearrow\downarrow) ; \mathrm{swap} ; \mathrm{jV}\,(\mathsf{sel}\,(f,g)) ; \mathrm{swap} ; \mathrm{jV}\,(\mathsf{dup}\, f\searrow\downarrow)$$

$$= \begin{pmatrix} \lambda a.\mathtt{if}\ f\nearrow\downarrow\ a\ \mathtt{then}\ f\nearrow a\ \mathtt{else} & \begin{pmatrix} \mathtt{let}\ y = g\nearrow a\ \mathtt{in} \\ \mathtt{if}\ \neg f\searrow\downarrow\ y\ \mathtt{then}\ y\ \mathtt{else}\ \mathrm{fail} \end{pmatrix} \\ \ldots & \end{pmatrix}$$

(symmetric first-match policy, Yokoyama 2012)

Problem here: $(f \oplus^{\mathcal{D}} g)\nearrow \neq f\nearrow \oplus^{\mathcal{C}} g\nearrow$ because $f\searrow$ is required to define $\oplus^{\mathcal{D}}$ but $-\searrow$ doesn't exist for morphisms in $\mathcal{C}$.

The (somewhat) ugly solution is to have two different `case`-expressions. The one that only works for $\mathcal{D}$ would still be usable in purely $\mathcal{C}$-programs. So $\mathtt{case}^{\mathcal{D}}$ would be defined only for cases which use variables linearly and would use first-match-policy and would be usable in reversible and irreversible functions. $\mathtt{case}^{\mathcal{C}}$ is the normal one and only usable in irreversible functions. So we can keep $[\![\mathtt{case}^{\mathcal{D}} \ldots]\!]\nearrow = [\![\mathtt{case}^{\mathcal{C}} \ldots]\!]$.

Other option in literature is to look at structure of final expressions in each case and ensure at compile-time that cases don't intersect in reverse. New idea: do this on the type level. So instead of having to write cases like $\mathtt{case} \ldots \mathtt{of} \ldots \Rightarrow \ldots \mathtt{in}\, \mathit{Left}\, x; \ldots \Rightarrow \ldots \mathtt{in}\, \mathit{Right}\, y$; we can write any expression with types $\mathit{Left}\, A$ and $\mathit{Right}\, B$, i.e., they don't intersect, resulting in the union type $\mathit{Left}\, A \cup \mathit{Right}\, B$. Needs union types and decidable check that intersection is empty.

Other (minor) problem: case can fail in expression as well as pattern. Not typically how functional languages work.

1.2 11.3. – "Pure" pattern matching

Irreversibly An irreversible pattern matching for some type T is a natural transformation $rec_{T,C} : (\textit{something simpler than } [T \to C]) \to [T \to C]$.

- **Variable** A
  Iso: $rec_A : [A \to C] \cong [A \to C]$
  $[\![\lambda v.e : [A \to C]]\!] : \Gamma \to [A \to C] = \lambda\gamma.\lambda().rec_A\,([\![v : A \vdash e : C]\!]\ \gamma)$
- **Product** $A \times B$
  Iso: $rec_\times : [A \times B \to C] \cong [A \times B \to C]$
  $[\![\lambda(v_1, v_2).e : [A \times B \to C]]\!] = rec_\times\ [\![v_1 : A, v_2 : B \vdash e : C]\!]$
- **Co-Product** $A + B$
  Iso: $rec_+ : [A \to C] \times [B \to C] \cong [A + B \to C]$
  $[\![\lambda\iota_1\, v_1.e_1 \mid \iota_2\, v_2.e_2 : [A + B \to C]]\!] = rec_+\,([\![v_1 : A \vdash e_1 : C]\!], [\![v_2 : B \vdash e_2 : C]\!])$
- **Recursive Type** T
  Iso: $f : T \cong F\,T$ (usually via initial F-Algebra (T, f^{-1}))
  Example T is list of A: $F\,X = 1 + A \times X$
  $[\![\lambda\iota_1().e_1 \mid \iota_2(v_1, v_2).e_2 : [T \to C]]\!] = f\,;\, [\![\lambda \ldots : [1 + A \times T \to C]]\!]$
  (no syntax in equi-recursive type system)

Reversibly A pattern matcher for some type T is a natural transformation $rec_{T,C}\colon$ (*something simpler than* $[T \leftrightarrows C]) \to [T \leftrightarrows C]$.

- **Variable** $v\colon A$
  Iso: $rec_A\colon [A \leftrightarrows C] \cong [A \leftrightarrows C]$
  $[\![\lambda v.e\colon [A \leftrightarrows C]]\!] = rec_A\ [\![v\colon A \vdash e\colon A]\!]$
- ~~**Product** $A \times B$~~ **Tensor-Product** $A \otimes B$
  Iso: $rec_\otimes\colon [A \otimes B \leftrightarrows C] \cong [A \otimes B \leftrightarrows C]$
  $[\![\lambda(v_1, v_2).e\colon [A \times B \leftrightarrows C]]\!] = rec_\times\ [\![v_1\colon A, v_2\colon B \vdash e\colon C]\!]$
- **Recursive Type** T
  Iso: $f\colon T \cong F\,T$ (initial F-Algebra (T, f) is also a terminal F-Coalgebra in a dagger category with F a dagger functor)
  Example T is list of A: $F\,X = 1 + A \times X$
  $[\![\lambda\iota_1().e_1 \mid \iota_2(v_1, v_2).e_2\colon [T \leftrightarrows C]]\!] = f\,;\,[\![\lambda \ldots : [1 + A \times T \leftrightarrows C]]\!]$
- ~~**Co-Product** $A + B$~~ **Tensor-Product** $A \oplus B$
  Iso: $rec'_\oplus\colon [A \leftrightarrows C] \times [B \leftrightarrows D] \cong [A \oplus B \leftrightarrows C \oplus D]$
  $[\![\Gamma; \emptyset \vdash \lambda\iota_1\, v_1.\iota_1\, e_1 \mid \iota_2\, v_2.\iota_2\, e_2]\!] = rec'_\oplus\,([\![\lambda v_1.e_1]\!]\ \gamma, [\![\lambda v_2.e_2]\!]\ \gamma)$
  Preserves all thinkable properties of a janus. Which ones?
- ~~**Co-Product** $A + B$~~ **Biproduct** $A \oplus B$
  Iso: $rec_\oplus\colon [A \leftrightarrows C] \times [B \leftrightarrows C] \cong [A \oplus B \leftrightarrows C]$
  $[\![\Gamma; \emptyset \vdash \lambda\iota_1 v_1.e_1 \mid \iota_2 v_2.e_2]\!] = rec_\oplus\,([\![\lambda v_1.e_1]\!]\ \gamma, [\![\lambda v_2.e_2]\!]\ \gamma)$
  In Set we have

$$C^A \times C^B \cong C^{A+B} \tag{10}$$

$$A^C \times B^C \ncong (A + B)^C \tag{11}$$

  Is there a monad F such that

$$(F\,C)^A \times (F\,C)^B \cong (F\,C)^{A+B} \tag{12}$$

$$(F\,A)^C \times (F\,B)^C \cong (F\,(A + B))^C \tag{13}$$

  $F\,X = D$ trivial. $F\,X = D^X$ contravariant – except for $D = 2 \implies$ covariant powerset monad. Does not preserve isomorphisms. $\mathsf{Set}_\mathcal{P} \cong \mathsf{Rel}$. What about $\mathsf{Set}_{\mathcal{P}_{<\infty}}$? Other solutions for F?

1.3 11.3. – "Impure" pattern matching

Monoid on hom-sets $([A \to B], \oplus, \mathrm{fail}_{A,B})$.
$[\![\mathtt{case}^\mathcal{D} \ldots]\!] \nearrow \leq [\![\mathtt{case}^\mathcal{C} \ldots]\!]$?
$[\![\mathtt{case}^\mathcal{D} \ldots]\!] \nearrow \oplus [\![\mathtt{case}^\mathcal{C} \ldots]\!] = [\![\mathtt{case}^\mathcal{C} \ldots]\!] = [\![\mathtt{case}^\mathcal{C} \ldots]\!] \oplus [\![\mathtt{case}^\mathcal{D} \ldots]\!] \nearrow$
Monoid induces partial order $a \leq b \iff a \oplus b = b = b \oplus a$ if $a \oplus a = a$.

2 20.4. – Nambooripad order

Assume $f\nearrow ; f\searrow \leq \mathrm{id}$ (i.e., $f\nearrow ; f\searrow \oplus \mathrm{id} = \mathrm{id} = \mathrm{id} \oplus f\nearrow ; f\searrow$) and $g\nearrow ; g\searrow \leq \mathrm{id}$.

What does $(f\nearrow \oplus g\nearrow) ; (f\searrow \oplus g\searrow) \leq \mathrm{id}$ imply?

$$f\nearrow ; f\searrow \oplus f\nearrow ; g\searrow \oplus g\nearrow ; f\searrow \oplus \mathrm{id} = \mathrm{id} \tag{14}$$
$$\mathrm{id} \oplus f\nearrow ; g\searrow \oplus g\nearrow ; f\searrow \oplus g\nearrow ; g\searrow = \mathrm{id} \tag{15}$$

or

$$f\nearrow ; f\searrow \,/\, (f\nearrow ; g\searrow \oplus g\nearrow ; f\searrow \oplus \mathrm{id}) = f\nearrow ; f\searrow \,/\, \mathrm{id} \tag{16}$$
$$(\mathrm{id} \oplus f\nearrow ; g\searrow \oplus g\nearrow ; f\searrow) \setminus g\nearrow ; g\searrow = \mathrm{id} \setminus g\nearrow ; g\searrow \tag{17}$$

3 20.4. – Monoid enrichment

Exercise:

- Free monoid on set A: A^*, concatenation, empty list.
- Free commutative monoid on A: $A \to \mathbb{N}$ with finite non-zero/finite multiset, point-wise addition/$\cup$, $\lambda n.0/\emptyset$.
- Free idempotent monoid on A: A^* without any duplicated subsequence, concatenate and remove duplicated subsequences, empty list.
- Coproduct of two monoids $(A, +_A, 0_A)$ and $(B, +_B, 0_B)$: $(A \setminus \{0\} + B \setminus \{0\})^*$ alternating, concat+reduce, empty list.
- Coproduct of two commutative monoids: $A \times B$, pointwise, $(0_A, 0_B)$. (Need commutativity for $[f, g]$ a homomorphism)
- Coproduct of two idempotent monoids: same as normal but with duplicated subsequences removed

Equivalence classes/idempotent functions/closure operators?

Tensorproduct of (idempotent) monoids?

S.M. – Given monoids M and N, $M \otimes N$ is a monoid with a monoid morphism $\eta\colon M \times N \to M \otimes N$ such that for all bimorphisms $f\colon M \times N \to C$ there exists a unique monoid morphism $\bar{f}\colon M \otimes N \to C$ such that $\eta ; \bar{f} = f$.

This means that the bimorphism $;\colon [A \to B] \times [B \to C] \to [A \to C]$ corresponds one-to-one to a monoid morphism $\bar{;}\colon [A \to B] \otimes [B \to C] \to [A \to C]$.

3.1 17.5. – Doesn't work, but almost does

Let Set_* be the category of pointed sets and functions that preserve the point. Set_* is enriched over Set via:

$$\mathsf{Set}_*(A,B) \in \mathsf{Set} \tag{18}$$
$$= \{f \in \mathsf{Set}(A,B) \mid f(*_A) = *_B\} \tag{19}$$
$$-\,;^{\mathsf{Set}_*}\,- : \mathsf{Set}_*(A,B) \times \mathsf{Set}_*(B,C) \to \mathsf{Set}_*(A,C) \tag{20}$$
$$= -\,;^{\mathsf{Set}}\,- \tag{21}$$
$$\mathrm{id}^{\mathsf{Set}_*} : 1 \to \mathsf{Set}_*(A,A) \tag{22}$$
$$= \lambda().\mathrm{id}_A^{\mathsf{Set}} \tag{23}$$

Every $\mathsf{Set}_*(A,B)$ is a monoid-object in Set as follows:

$$\eta_{\mathsf{Set}_*(A,B)} : 1 \to \mathsf{Set}_*(A,B) \tag{24}$$
$$= \lambda().\lambda a.*_B \tag{25}$$
$$\mu_{\mathsf{Set}_*(A,B)} : \mathsf{Set}_*(A,B) \times \mathsf{Set}_*(A,B) \to \mathsf{Set}_*(A,B) \tag{26}$$
$$= \lambda(f,g).\lambda a. \begin{cases} f(a) \text{ if } f(a) \neq *_B \\ g(a) \text{ otherwise} \end{cases} \tag{27}$$

In fact, $\eta\colon 1 \to \mathsf{Set}_*(-,-)$ and $\mu\colon \mathsf{Set}_*(-,-)^2 \to \mathsf{Set}_*(-,-)$ are natural transformations (where 1, $\mathsf{Set}_*(-,-)$ and $\mathsf{Set}_*(-,-)^2$ are functors from $\mathsf{Set}_*^{\mathrm{op}} \times \mathsf{Set}_* \to \mathsf{Set}$), i.e., the following commutes for all $f\colon A' \to A$ and $g\colon B \to B'$ in Set_*:

$$\begin{array}{ccc} 1 & \xrightarrow{\mathrm{id}_1} & 1 \\ \downarrow{\scriptstyle \eta_{\mathsf{Set}_*(A,B)}} & & \downarrow{\scriptstyle \eta_{\mathsf{Set}_*(A',B')}} \\ \mathsf{Set}_*(A,B) & \xrightarrow{\mathsf{Set}_*(f,g)} & \mathsf{Set}_*(A',B') \end{array}$$

i.e., $f\,;\eta\,;g = \eta$

$$\begin{array}{ccc} \mathsf{Set}_*(A,B) \times \mathsf{Set}_*(A,B) & \xrightarrow{\mathsf{Set}_*(f,g)\times\mathsf{Set}_*(f,g)} & \mathsf{Set}_*(A',B') \times \mathsf{Set}_*(A',B') \\ \downarrow{\scriptstyle \mu_{\mathsf{Set}_*(A,B)}} & & \downarrow{\scriptstyle \mu_{\mathsf{Set}_*(A',B')}} \\ \mathsf{Set}_*(A,B) & \xrightarrow{\mathsf{Set}_*(f,g)} & \mathsf{Set}_*(A',B') \end{array}$$

i.e., $(f\,;h\,;g)\ \mu\ (f\,;h'\,;g) = f\,;(h\ \mu\ h')\,;g$

Enrichment over Mon?

We need $-\,;^{\mathsf{Set}_*}\,-$ in Mon, so $\mathsf{Set}_*(A,B) \times \mathsf{Set}_*(A,B)$ needs to be a monoid (which it is by pointwise η and μ). It also needs to be a monoid-homomorphism, which implies:

$$(f\ \mu\ f')\,;(g\ \mu\ g') \tag{28}$$
$$= -\ ;\ -\ (\mu((f,g),(f',g'))) \tag{29}$$
$$= \mu(-\,;\,-\,((f,g),(f',g'))) \tag{30}$$
$$= (f\,;g)\ \mu\ (f'\,;g') \tag{31}$$
$$\neq (f\,;g)\ \mu\ (f\,;g')\ \mu\ (f'\,;g)\ \mu\ (f'\,;g') \tag{32}$$

It works out if we use $\otimes$ instead of $\times$, but $\otimes$ is not associative. `https://math.stackexchange.com/questions/205871/tensor-product-of-monoids-and-arnoredirect=1&lq=1`

Enrichment over multicategory?

I.e. now $-\,;-\colon [A \to B], [B \to C] \to [A \to C]$.

3.2 Pure vs. Impure Pattern Matching

Let there be a tensor product $(\oplus, a, r, l, \emptyset)$ that distributes over the standard tensor product via $d\colon A \otimes (B \oplus C) \cong (A \otimes B) \oplus (A \otimes C)$. Then we define 'pure pattern-matching' as follows:

$$\frac{\begin{array}{c}\Gamma, \Delta' \,;\, \Delta \vdash_\dagger e_0 : j\ A_1 \oplus A_2 \\ \Gamma \,;\, \Delta', v_1 \colon A_1 \vdash_\dagger e_1 : j\ B_1 \\ \Gamma \,;\, \Delta', v_2 \colon A_2 \vdash_\dagger e_2 : j\ B_2\end{array}}{\Gamma \,;\, \Delta, \Delta' \vdash_\dagger \mathtt{case}\, e_0\, \mathtt{of}\, \iota_1\, v_1 \;\Rightarrow\; \iota_1\, e_1;\, \iota_2\, v_2 \;\Rightarrow\; \iota_2\, e_2;\colon j\ B_1 \oplus B_2} \tag{33}$$

$$[\![\mathtt{case}\, e_0\, \mathtt{of}\, \iota_1\, v_1 \;\Rightarrow\; \iota_1\, e_1;\, \iota_2\, v_2 \;\Rightarrow\; \iota_2\, e_2;]\!]_\dagger = \lambda\gamma.\mathrm{j}\forall\lambda\delta'.\, [\![e_0]\!]_\dagger\ (\gamma, \delta')\,;\, d\,; \tag{34}$$
$$((([\![e_1]\!]_\dagger\ \gamma) \oplus ([\![e_2]\!]_\dagger\ \gamma))\,;\, d^{-1}$$
$$[\![\mathtt{case}\, e_0\, \mathtt{of};]\!]_\dagger = [\![e_0]\!]_\dagger\ ;\mathrm{id}_\emptyset \tag{35}$$

Let the hom-sets be equipped with a monoid structure $(+, 0)$. Then we define 'impure pattern-matching' as follows:

$$\frac{\begin{array}{c}\Gamma, \Delta' \,;\, \Delta \vdash_\dagger e_0 : j\ A \\ \Gamma \,;\, \Delta', v_1 \colon A \vdash_\dagger e_1 : j\ B \\ \Gamma \,;\, \Delta', v_2 \colon A \vdash_\dagger e_2 : j\ B\end{array}}{\Gamma \,;\, \Delta, \Delta' \vdash_\dagger \mathtt{case}\, e_0\, \mathtt{of}\, v_1 \;\Rightarrow\; e_1;\, v_2 \;\Rightarrow\; e_2;\colon j\ B} \tag{36}$$

$$[\![\mathtt{case}\, e_0\, \mathtt{of}\, p_1 \;\Rightarrow\; e_1;\, p_2 \;\Rightarrow\; e_2;]\!]_\dagger = \lambda\gamma.\mathrm{j}\forall\lambda\delta'.\, [\![e_0]\!]_\dagger\ (\gamma, \delta')\,; \tag{37}$$
$$(([\![e_1]\!]_\dagger\ \gamma) + ([\![e_2]\!]_\dagger\ \gamma))$$
$$[\![\mathtt{case}\, e_0\, \mathtt{of};]\!]_\dagger = [\![e_0]\!]_\dagger\ ;0 \tag{38}$$

Lemma 1. *Now let there be the following morphisms natural in A and B: $\iota_1\colon A \to A \oplus B$, $\iota_2\colon B \to A \oplus B$, $\pi_1\colon A \oplus B \to A$, $\pi_2\colon A \oplus B \to B$. Then the following two properties are equivalent.*

(i) $\pi_1\,;\iota_1 + \pi_2\,;\iota_2 = \mathrm{id} \oplus \mathrm{id}$.
(ii) $f \oplus g = (\pi_1\,;f\,;\iota_1) + (\pi_2\,;g\,;\iota_2)$ *for all* $f, g\colon A \to B$.

Proof. By assumption $\iota_1\colon \Pi_1 \to \oplus$ and $\pi_1\colon \oplus \to \Pi_1$ are natural transformations with $\Pi_1\colon \mathcal{C}\times\mathcal{D} \to \mathcal{C}$ the projection functor of the cartesian product of categories. So the following diagram commutes (and analogously for π_2 and ι_2).

$$\begin{array}{ccccc} A \oplus B & \xrightarrow{\pi_1} & A & \xrightarrow{\iota_1} & A \oplus B \\ \downarrow{\scriptstyle f \oplus g} & & \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle f \oplus g} \\ A' \oplus B' & \xrightarrow{\pi_1} & A' & \xrightarrow{\iota_1} & A' \oplus B' \end{array}$$

(1.(i) $\implies$ 1.(ii))

$$\begin{aligned} f \oplus g &= (\pi_1 \,;\, \iota_1 + \pi_2 \,;\, \iota_2) \,;\, (f \oplus g) && \text{(1.(i))} && (39) \\ &= \pi_1 \,;\, \iota_1 \,;\, (f \oplus g) + \pi_2 \,;\, \iota_2 \,;\, (f \oplus g) && \text{(naturality of monoid)} && (40) \\ &= (\pi_1 \,;\, f \,;\, \iota_1) + (\pi_2 \,;\, g \,;\, \iota_2) && \text{(naturality of } \iota_1, \pi_1, \iota_2, \pi_2) && (41) \end{aligned}$$

(1.(ii) $\implies$ 1.(i)) Immediately with $f = g = \mathrm{id}$.

With 1.(*ii*) the pure pattern-matching is exactly a special case of the impure pattern-matching if $\iota_i^\dagger = \pi_i$.

3.3 Asymmetric First-Match Policy

Let $\mathsf{Set}_{0\infty}$ be the category of sets with two chosen elements 0 and ∞ and functions that preserve 0 and ∞. Note that this is isomorphic to the Kleisli-category over the monad $X + 2$. Define a monoid on the hom-sets as follows:

$$(f + g)\, a = \begin{cases} f\, a \text{ if } g\, a = 0 \\ g\, a \text{ if } f\, a = 0 \\ \infty \text{ otherwise} \end{cases} \tag{42}$$

$$\dot{0}\, a = 0 \text{ if } a \neq \infty \text{ else } \infty \tag{43}$$

$$\dot{\infty}\, a = \infty \text{ if } a \neq 0 \text{ else } 0 \tag{44}$$

Lemma 2. *(i)* $(f + g) + h = f + (g + h)$
(ii) $f + \dot{0} = f = \dot{0} + f$
(iii) $f + \dot{\infty} = \dot{\infty} = \dot{\infty} + f$
(iv) $f + g = g + f$
(v) $\dot{0} \,;\, f = \dot{0}$
(vi) $\dot{\infty} \,;\, f = \dot{\infty}$
(vii) $f \,;\, \dot{0} = \lambda a.\infty$ *if* $f\, a = \infty$ *else* 0
(viii) $f \,;\, \dot{\infty} = \lambda a.0$ *if* $f\, a = 0$ *else* ∞
(ix) $f + f = f \,;\, \dot{\infty}$
(x) $f \,;\, (g + h) = f \,;\, g + f \,;\, h$ *(NEW)*
(xi) $(f + g) \,;\, h \geq f \,;\, h + g \,;\, h$ *(NEW)*
(xii) $(f_{0\infty} + g_{0\infty}) \,;\, h_{0\infty} = f_{0\infty} \,;\, h_{0\infty} + g_{0\infty} \,;\, h_{0\infty}$ *with* $f, g \in \mathsf{Set}(A, B)$ *(NEW)*

This monoid extends to the janus category as usual: $(f\nearrow, f\searrow) + (g\nearrow, g\searrow) = (f\nearrow + g\nearrow, f\searrow + g\searrow)$.

Now we define the notion of partial isomorphisms in $\mathsf{JSet}_{0\infty}$.

Definition 1. *A janus* $f\colon A \leftrightarrows B$ *is called* partial semi-isomorphism *if*

$$\forall a \in A \mid \begin{array}{c} f\searrow(f\nearrow a) \in \{0, a, \infty\} \wedge \\ f\searrow(f\nearrow a) = 0 \end{array} \implies f\nearrow a = 0 \tag{45}$$

equivalently

$$\forall a \in A \mid f\nearrow a \neq 0 \implies f\searrow(f\nearrow a) \in \{a, \infty\} \tag{46}$$

A janus is called partial isomorphism *if both it and its reverse are partial semi-isomorphisms.*

Lemma 3. *Let* f, g *be partial semi-isomorphisms in* $\mathsf{JSet}_{0\infty}$*. Then* $f + g$ *is a partial semi-isomorphism.*

Proof. Let $(f+g)\searrow((f+g)\nearrow a) = a'$. Then we have five possibilities.

1. $a' = \infty \in \{0, a, \infty\}$.
2. $a' = f\searrow(f\nearrow a) \in \{0, a, \infty\}$.
3. $a' = g\searrow(g\nearrow a) \in \{0, a, \infty\}$.
4. $a' = g\searrow(f\nearrow a)$, which implies $f\searrow(f\nearrow a) = 0$, so $f\nearrow a = 0$, so $a' = 0 \in \{0, a, \infty\}$.
5. $a' = f\searrow(g\nearrow a)$ analogous to above.

Now assume $a' = 0$. So $f\searrow((f+g)\nearrow a) = 0$ and $g\searrow((f+g)\nearrow a) = 0$. Then we have two cases.

1. $g\nearrow a = 0$. Then $(f+g)\nearrow a = f\nearrow a$. But because $f\searrow((f+g)\nearrow a) = 0$ we have $f\nearrow a = 0$.
2. $f\nearrow a = 0$. Analogously.

So $(f+g)\nearrow a = f\nearrow a = g\nearrow a = 0$.
OR
Let $(f+g)\nearrow a \neq 0$.

1. $f\nearrow a = 0 \wedge g\nearrow a = 0$. Then $(f+g)\nearrow a = 0$ contradicting assumption.
2. $f\nearrow a \neq 0 \wedge g\nearrow a \neq 0$. Then $(f+g)\nearrow a = \infty$, so $(f+g)\searrow((f+g)\nearrow a) = \infty \in \{a, \infty\}$.
3. $f\nearrow a \neq 0 \wedge g\nearrow a = 0$. Then $(f+g)\nearrow a = f\nearrow a$ and $f\searrow(f\nearrow a) \in \{a, \infty\}$. So $(f+g)\searrow((f+g)\nearrow a) \in \{a, \infty\}$.
4. $g\nearrow a \neq 0 \wedge f\nearrow a = 0$. Analogously.

Corollary 1. *Let* f, g *be partial isomorphisms in* $\mathsf{JSet}_{0\infty}$*. Then* $f+g$ *is a partial isomorphism.*

Question: Programming language with commutative case-expression/statement

Now define for every function $f\colon A \to B$ functions $\neg f, {\sim} f\colon A \to A$ as

$$\neg f\, a = \begin{cases} \infty \text{ if } f\, a = \infty \\ a \text{ if } f\, a = 0 \\ 0 \text{ otherwise} \end{cases} \tag{47}$$

$$ {\sim} f\, a = \begin{cases} a \text{ if } f\, a = 0 \\ \infty \text{ otherwise} \end{cases} \tag{48}$$

and define for a janus $f\colon A \leftrightarrows B$ januses $\neg f\colon A \leftrightarrows A = (\neg(f\nearrow), {\sim}(f\nearrow))$ and ${\sim} f\colon A \leftrightarrows A = ({\sim}(f\nearrow), \neg(f\nearrow))$ which are partial isomorphisms.

Lemma 4. *(i)* $\neg(f + \neg f\,; g) = \neg f\,; \neg g$
(ii) $\neg(f + \neg f\,; g\,; {\sim} f')\,; h\,; {\sim}(f' + \neg f\,; g'\,; {\sim} f') = \neg f\,; \neg g\,; h\,; {\sim} g'\,; {\sim} f'$

Now define two new monoids on the hom-sets:

$$(f\ '{+}\ g) = f + \neg f\,; g \tag{49}$$

$$(f\ '{+}'\ g) = f + \neg f\,; g\,; {\sim}(f^\dagger) \tag{50}$$

$'+'$ will never add ∞ and is basically the symmetric first-match policy but is defined only for $\mathsf{JSet}_{0\infty}$. - $'+$ is defined for both $\mathsf{Set}_{0\infty}$ and $\mathsf{JSet}_{0\infty}$ and for $\mathsf{Set}_{0\infty}$ $'+$ is basically the same as for Set_*. The downside is that $(f\ '{+}\ g)^\dagger \neq f^\dagger\ '{+}\ g^\dagger$, but another monoid $(f\ {+}'\ g) = f + g\,; \neg(f^\dagger)$.

3.4 8.9. – Generalization

Definition 2. *A* case category* *is a category where*

- *each object A is a set equipped with a preorder $\leq_A$ and a monoid $(+_A, 0_A)$, and*
- *morphisms are monotone, superadditive functions, i.e., for all $f\colon A \to B, a, a' \in A$ we have $a \leq a' \implies f\, a \leq f\, a'$ and $f\, a +_B f\, a' \leq f\,(a +_A a')$.*

A case category is called* increasing *if $a + a'' \leq a + a' + a''$ for all $a, a', a'' \in A$ for any object A.*

A case category is called* commutative *if $a + a' = a' + a$ for all $a, a' \in A$ for any object A.*

Lemma 5. *In an increasing case category**

- *0 is always the smallest element and*
- *$f\, a \leq f\, 0 + f\, a + f\, 0 \leq f\, a$.*

Proof. $0 = 0 + 0 \leq 0 + a + 0 = a$.

From increasingness and superadditivity we have

$$f\, a \leq f\, 0 + f\, a + f\, 0 \leq f\,(0 + a + 0) = f\, a\,.$$

Definition 3. *A pair of morphisms* $(f\nearrow\colon A \to B, f\searrow\colon B \to A)$ *in a case category* is called a* case-partial semi-isomorphism* *if*

$$\forall a \mid f\nearrow a \neq 0 \implies a \leq f\searrow(f\nearrow a)\ .$$

Lemma 6. *Case-partial semi-isomorphisms* are closed under composition.*

Proof. Let $f\colon A \to B$ and $g\colon B \to C$ be case-partial semi-isomorphisms*.

$$\begin{aligned}
& g\nearrow(f\nearrow a) \neq 0 && (51)\\
\implies & f\nearrow a \leq g\searrow(g\nearrow(f\nearrow a)) && (52)\\
\implies & f\searrow(f\nearrow a) \leq f\searrow(g\searrow(g\nearrow(f\nearrow a))) && (53)\\
\implies & a \leq f\searrow(g\searrow(g\nearrow(f\nearrow a))) && (54)
\end{aligned}$$

Lemma 7. *If* f *and* g *are case-partial semi-isomorphisms* in an increasing case category* then so is* $f \dotplus g$.

Proof.

$$\begin{aligned}
& (f\nearrow \dotplus g\nearrow)\, a \neq 0 && (55)\\
\implies & \{\text{definition}\} && (56)\\
& f\nearrow a + g\nearrow a \neq 0 && (57)\\
\implies & \{b = 0 \wedge b' = 0 \implies b + b' = 0\} && (58)\\
& f\nearrow a \neq 0 \vee g\nearrow a \neq 0 && (59)\\
\implies & \{f \text{ and } g \text{ are case-partial semi-isomorphisms }\} && (60)\\
& a \leq f\searrow(f\nearrow a) \vee a \leq g\searrow(g\nearrow a) && (61)\\
\implies & \{\text{increasing}\} && (62)\\
& a \leq f\searrow(f\nearrow a) + f\searrow(g\nearrow a) \vee a \leq g\searrow(f\nearrow a) + g\searrow(g\nearrow a) && (63)\\
\implies & \{f\searrow \text{ and } g\searrow \text{ are superadditive}\} && (64)\\
& a \leq f\searrow(f\nearrow a + g\nearrow a) \vee a \leq g\searrow(f\nearrow a + g\nearrow a) && (65)\\
\implies & \{\text{increasing}\} && (66)\\
& a \leq f\searrow(f\nearrow a + g\nearrow a) + g\searrow(f\nearrow a + g\nearrow a) \vee && (67)\\
& a \leq f\searrow(f\nearrow a + g\nearrow a) + g\searrow(f\nearrow a + g\nearrow a) && (68)\\
\implies & \{\text{definition}\} && (69)\\
& a \leq (f \dotplus g)\searrow((f \dotplus g)\nearrow a) && (70)
\end{aligned}$$

Definition 4. Set_* *is the category of pointed sets* $A_* = (A, *_A)$ *and functions preserving the points.*

$\mathsf{Set}_{0\infty}$ *is the category of sets with two points* $A_{0\infty} = (A, 0_A, \infty_A)$ *and functions preserving both points.*

Set^* *is the category of lists* A^* *and functions that commute with list concatenation, i.e.,* $f\,(a + a') = f\,a + f\,a'$.

$\mathcal{P}(\mathsf{Set})$ *is the category of powersets* $\mathcal{P}$ *and continuous functions, i.e.,* $f\,(a \cup a') = f\,a \cup f\,a'$.

$\mathcal{P}_{<\infty}(\mathsf{Set})$ *is the full subcategory of* $\mathcal{P}(\mathsf{Set})$ *containing only the finite sets.*

(?) $\mathcal{D}(\mathsf{Set})$ *is the category of finite probability distributions and continuous functions, i.e.,* $\mathcal{D}(A) = \{d\colon A \to \mathbb{R}_{\geq 0} \mid \{a\colon A \mid d\,(a) \neq 0\}$ *is finite and non-empty*$\}$ *and* $f : \mathcal{D}(A) \to \mathcal{D}(B)$ *such that* $f\,(p \dotplus p') = (f\,p) \dotplus (f\,p')$.

(?) $\mathcal{Q}(\mathsf{Set})$ *is the category of finite quantum distributions (???) and continuous functions, i.e.,* $\mathcal{Q}(A) = \{f\colon A \to \mathbb{C} \mid \{a\colon A \mid f\,(a) \neq 0\}$ *is finite and non-empt and* $f : \mathcal{Q}(A) \to \mathcal{Q}(B)$ *such that* $f\,(p \dotplus p') = (f\,p) \dotplus (f\,p')$.

Lemma 8. Set_*, $\mathsf{Set}_{0\infty}$, Set^*, $\mathcal{P}(\mathsf{Set})$ *and* $\mathcal{P}_{<\infty}(\mathsf{Set})$ *are case-categories* with*

- *for* Set_**:*

$$\leq_{A_*} = \{(a, *) \mid a \in A_*\} \tag{71}$$

$$a +_{A_*} a' = \begin{cases} a \text{ if } a \neq * \\ a' \text{ otherwise} \end{cases} \tag{72}$$

- *for* $\mathsf{Set}_{0\infty}$*:*

$$\leq_{A_{0\infty}} = \bigcup \{\{(0, a), (a, \infty)\} \mid a \in A_{0\infty}\} \tag{73}$$

$$a +_{A_{0\infty}} a' = \begin{cases} a \text{ if } a' = 0 \\ a' \text{ if } a = 0 \\ \infty \text{ otherwise} \end{cases} \tag{74}$$

- *for* Set^**:*

$$\leq_{A^*} = \{((a_i)_{i \in n\downarrow}, (a'_i)_{i \in n'\downarrow}) \mid \exists I \subseteq n'\downarrow \mid (a_i)_{i \in n\downarrow} = (a'_i)_{i \in I}\} \tag{75}$$

$$\text{where } n\downarrow = \{1, \ldots, n\} \tag{76}$$

$$+_{A^*} \text{ is list concatenation} \tag{77}$$

and

- *for* $\mathcal{P}(\mathsf{Set})$ *and* $\mathcal{P}_{<\infty}(\mathsf{Set})$*:* $\leq = \subseteq$, $+ = \cup$

Proof. Set_*, $\mathsf{Set}_{0\infty}$, Set^*, $\mathcal{P}(\mathsf{Set})$ and $\mathcal{P}_{<\infty}(\mathsf{Set})$ are categories.

Let $f\colon A \to B$ in Set_*, $a \in A$, $b \in B$, $\alpha, \alpha' \in A_*$, $\beta' \in B_*$. We have the following cases.

α	α'	$f\,\alpha$	$f\,\alpha'$	$\alpha + \alpha'$	$f\,\alpha + f\,\alpha'$		$f\,(\alpha + \alpha')$
a	α'	b	β'	a	b	$\leq$	b
a	α'	$*$	β'	a	β'	$\leq$	$*$
$*$	α'	$*$	β'	a'	β'	$\leq$	β'

Let $f\colon A \to B$ in $\mathsf{Set}_{0\infty}$, $a, a' \in A$, $\alpha' \in A_{0\infty}$, $\beta, \beta', \beta'' \in B_{0\infty}$. We have the following cases skipping symmetric ones, because the monoid is commutative.

α	α'	$f\,\alpha$	$f\,\alpha'$	$\alpha + \alpha'$	$f\,\alpha + f\,\alpha'$		$f\,(\alpha + \alpha')$
a	a'	β	β'	∞	β''	$\leq$	∞
∞	α'	∞	β'	∞	∞	$\leq$	∞
0	α'	0	β'	0	β'	$\leq$	β'

For Set^*, $\mathcal{P}(\mathsf{Set})$ and $\mathcal{P}_{<\infty}(\mathsf{Set})$ we have $f\,(a{+}a') = (f\,a){+}(f\,a')$ by definition.

Lemma 9. *$\mathsf{Set}_{0\infty}$, $\mathcal{P}(\mathsf{Set})$ and $\mathcal{P}_{<\infty}(\mathsf{Set})$ are increasing, commutative case-categories, Set^* is increasing but not commutative, and Set_* is neither.*

Proof. For Set_* we have $a + a' = a \neq a' = a' + a$ and $a' \not\leq a = a + a'$ for $a \neq a' \in A$.

We show that $\mathsf{Set}_{0\infty}$ is increasing and commutative by listing all cases. Let $a, a' \in A$ and $\alpha, \alpha' \in A_{0\infty}$.

α'	α		$\alpha + \alpha'$		$\alpha' + \alpha$
a'	a	$\leq$	∞	$=$	∞
0	a	$\leq$	a	$=$	a
∞	a	$\leq$	∞	$=$	∞
a'	0	$\leq$	a'	$=$	a'
0	0	$\leq$	0	$=$	0
∞	0	$\leq$	∞	$=$	∞
a'	∞	$\leq$	∞	$=$	∞
0	∞	$\leq$	∞	$=$	∞
∞	∞	$\leq$	∞	$=$	∞

Because of commutativity we have $\alpha + \alpha'' \leq \alpha + \alpha'' + \alpha' = \alpha + \alpha' + \alpha''$.

For Set^* it is easy to see that concatenation is not commutative but increasing.

For $\mathcal{P}(\mathsf{Set})$ and $\mathcal{P}_{<\infty}(\mathsf{Set})$ we have $a \cup a' = a' \cup a$, $a \subseteq a \cup a'$ for $a, a' \in \mathcal{P}(A)$, because powersets are lattices.

Lemma 10. *Pointwise addition $(\dot{+}, \dot{0})$ and ordering $(\dot{\leq})$ induces a partial order and a monoid on the morphisms of a case category*. If the category is commutative rsp. increasing then so is the induced morphism structure.*

Proof. $f \mathrel{\dot{\leq}} f \iff \forall a \mid f\,a \leq f\,a \iff \top$

$f \mathrel{\dot{\leq}} g \iff \forall a \mid f\,a \leq g\,a \implies \forall a \mid g\,a \not\leq g\,a \implies f \mathrel{\dot{\not\leq}} g$

$(f \mathbin{\dot{+}} g) \mathbin{\dot{+}} h = \lambda a.(f\,a + g\,a) + h\,a = \lambda a.f\,a + (g\,a + h\,a) = f \mathbin{\dot{+}} (g \mathbin{\dot{+}} h)$

$f \mathrel{\dot{\leq}} g \mathrel{\dot{\leq}} h \iff \forall a \mid f\,a \leq g\,a \leq h\,a \implies \forall a \mid f\,a \leq h\,a \iff f \mathrel{\dot{\leq}} h$

$f \mathbin{\dot{+}} \dot{0} = \lambda a.f\,a + 0 = f$

$\dot{0} \mathbin{\dot{+}} f = \lambda a.0 + f\,a = f$

$f \mathbin{\dot{+}} g = \lambda a.f\,a + g\,a = \lambda a.g\,a + f\,a = g \mathbin{\dot{+}} f$

$f = \lambda a.f\,a \mathrel{\dot{\leq}} \lambda a.f\,a + g\,a = f \mathbin{\dot{+}} g$

Lemma 11. *For a commutative case category* we have*

$$(f \mathbin{\dot{+}} g)\,\alpha + (f \mathbin{\dot{+}} g)\,\alpha' \leq (f \mathbin{\dot{+}} g)\,(\alpha + \alpha') \tag{78}$$

for all $f, g \colon A \to B$ in $\mathcal{C}$ and $\alpha, \alpha' \in A$.

Proof.

$$
\begin{aligned}
&(f \dot{+} g)\,\alpha + (f \dot{+} g)\,\alpha' && (79)\\
=&f\,\alpha + g\,\alpha + f\,\alpha' + g\,\alpha' && (80)\\
=&f\,\alpha + f\,\alpha' + g\,\alpha + g\,\alpha' && (81)\\
\leq&f\,(\alpha + \alpha') + g\,(\alpha + \alpha') && (82)\\
=&(f \dot{+} g)\,(\alpha + \alpha') && (83)
\end{aligned}
$$

Lemma 12. *$\dot{+}$ is closed in Set_*, $\mathsf{Set}_{0\infty}$, $\mathcal{P}(\mathsf{Set})$ and $\mathcal{P}_{<\infty}(\mathsf{Set})$.*

Proof. Let $f, g\colon A_* \to B_*$ in Set_*, $a, a' \in A$, $b, b'' \in B$, $\alpha, \alpha' \in A_*$, $\beta, \beta', \beta'', \beta''' \in B_*$.

$$(f \dot{+} g)\,* = f\,* + g\,* = * + * = * \tag{84}$$

α	α'	$f\,\alpha$	$f\,\alpha'$	$g\,\alpha$	$g\,\alpha'$	$(f \dot{+} g)\,\alpha + (f \dot{+} g)\,\alpha'$		$(f \dot{+} g)\,(\alpha + \alpha')$
a	α'	b	β'	β''	β'''	b	$\leq$	b
a	a'	$*$	β'	$*$	β'''	β'	$\leq$	$*$
a	a'	$*$	β'	b''	β'''	b''	$\leq$	b''
a	$*$	β	$*$	β''	$*$	$\beta + \beta''$	$\leq$	$\beta + \beta''$
$*$	α'	$*$	β'	$*$	β'''	$\beta' + \beta'''$	$\leq$	$\beta' + \beta'''$

Let $f, g\colon A_{0\infty} \to B_{0\infty}$ in $\mathsf{Set}_{0\infty}$, $a, a' \in A$, $\alpha, \alpha' \in A_{0\infty}$, $\beta, \beta', \beta'', \beta''' \in B_{0\infty}$.

$$
\begin{aligned}
(f \dot{+} g)\,0 &= f\,0 + g\,0 = 0 + 0 = 0 && (85)\\
(f \dot{+} g)\,\infty &= f\,\infty + g\,\infty = \infty + \infty = \infty && (86)
\end{aligned}
$$

Lemma 13. *$\dot{+}$ is not closed in Set^*, but the following is closed and a monoid with $\dot{0}$ as unit.*

$$
\begin{aligned}
(f +^* g) &= \lambda(a_i)_{i\in n\downarrow}.\mathsf{join}_B\,(f\,(a_i)_1 + g\,(a_i)_1)_{i\in n\downarrow} && (87)\\
&= \lambda(a_i)_{i\in n\downarrow}.f\,(a_1)_1 + g\,(a_1)_1 + \cdots + f\,(a_n)_1 + g\,(a_n)_1 && (88)
\end{aligned}
$$

where $f, g\colon A^* \to B^*$, $(a)_1$ is a singleton list and $\mathsf{join}_B\colon B^{**} \to B^*$ is the function that flattens a list of lists, i.e., the multiplication operation of the list monad.

Proof.

$$
\begin{aligned}
&(f \dot{+} g)\,a + (f \dot{+} g)\,a' && (89)\\
=&f\,a + f\,a' + g\,a + g\,a' && (90)\\
\neq&f\,a + g\,a + f\,a' + g\,a' && (91)\\
=&(f \dot{+} g)\,(a + a') && (92)
\end{aligned}
$$

$$
\begin{aligned}
&(f +^* g)\,(a_i)_{i\in n\downarrow} + (f +^* g)\,(a'_i)_{i\in n'\downarrow} && (93)\\
=&\mathsf{join}\,((f\,a_i, g\,a_i))_{i\in n\downarrow} + \mathsf{join}\,((f\,a'_i, g\,a'_i))_{i\in n'\downarrow} && (94)\\
=&\mathsf{join}\,(((f\,a_i, g\,a_i))_{i\in n\downarrow} + ((f\,a'_i, g\,a'_i))_{i\in n'\downarrow}) && (95)\\
=&(f +^* g)\,((a_i)_{i\in n\downarrow} + (a'_i)_{i\in n'\downarrow}) && (96)
\end{aligned}
$$

$$
\begin{aligned}
&((f +^* g) +^* h)\,(a_i)_{i\in n\downarrow} && (97)\\
=&\mathsf{join}\,((f +^* g)\,(a_i)_1 + h\,(a_i)_1)_{i\in n\downarrow} && (98)\\
=&\mathsf{join}\,(f\,(a_i)_1 + g\,(a_i)_1 + h\,(a_i)_1)_{i\in n\downarrow} && (99)\\
=&\mathsf{join}\,(f\,(a_i)_1 + (g +^* h)\,(a_i)_1)_{i\in n\downarrow} && (100)\\
&(f +^* (g +^* h))\,(a_i)_{i\in n\downarrow} && (101)
\end{aligned}
$$

Lemma 14. *Case-partial semi-isomorphisms* are closed under* $+^*$.

Proof. We first show that list-homomorphisms are increasing pointwise, i.e., $f +^* h \mathrel{\dot\leq} f +^* g +^* h$.

$$
\begin{aligned}
&(f +^* h)\,(a_i)_{i\in n\downarrow} && (102)\\
=&\{\text{definition}\} && (103)\\
&f\,(a_1)_1 + h\,(a_1)_1 + \cdots + f\,(a_n)_1 + h\,(a_n)_1 && (104)\\
=&\{\text{lists are increasing}\} && (105)\\
&f\,(a_1)_1 + g\,(a_1) + h\,(a_1)_1 + \cdots + f\,(a_n)_1 + g\,(a_n)_1 + h\,(a_n)_1 && (106)\\
=&\{\text{definition}\} && (107)\\
&(f +^* g +^* h)\,(a_i)_{i\in n\downarrow} && (108)
\end{aligned}
$$

Then we show $g \mathrel{\dot\leq} g' \implies f\,;g\,;h \mathrel{\dot\leq} f\,;g'\,;h$.

$$
\begin{aligned}
&g \mathrel{\dot\leq} g' && (109)\\
\iff&\{\text{definition}\} && (110)\\
&g\,b \leq g'\,b && (111)\\
\implies&\{h \text{ is monotone}\} && (112)\\
&h\,(g\,(f\,a)) \leq h\,(g'\,(f\,a)) && (113)\\
\iff&\{\text{definition}\} && (114)\\
&f\,;g\,;h \mathrel{\dot\leq} f\,;g'\,;h && (115)
\end{aligned}
$$

Now, because $f\,a = 0 \wedge g\,a = 0 \implies (f +^* g)\,0 = 0$ we have $(f+g)\,a \neq 0 \implies f\,a \neq 0 \vee g\,a \neq 0$. Assuming $f\,a \neq 0$ we get

$$
\begin{aligned}
&a && (116)\\
\leq&(f{\nearrow}\,;f{\searrow})\,a && (117)\\
\leq&(f{\nearrow}\,;(f{\searrow} +^* g{\searrow}))\,a && (118)\\
\leq&((f{\nearrow} +^* g{\nearrow})\,;(f{\searrow} +^* g{\searrow}))\,a && (119)\\
& && (120)
\end{aligned}
$$

And analogously for $g\,a \neq 0$.

3.5 8.9. – Generalization Categorical (WIP)

Definition 5. *A* case category *is a 2-category where*

- *the hom-categories are (pseudo-)monoidal via* $(+, 0)$*, and*
- *for every morphism* $f\colon A \to B$ *there is a morphism* $\overline{f} : A \to A$*, where*
- *we write* $f \leq g$ *for* $\exists \alpha \in \mathcal{C}(A, B)(f, g)$*, and*
- *we write* $f \simeq g$ *for* $f \leq g \wedge g \leq f$

subject to the following conditions

$$g \leq g' \implies f\,;g\,;h \leq f\,;g'\,;h \tag{121}$$
$$f \leq f' \wedge g \leq g' \implies f + g \leq f' + g' \tag{122}$$
$$f\,;g\,;h + f\,;g'\,;h \leq f\,;(g + g')\,;h \tag{123}$$
$$f \leq g \implies \overline{f} \leq \overline{g} \tag{124}$$
$$\overline{f + g} \leq \overline{f} + \overline{g} \tag{125}$$
$$f \simeq \overline{f}\,;f \tag{126}$$
$$\overline{f}\,;\overline{g} \simeq \overline{g}\,;\overline{f} \tag{127}$$
$$\overline{f}\,;\overline{g} \geq \overline{\overline{f}\,;g} \tag{128}$$
$$f\,;\overline{g} \leq \overline{f\,;g}\,;f \tag{129}$$

A case category is called increasing *if* $f + h \leq f + g + h$.
A case category is called commutative *if* $f + g \simeq g + f$.

Lemma 15. $\overline{f} \leq f\,;g \implies \overline{f\,;\overline{h}} \leq f\,;\overline{h}\,;g$

Proof.

$$\overline{f\,;\overline{h}} \tag{130}$$
$$\simeq \overline{\overline{f}\,;f\,;\overline{h}} \tag{131}$$
$$\leq \overline{f}\,;\overline{f\,;\overline{h}} \tag{132}$$
$$\simeq \overline{f\,;\overline{h}}\,;\overline{f} \tag{133}$$
$$\leq \overline{f\,;\overline{h}}\,;f\,;g \tag{134}$$
$$\leq f\,;\overline{\overline{h}}\,;g \tag{135}$$
$$= f\,;\overline{\overline{h}\,;\mathrm{id}}\,;g \tag{136}$$
$$\leq f\,;\overline{h}\,;\overline{\mathrm{id}}\,;g \tag{137}$$
$$= f\,;\overline{h}\,;\overline{\mathrm{id}}\,;\mathrm{id}\,;g \tag{138}$$
$$\simeq f\,;\overline{h}\,;\mathrm{id}\,;g \tag{139}$$
$$= f\,;\overline{h}\,;g \tag{140}$$
$$\tag{141}$$

Definition 6. *A case-partial semi-isomorphism is a pair of morphisms* $f\nearrow\colon A \to B$ *and* $f\searrow\colon B \to A$ *in a case category such that*

$$\overline{f\nearrow} \leq f\nearrow\,;f\searrow \tag{142}$$

Lemma 16. *Case-partial semi-isomorphisms are closed under composition.*

Case-partial semi-isomorphisms are closed under $+$ in increasing case-categories.

Proof. Let $f\nearrow, g\nearrow \colon A \to B$ and $f\searrow, g\searrow \colon B \to A$ and $(f\nearrow, f\searrow)$ and $(g\nearrow, g\searrow)$ be case-partial semi-isomorphisms.

$$\begin{aligned}
&\overline{f\nearrow ; g\nearrow} && (143)\\
\leq &\{\text{Equation (128)}\} && (144)\\
&\overline{f\nearrow ; \overline{g\nearrow}} && (145)\\
\leq &\{\text{assumption, Lemma 15}\} && (146)\\
&f\nearrow ; \overline{g\nearrow} ; f\searrow && (147)\\
\leq &\{\text{assumption, Equation (121)}\} && (148)\\
&f\nearrow ; g\nearrow ; g\searrow ; f\searrow && (149)
\end{aligned}$$

So $(f\nearrow ; g\nearrow, g\searrow ; f\nearrow)$ is a case-partial semi-isomorphism.

$$\begin{aligned}
&\overline{f\nearrow + g\nearrow} && (150)\\
\leq &\{\text{Equation (125)}\} && (151)\\
&\overline{f\nearrow} + \overline{g\nearrow} && (152)\\
\leq &\{\text{assumption, Equation (122)}\} && (153)\\
&f\nearrow ; f\searrow + g\nearrow ; g\searrow && (154)\\
\leq &\{\text{increasing}\} && (155)\\
&f\nearrow ; f\searrow + f\nearrow ; g\searrow + g\nearrow ; f\searrow + g\nearrow ; g\searrow && (156)\\
\leq &\{\text{Equation (123), Equation (122)}\} && (157)\\
&f\nearrow ; (f\searrow + g\searrow) + g\nearrow ; (f\searrow + g\searrow) && (158)\\
\leq &\{\text{Equation (123), Equation (122)}\} && (159)\\
&(f\nearrow + g\nearrow) ; (f\searrow + g\searrow) && (160)
\end{aligned}$$

So $(f\nearrow + g\nearrow, f\searrow + g\searrow)$ is a case-partial semi-isomorphism.

Definition 7. *A dagger case category is a case category that is also a dagger category such that $(f+g)^\dagger = f^\dagger + g^\dagger$.*

A $\mathrm{j\mathbb{V}}$*-situation is when a symmetric monoidal dagger case category $\mathcal{D}$ is enriched over a cartesian closed case category $\mathcal{C}$ that has a natural transformation $\mathrm{j\mathbb{V}}^{\mathcal{D}} : [C \to [A \leftrightarrows B]] \to [A \otimes C \leftrightarrows B \otimes C]$, and there is a symmetric monoidal identity-on-objects functor $-\nearrow \colon \mathcal{D} \to \mathcal{C}$ such that*

$$\begin{aligned}
(\mathrm{j\mathbb{V}}^{\mathcal{D}}\, f)\nearrow = \mathrm{j\mathbb{V}}^{\mathcal{C}}\, (f\, ;\, -\nearrow) = \lambda f.\lambda(a,c).((f\, c)\nearrow a, c) && (161)\\
(f+g)\nearrow = f\nearrow + g\nearrow && (162)\\
f \leq g \implies f\nearrow \leq g\nearrow && (163)
\end{aligned}$$

Lemma 17. *The janus category $\mathcal{JC}$ of a case category $\mathcal{C}$ is a dagger case category with*

$$f + g = (f\nearrow + g\nearrow, f\searrow + g\searrow) \tag{164}$$
$$0 = (0, 0) \tag{165}$$
$$f \leq g \iff f\nearrow \leq g\nearrow \wedge f\searrow \leq g\searrow \tag{166}$$

Proof. The monoid and partial order laws are easy to check. The dagger structure is the normal janus category one.

$$f\,;g\,;h + f\,;g'\,;h \tag{167}$$
$$=((f\,;g\,;h)\nearrow + (f\,;g'\,;h)\nearrow, (f\,;g\,;h)\searrow + (f\,;g'\,;h)\searrow) \tag{168}$$
$$=((f\nearrow\,;g\nearrow\,;h\nearrow) + (f\nearrow\,;g'\nearrow\,;h\nearrow), (h\searrow\,;g\searrow\,;f\searrow) + (h\searrow\,;g'\searrow\,;f\searrow)) \tag{169}$$
$$\leq(f\nearrow\,;(g\nearrow + g'\nearrow)\,;h\nearrow, h\searrow\,;(g\searrow + g'\searrow)\,;f\searrow) \tag{170}$$
$$=(f\,;(g + g')\,;h)\nearrow, (f\,;(g + g')\,;h)\searrow) \tag{171}$$
$$=f\,;(g + g')\,;h \tag{172}$$

$$0 \tag{173}$$
$$=(0, 0) \tag{174}$$
$$\leq(f\nearrow\,;0\,;h\nearrow, h\searrow\,;0\,;f\searrow) \tag{175}$$
$$=f\,;0\,;h \tag{176}$$

$$(f + g)^\dagger \tag{177}$$
$$=(f\nearrow + g\nearrow, f\searrow + g\searrow)^\dagger \tag{178}$$
$$=(f\searrow + g\searrow, f\nearrow + g\nearrow) \tag{179}$$
$$=f^\dagger + g^\dagger \tag{180}$$

Lemma 18. Set_*, $\mathsf{Set}_{0\infty}$, Set^*, $\mathcal{P}(\mathsf{Set})$ *and* $\mathcal{P}_{<\infty}(\mathsf{Set})$ *are case-categories with the hom-posets and hom-monoids from the previous section and the restrictions as follows:*

- Set_**:* $\overline{f} = \lambda\alpha.\begin{cases} * \text{ if } f\,\alpha = * \\ \alpha \text{ otherwise} \end{cases}$
- $\mathsf{Set}_{0\infty}$*:* $\overline{f} = \lambda\alpha.\begin{cases} 0 \text{ if } f\,\alpha = 0 \\ \infty \text{ if } f\,\alpha = \infty \\ \alpha \text{ otherwise} \end{cases}$
- Set^**:* $\overline{f} = \lambda(a_i)_{i\in n\downarrow}.\mathsf{join}\,((a_i)_{j\in|f\,(a_i)_1|\downarrow})_{i\in n\downarrow}$
- $\mathcal{P}(\mathsf{Set})$, $\mathcal{P}_{<\infty}(\mathsf{Set})$*:* $\overline{f} = \lambda\alpha.\{a \in \alpha \mid f\,\{a\} \neq \emptyset\}$

Proof.

$$
\begin{aligned}
& g \leq g' && (181)\\
\iff & g\,b \leq g'\,b && (182)\\
\implies & h\,(g\,(f\,a)) \leq h\,(g\,(f\,a)) && \{f \text{ and } h \text{ are monotone}\} \quad (183)\\
\iff & f\,;g\,;h \leq f\,;g'\,;h && (184)
\end{aligned}
$$

$$
\begin{aligned}
& f \leq f' \wedge g \leq g' && (185)\\
\implies & && (186)
\end{aligned}
$$

$$
\begin{aligned}
& h\,c + h\,c' \leq h\,(c + c') && \{h \text{ superaddit} \quad (187)\\
\implies & h\,(g\,(f\,a)) + h\,(g'\,(f\,a)) \leq h\,(g\,(f\,a) + g'\,(f\,a))) && (188)\\
\implies & h\,(g\,(f\,a)) + h\,(g\,(f\,a')) \leq h\,((g \dot{+} g')\,(f\,a)) && \{g \text{ superadditive, } h \text{ monoto} \quad (189)\\
\implies & f\,;g\,;h \dot{+} f\,;g'\,;h \leq f\,;(g \dot{+} g')\,;h && (190)
\end{aligned}
$$

Set_*:

α	$f\,\alpha$	$g\,\alpha$	$\overline{f}\,\alpha$	$\overline{g}\,\alpha$	$(f+g)\,\alpha$	$\overline{f+g}\,\alpha$		$\overline{f}\,\alpha + \overline{g}\,\alpha$
a	b	β'	a	α'	b	a	$\leq$	a
a	$*$	b'	$*$	a	b'	a	$\leq$	a
α	$*$	$*$	$*$	$*$	$*$	$*$	$\leq$	$*$

$\mathsf{Set}_{0\infty}$:

α	$f\,\alpha$	$g\,\alpha$	$\overline{f}\,\alpha$	$\overline{g}\,\alpha$	$(f+g)\,\alpha$	$\overline{f+g}\,\alpha$		$\overline{f}\,\alpha + \overline{g}\,\alpha$
a	b	b'	a	a	∞	∞	$\leq$	∞
a	b	0	a	0	b	a	$\leq$	a
α	β	∞	α'	∞	∞	∞	$\leq$	∞
α	0	0	0	0	0	0	$\leq$	0

Set^*:

$$
\begin{aligned}
& \overline{f +^* g}\,(a_i)_{i \in n\downarrow} && (191)\\
=& \mathsf{join}\,((a_i)_{j \in |f\,(a_i)_1 + g\,(a_i)_1|\downarrow})_{i \in n\downarrow} && (192)\\
=& \mathsf{join}\,((a_i)_{j \in |f\,(a_i)_1|\downarrow} + ((a_i)_{j \in |g\,(a_i)_1|\downarrow})_{i \in n\downarrow} && (193)\\
=& (\overline{f} +^* \overline{g})\,(a_i)_{i \in n\downarrow} && (194)
\end{aligned}
$$

$$
\begin{aligned}
& (f\,;g\,;h \dot{+} f\,;g'\,;h)\,a && (195)\\
=& h\,(g\,(f\,a)) + h\,(g'\,(f\,a)) && (196)\\
\leq& h\,(g\,(f\,a) + g'\,(f\,a)) && (197)\\
=& (f\,;(g \dot{+} g')\,;h)\,a && (198)
\end{aligned}
$$

$$
\begin{aligned}
&(f\,;g\,;h +^{*} f\,;g'\,;h)\,(a_i)_{i\in n\downarrow} && (199)\\
=&\mathsf{join}_B\,((f\,;g\,;h)\,(a_i)_1 + (f\,;g'\,;h)\,(a_i)_1)_{i\in n\downarrow} && (200)\\
\leq & && (201)\\
=&h\,(\mathsf{join}_B\,(g\,((f\,a)_i)_1 + g'\,((f\,a)_i)_1)_{i\in|f\,a|\downarrow}) && (202)\\
=&(f\,;(g \dot{+} g')\,;h)\,a && (203)
\end{aligned}
$$

Lemma 19. $\mathsf{JSet}_{0\infty}$ *with the symmetric first-match policy is a case category.*

Proof. The monoid structure is inherited from $\mathsf{Set}_{0\infty}$.

3.6 21.10. – Generalization (Final)

Recap Restriction Categories

Definition 8. *A restriction category is a category equipped with a mapping that assigns to every morphism $f\colon A \to B$ a morphism $\overline{f}\colon A \to A$ such that*

(i) $\overline{f}\,;f = f$
(ii) $\overline{f}\,;\overline{f'} = \overline{f'}\,;\overline{f}$
(iii) $\overline{\overline{f}\,;g} = \overline{f}\,;\overline{g}$
(iv) $f\,;\overline{g} = \overline{f\,;g}\,;f$

Lemma 20. *In any restriction category we have the following.*

(i) $\overline{\overline{f^*}} = \overline{f^*}$
(ii) $\overline{f^*\,;\overline{g^*}} = \overline{f^*\,;g^*}$
(iii) $\overline{f} = f\,;g \implies \overline{f\,;\overline{h}} = f\,;\overline{h}\,;g$

Proof.

$$
\begin{aligned}
\overline{\overline{f}} &= \overline{\overline{f}\,;\mathrm{id}} && (204)\\
&= \overline{f}\,;\overline{\mathrm{id}} && (205)\\
&= \overline{f}\,;\overline{\mathrm{id}}\,;\mathrm{id} && (206)\\
&= \overline{f}\,;\mathrm{id} && (207)\\
&= \overline{f} && (208)\\
& && (209)
\end{aligned}
$$

$$
\begin{aligned}
\overline{f\,;\overline{g}} &= \overline{\overline{f\,;g}\,;f} && (210)\\
&= \overline{f\,;g}\,;\overline{f} && (211)\\
&= \overline{f}\,;\overline{f\,;g} && (212)\\
&= \overline{\overline{f}\,;f\,;g} && (213)\\
&= \overline{f\,;g} && (214)
\end{aligned}
$$

$$\overline{f\,;\overline{h}} = \overline{\overline{f}\,;f\,;\overline{h}} \tag{215}$$
$$= \overline{f}\,;\overline{f\,;\overline{h}} \tag{216}$$
$$= \overline{\overline{f\,;\overline{h}}\,;\overline{f}} \tag{217}$$
$$= \overline{\overline{f\,;\overline{h}}\,;f\,;g} \tag{218}$$
$$= f\,;\overline{\overline{h}}\,;g \tag{219}$$
$$= f\,;\overline{h}\,;g \tag{220}$$

Case Categories and Relaxed Isomorphisms

Definition 9. *A category $\mathcal{C}$ is called a* case category *if*

- *the hom-sets are equipped with a partial order $(\leq_{\mathcal{C}(A,B)})$,*
- *the hom-sets are equipped with a monoid structure $(+_{\mathcal{C}(A,B)}, 0_{\mathcal{C}(A,B)})$ and*
- *the category is equipped with a restriction structure,*

such that

(i) $g \leq g' \implies f\,;g\,;h \leq f\,;g'\,;h$
(ii) $f \leq f' \implies \overline{f} \leq \overline{f'}$
(iii) $\overline{f+f'} \leq \overline{f} + \overline{f'}$ *and* $\overline{0_{A,B}} = 0_{A,A}$

A case category is called stable *if*

(iv) $f \leq f' \implies f'' + f + f''' \leq f'' + f' + f'''$

A case category is called superadditive *rsp.* subadditive *rsp.* additive *if*

(v) $f\,;g\,;h + f\,;g'\,;h \leq$ *rsp.* $\geq$ *rsp.* $= f\,;(g+g')\,;h$

A case category is called increasing *if*

(vi) $f + f'' \leq f + f' + f''$

A case category has zero morphisms *if*

(vii) $f\,;0 = 0 = 0\,;f$

A case functor *is a functor between case categories that preserves partial order, monoid and restrictions.*

Remark 1. 9.(i) implies that case-categories are enriched over posets.
9.(ii) implies that $\overline{-}$ is a monotone map.
9.(iii) implies that $\overline{-}$ is a lax monoid homomorphism.
9.(iv) implies that the hom-set monoids are partially ordered monoids.
9.(v) implies that the monoid operation is a (oplax/lax) natural transformation.

Remark 2. The partial order of Set^* (see later) is neither the partial order induced by the restriction structure nor the Nambooripad order induced by its monoid. Neither is it based on semilattices which would allow the monoid and partial order be derived from that.

!Aufsplitten:

Lemma 21. *Any category can be extended to a (trivial) case category.*

Proof. Let $\mathcal{C}$ be any category. Define the case category $\mathcal{C}'$ as the following

- the objects of $\mathcal{C}'$ are the objects of $\mathcal{C}$,
- the morphisms of $\mathcal{C}'$ are the morphisms of $\mathcal{C}$ with a new additonal zero-morphism for each hom-set,
- composition the same as $\mathcal{C}$ with zero-morphisms always composing to zero-morphisms,
- the partial order on the hom-sets is equality,
- the monoid on the hom-sets is $f + f' = \begin{cases} f \text{ if } f \neq 0 \\ f' \text{ if } f = 0 \end{cases}$ and 0,
- the restriction structure is given by $\overline{0} = 0$ and otherwise $\overline{f} = \mathrm{id}$.

Definition 10. *A pair of morphisms $(f, g)\colon (A \to B) \times (B \to A)$ of a case category is called*

(i) *a* total semi-isomorphism *if* $\mathrm{id} = f\,;g$,
(ii) *a* supertotal semi-isomorphism *if* $\mathrm{id} \leq f\,;g$,
(iii) *a* partial semi-isomorphism *if* $\overline{f} = f\,;g$, *and*
(iv) *a* superpartial semi-isomorphism *if* $\overline{f} \leq f\,;g$.

A total (rsp. supertotal rsp. partial rsp. superpartial) semi-isomorphism (f, g) is called a total (rsp. supertotal rsp. partial rsp. superpartial) isomorphism *if also (g, f) is a total (rsp. supertotal rsp. partial rsp. superpartial) semi-isomorphism.*

Example 1. Let the ambient category be Set^*, the category of lists (free monoids over Set) and functions that commute with list concatenation ($f\,(a + a') = f\,a + f\,a'$, i.e., f is additive), which is the Kleisli-category of the list monad. This can be used to model a programming language with non-determinism (or more appropriately with built-in backtracking). Now define the following pair of functions:

$$g{\nearrow}\colon \{\star\}^* \to \{0,1\}^* \tag{221}$$
$$g{\nearrow}\,\mathsf{nil} = \mathsf{nil} \tag{222}$$
$$g{\nearrow}\,\mathsf{cons}\,(\star, r) = \mathsf{cons}\,(0, \mathsf{cons}\,(1, g{\nearrow}\,r)) \tag{223}$$
$$g{\searrow}\colon \{0,1\}^* \to \{\star\}^* \tag{224}$$
$$g{\searrow}\,\mathsf{nil} = \mathsf{nil} \tag{225}$$
$$g{\searrow}\,\mathsf{cons}\,(\_, r) = \mathsf{cons}\,(\star, g{\searrow}\,r) \tag{226}$$

The pair $(g\nearrow, g\searrow)$ is very useful as the basic building block for divergent (in the non-determinism sense) reversible (conversible?) computations, but it is neither an isomorphism nor a partial isomorphism. It is however a supertotal isomorphism which means that $(g\nearrow ; g\searrow)\, x$ will contain at least the elements of x in the same order and multiplicity and similarly for $g\searrow ; g\nearrow$.

Lemma 22. *A case functor preserves all the properties pairs of Definition 10.*

Proof. Let F be a case functor and (f, g) be a superpartial semi-isomorphism.

$$\overline{F\, f} = F\, \overline{f} \leq F\, (f\,;g) = F\, f\,;F\, g \tag{227}$$

Others are even easier.

Because by definition F preserves all structure in the defintion.

Lemma 23. *In a stable case category we have* $\overline{f} \leq f\,;g \implies \overline{f\,;\overline{h}} \leq f\,;\overline{h}\,;g$.

Proof.

$$\begin{aligned}
\overline{f\,;\overline{h}} &= \overline{\overline{f}\,;f\,;\overline{h}} && (228)\\
&= \overline{f}\,;\overline{f\,;\overline{h}} && (229)\\
&= \overline{f\,;\overline{h}}\,;\overline{f} && (230)\\
&\leq \overline{f\,;\overline{h}}\,;f\,;g && (231)\\
&= f\,;\overline{\overline{h}}\,;g && (232)\\
&= f\,;\overline{h}\,;g && (233)
\end{aligned}$$

Lemma 24. *If* (f, g) *and* (f', g') *are total rsp. supertotal rsp. partial rsp. superpartial semi-isomorphisms then so is* $(f\,;f', g'\,;g)$.

Proof. Let (f, g) and (f', g') be total semi-isomorphisms.

$$\begin{aligned}
&\mathrm{id} && (234)\\
=\ &\{\text{Def. 10.(i)}\} && (235)\\
&f\,;g && (236)\\
=\ &\{\mathrm{id}\} && (237)\\
&f\,;\mathrm{id}\,;g && (238)\\
=\ &\{\text{Def. 10.(i)}\} && (239)\\
&f\,;f'\,;g'\,;g && (240)
\end{aligned}$$

Let (f, g) and (f', g') be supertotal semi-isomorphisms.

$$
\begin{aligned}
& \mathrm{id} && (241)\\
\leq\ & \{\text{Def. 10.(ii)}\} && (242)\\
& f\,;g && (243)\\
=\ & \{\mathrm{id}\} && (244)\\
& f\,;\mathrm{id}\,;g && (245)\\
\leq\ & \{\text{Def. 10.(ii), Def. 9.(i)}\} && (246)\\
& f\,;f'\,;g'\,;g && (247)
\end{aligned}
$$

Let (f, g) and (f', g') be partial semi-isomorphisms.

$$
\begin{aligned}
& \overline{f\,;f'} && (248)\\
=\ & \{20.(\mathrm{ii})\} && (249)\\
& \overline{f\,;\overline{\overline{f'}}} && (250)\\
=\ & \{\text{Def. 10.(iii), Lemma 20.(iii)}\} && (251)\\
& f\,;\overline{f'}\,;g && (252)\\
=\ & \{\text{Def. 10.(iii)}\} && (253)\\
& f\,;f'\,;g'\,;g && (254)
\end{aligned}
$$

Let (f, g) and (f', g') be superpartial semi-isomorphisms.

$$
\begin{aligned}
& \overline{f\,;f'} && (255)\\
=\ & \{20.(\mathrm{ii})\} && (256)\\
& \overline{f\,;\overline{\overline{f'}}} && (257)\\
\leq\ & \{\text{Def. 10.(iv), Lemma 23}\} && (258)\\
& f\,;\overline{f'}\,;g && (259)\\
\leq\ & \{\text{Def. 10.(iv), Def. 9.(i)}\} && (260)\\
& f\,;f'\,;g'\,;g && (261)
\end{aligned}
$$

Lemma 25. *If (f, g) and (f', g') supertotal rsp. superpartial semi-isomorphisms of a stable, superadditive and increasing case category then so is $(f + f', g + g')$.*

If (f, g) and (f', g') total rsp. partial semi-isomorphisms of a additive and increasing case category with equality as partial order then so is $(f + f', g + g')$.

Proof. Let (f, g) and (f', g') be supertotal.

$$\begin{aligned}
& \overline{f + f'} && (262)\\
\leq\ & \{\text{Def. 9.(iii)}\} && (263)\\
& \overline{f} + \overline{f'} && (264)\\
=\ & \{\text{Def. 10.(ii)}\} && (265)\\
& f\,;g + f'\,;g' && (266)\\
\leq\ & \{\text{Def. 9.(vi)}\} && (267)\\
& f\,;g + f\,;g' + f'\,;g + f'\,;g' && (268)\\
\leq\ & \{\text{Def. 9.(iv), Def. 9.(v)}\} && (269)\\
& f\,;(g + g') + f'\,;(g + g') && (270)\\
\leq\ & \{\text{Def. 9.(iv), Def. 9.(v)}\} && (271)\\
& (f + f')\,;(g + g') && (272)
\end{aligned}$$

Let (f, g) and (f', g') be superpartial.

$$\begin{aligned}
& \overline{f + f'} && (273)\\
\leq\ & \{\text{Def. 9.(iii)}\} && (274)\\
& \overline{f} + \overline{f'} && (275)\\
\leq\ & \{\text{Def. 10.(iv), Def. 9.(iv)}\} && (276)\\
& f\,;g + f'\,;g' && (277)\\
\leq\ & \{\text{as above}\} && (278)\\
& (f + f')\,;(g + g') && (279)
\end{aligned}$$

For total and partial semi-isomorphisms the proofs are the same as above with $\leq$ replaced by $=$. Stableness is trivially true when the partial order is equality.

Recap Kleisli-triple

Definition 11. *A Kleisli-triple on a category $\mathcal{C}$ is a triple $(T, \eta, -^*)$ where*

- *T is mapping of objects of $\mathcal{C}$,*
- *$\eta_A \colon A \to T\,A$ is a family of morphisms of $\mathcal{C}$,*
- *$f^* \colon T\,A \to T\,B$ for $f \colon A \to T\,B$ is a mapping of morphisms of $\mathcal{C}$*

such that

(i) $\eta_A^* = \mathrm{id}_{T\,A}$
(ii) $\eta\,; f^* = f$
(iii) $f^*\,; g^* = (f\,; g^*)^*$

The Kleisli-category C_T of a Kleisli-triple over a category $\mathcal{C}$ is the subcategory of $\mathcal{C}$ that has only the objects of the form $T\,A$ and only the morphisms of the form f^*.

Case Monad and Examples

Definition 12. *A Kleisli-triple* $(T, \eta, -^*)$ *on* Set *is called a* case-monad *if*

- *all objects* $T\,A$ *are equipped with a partial order* $\leq_A$,
- *all objects* $T\,A$ *are equipped with a monoid* $(+_A, 0_A)$,
- *for every morphism* $f\colon A \to T\,B$ *there is a morphism* $\overline{f}\colon A \to T\,A$ *adhering to the restriction laws where* $\overline{f^*} = \overline{f}^*$ *(abusing notation)*

such that for all A, B, $f\colon A \to B$

- f^* *is monotone,*
- *the restrictions are pointwise compatible with the partial order and the monoid (see Definition 9).*

A case-monad where $b \leq b' \implies a + b + c \leq a + b' + c$ *is called* stable.

A case-monad where all f^* *are superadditive rsp. subadditive rsp. additive is called* superadditive *rsp.* subadditive *rsp.* additive.

A case-monad where $a + c \leq a + b + c$ *is called* increasing.

!Converse?

Lemma 26. *The properties of a case-monad* T *extend to its Kleisli-arrows in the following ways:*

(i) $f^* \leq_{A\to B} f'^* \iff \forall a \in A \mid f\,a \leq f'\,a$ *is a partial order.*
(ii) $(+_{A\to B}, 0_{A\to B})$ *with* $(f^* +_{A\to B} f'^*) = (\lambda a.f\,a + f'\,a)^*$ *and* $0^*_{A\to B} = (\lambda a.0_B)^*$ *is a monoid.*
(iii) If T *is stable then* $f^* \leq f'^* \implies f''^* + f^* + f'''^* \leq f''^* + f'^* + f'''^*$.
(iv) If T *is increasing then* $f^* + f''^* \leq f^* + f'^* + f''^*$.

Proof. Partial order.

$$f^* \leq f^* \iff \forall a \in A \mid f\,a \leq f\,a \iff \top \tag{280}$$
$$f^* \leq f'^* \wedge f'^* \leq f''^* \iff \forall a \in A \mid f\dot{a} \leq f'\,a \wedge f'\dot{a} \leq f''\,a \tag{281}$$
$$\implies \forall a \in A \mid f\,a \leq f''\,a \tag{282}$$
$$\iff f^* \leq f''^* \tag{283}$$
$$f^* \leq f'^* \wedge f'^* \leq f^* \iff \forall a \in A \mid f\dot{a} \leq f'\,a \wedge f'\dot{a} \leq f\,a \tag{284}$$
$$\implies \forall a \in A \mid f\,a = f'\,a \tag{285}$$
$$\iff f^* = f'^* \tag{286}$$

Monoid.

$$f^* + 0^* = (\lambda a.f\ a + 0)^* = f^* \tag{287}$$
$$0^* + f^* = (\lambda a.0 + f\ a)^* = f^* \tag{288}$$
$$(f^* + f'^*) + f''^* = (\lambda a.f\ a + f'\ a)^* + f''^* \tag{289}$$
$$= (\lambda a.(\lambda a.f\ a + f'\ a)\ a + f''\ a)^* \tag{290}$$
$$= (\lambda a.(f\ a + f'\ a) + f''\ a)^* \tag{291}$$
$$= (\lambda a.f\ a + (f'\ a + f''\ a))^* \tag{292}$$
$$= (\lambda a.f\ a + (\lambda a.f'\ a + f''\ a)\ a)^* \tag{293}$$
$$= f^* + (\lambda a.f'\ a + f''\ a)^* \tag{294}$$
$$= f^* + (f'^* + f''^*) \tag{295}$$

Monotonicity.

$$g^* \leq g'^* \iff \forall b \in T\ B \mid g^*\ b \leq g'^*\ b \tag{296}$$
$$\implies \forall a \in T\ A \mid g^*\ (f^*\ a) \leq g'^*\ (f^*\ a) \tag{297}$$
$$\implies \forall a \in T\ A \mid h^*\ (g^*\ (f^*\ a)) \leq h^*\ (g'^*\ (f^*\ a)) \tag{298}$$
$$\iff f^*\,;g^*\,;h^* \leq f^*\,;g'^*\,;h^* \tag{299}$$

Stableness.

$$f^* \leq f'^* \iff \forall a \in A \mid f\ a \leq f'\ a \tag{300}$$
$$\implies \forall a \in A \mid f''\ a + f\ a + f'''\ \leq f''\ a + f'\ a + f'''\ a \tag{301}$$
$$\iff f''^* + f^* + f'''^* \leq f''^* + f^* + f'''^* \tag{302}$$

Increasing.

$$f^* + f''^* = (\lambda a.f\ a + f''\ a)^* \tag{303}$$
$$\leq (\lambda a.f\ a + f'\ a + f''\ a)^* \tag{304}$$
$$= f^* + f'^* + f''^* \tag{305}$$

Corollary 2. *The Kleisli-category of a case-monad is a case category. It is stable rsp. superadditive rsp. subadditive rsp. additive rsp. increasing if the case-monad is.*

!Aufdröseln

Lemma 27. *A Kleisli-triple $(T, \eta, -^*)$ such that*

- *T is a commutative monad,*
- *$T\ A$ is a join-semilattice $(\vee)$ with least element $(\bot)$,*
- *f^* is preserves finite join for all f*

is a stable, additive, increasing case-monad by

- $a + a' = a \vee a'$

- $0 = \bot$
- $a \leq a' \iff a \vee a' = a'$
- $\overline{f^*} = \langle f, \mathrm{id} \rangle^* ; (\pi_2 ; \eta)^* = (\lambda a.(\lambda\_.\eta\, a)^*\, (f\, a))^*$

Proof. Partial order, monoid and restriction structure are well-known.

Monotonicity:

$$\begin{aligned} a \leq a' &\iff a' = a \vee a' && (306)\\ &\implies f\, a' = f\, (a \vee a') = f\, a \vee f\, a' && (307)\\ &\iff f\, a \leq f\, a' && (308) \end{aligned}$$

Stableness:

$$\begin{aligned} b \leq b' &\iff b \vee b' = b' && (309)\\ &\implies a \vee b \vee b' \vee c = a \vee b' \vee c && (310)\\ &\iff (a \vee b \vee c) \vee (a \vee b' \vee c) = a \vee b' \vee c && (311)\\ &\iff a + b + c \leq a + b' + c && (312) \end{aligned}$$

Compatibility of restrictions:

$$\begin{aligned} \overline{f^* + f'^*} &= (\langle f, \mathrm{id} \rangle^* + \langle f', \mathrm{id} \rangle^*) ; (\pi_2 ; \eta)^* && (313)\\ &= \langle f, \mathrm{id} \rangle^* ; (\pi_2 ; \eta)^* + \langle f', \mathrm{id} \rangle^* ; (\pi_2 ; \eta)^* && (314)\\ &= \overline{f^*} + \overline{f'^*} && (315) \end{aligned}$$

$$\begin{aligned} f \leq f' &\iff \forall a \in A \mid f^*\, a \leq f'^*\, a && (316)\\ &\implies \forall a \in A \mid (\lambda b.\eta\, a)^*\, (f\, a) \leq (\lambda b.\eta\, a)^*\, (f'\, a) && (317)\\ &\implies \forall a \in A \mid \overline{f^*}\, a \leq \overline{f'^*}\, a && (318)\\ &\iff \overline{f^*} \leq \overline{f'^*} && (319) \end{aligned}$$

Increasing:

$$\begin{aligned} a + c &= a \vee c && (320)\\ &\leq a \vee c \vee b && (321)\\ &= a \vee b \vee c && (322)\\ &= a + b + c && (323) \end{aligned}$$

Corollary 3. *The powerset monad, the finite powerset monad, the multiset monad (with $+ = \max$), ~~the distribution monad~~ (no zeroes, trivial restrictions) are case-monads.*

Lemma 28. Set_* *is a subadditive case-monad that is neither stable nor increasing.*

Proof.

$$
\begin{aligned}
f^* \leq f'^* &\iff \forall a \in A \mid f\, a \leq f'\, a && (324)\\
&\iff \forall a \in A \mid f\, a = 0 \vee f\, a = f'\, a && (325)\\
&\implies \forall a \in A \mid f\, a = 0 \vee f'\, a \neq 0 && (326)\\
&\iff \overline{f^*} \leq \overline{f'^*} && (327)
\end{aligned}
$$

$$
\begin{aligned}
\overline{f^* + f'^*} &= \left(\lambda a. \begin{cases} a \text{ if } f\, a \neq 0 \vee f'\, a \neq 0 \\ 0 \text{ if } f\, a = 0 \vee f'\, a = 0 \end{cases}\right)^* && (328)\\
&= \overline{f^*} + \overline{f'^*} && (329)
\end{aligned}
$$

$$
0 \leq a \implies 0 + 0 + a' \leq 0 + a + a' \tag{330}
$$

Lemma 29. $\mathsf{Set}_{0\infty}$ *is a stable, superadditive, increasing case-monad by*

$$
\overline{f^*} = \left(\lambda a. \begin{cases} 0 \ \textit{if}\ f\, a = 0 \\ a \ \textit{if}\ f\, a \neq 0 \end{cases}\right)^* \tag{331}
$$

and has a case product given by

$$
\begin{aligned}
A \oplus B &= (A + B)_{0\infty} && (332)\\
f^* \oplus g^* &= [f\,;(\iota_1\,;\eta)^*, g\,;(\iota_2\,;\eta)^*] && (333)
\end{aligned}
$$

Proof. Let $f, f' \colon A \to T\,B$, $g \colon A \to T\,C$, $h \colon B \to T|, C$, $m \in T\,A$, $a \in A$, $b \in B$.

$$(\overline{f^*}\,;f^*)\,m = \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge f\,a = 0) \\ \infty \text{ if } m = \infty \vee (m = a \wedge f\,a = \infty) \\ b \text{ if } m = a \wedge f\,a = b \end{cases} \tag{334}$$

$$= f^*\,m \tag{335}$$

$$(\overline{f^*}\,;\overline{f'^*})\,m = \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge (f\,a = 0 \vee f\,a \neq 0 \wedge f'\,a = 0) \\ \infty \text{ if } m = \infty \\ a \text{ if } m = a \wedge f\,a \neq 0 \wedge f'\,a \neq 0 \end{cases} \tag{336}$$

$$= \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge (f'\,a = 0 \vee f'\,a \neq 0 \wedge f\,a = 0) \\ \infty \text{ if } m = \infty \\ a \text{ if } m = a \wedge f'\,a \neq 0 \wedge f\,a \neq 0 \end{cases} \tag{337}$$

$$= (\overline{f'^*}\,;\overline{f^*})\,m \tag{338}$$

$$(\overline{\overline{f^*}\,;g^*})\,m = \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge (f\,a = 0 \vee g\,a = 0)) \\ \infty \text{ if } m = \infty \\ a \text{ if } m = a \wedge f\,a \neq 0 \wedge g\,a \neq 0 \end{cases} \tag{339}$$

$$= (\overline{f^*}\,;\overline{g^*})\,m \tag{340}$$

$$(f^*\,;\overline{h^*})\,m = \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge (f\,a = 0 \vee (f\,a = b \wedge h\,b = 0))) \\ \infty \text{ if } m = \infty \vee (m = a \wedge f\,a = \infty) \\ b \text{ if } m = a \wedge f\,a = b \wedge h\,b \neq 0 \end{cases} \tag{341}$$

$$= \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge (f\,a = 0 \vee (f\,a = b \wedge h\,b = 0) \vee f\,a = 0) \\ \infty \text{ if } m = \infty \vee (m = a \wedge f\,a = \infty \wedge f\,a = \infty) \\ b \text{ if } m = a \wedge f\,a = b \wedge h\,b \neq 0 \wedge f\,a = b \end{cases} \tag{342}$$

$$= (\overline{f^*\,;h^*}\,;f^*)\,m \tag{343}$$

$$f^* \leq f'^* \iff \forall a \in A \mid f\, a \leq f'\, a \tag{344}$$
$$\iff \forall a \in A \mid f\, a = 0 \vee f'\, a = \infty \tag{345}$$
$$\implies \forall a \in A \mid f\, a = 0 \vee f'\, a \neq 0 \tag{346}$$
$$\iff \overline{f^*} \leq \overline{f'^*} \tag{347}$$
$$\overline{f^* + f'^*}\, m = \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge f\, a = 0 \wedge f'\, a = 0) \\ \infty \text{ if } m = \infty \\ a \text{ if } m = \infty \vee (m = a \wedge (f\, a \neq 0 \vee f'\, a \neq 0)) \end{cases} \tag{348}$$
$$\leq \begin{cases} 0 \text{ if } m = 0 \vee (m = a \wedge f\, a = 0 \wedge f'\, a = 0) \\ \infty \text{ if } m = \infty \vee m = a \wedge f\, a \neq 0 \wedge f'\, a \neq 0 \\ a \text{ if } m = a \wedge ((f\, a \neq 0) \neq (f'\, a \neq 0)) \end{cases} \tag{349}$$
$$= (\overline{f^*} + \overline{f'^*})\, m \tag{350}$$

$$(f^*, g^*) \leq (f'^*, g'^*) \iff \forall a \in A \mid f\, a \leq f'\, a \wedge \forall b \in B \mid g\, b \leq g\, b \tag{351}$$
$$\iff \forall x \in A + B \mid [f\,;\iota_1^*, g\,;\iota_2^*] \leq [f\,;\iota_1^*, g\,;\iota_2^*] \tag{352}$$
$$\iff (f^* \oplus g^*) \leq (f'^* \oplus g^*) \tag{353}$$
$$(f^* + f'^*) \oplus (g^* + g'^*) = [(f^* + f'^*)\,;(\iota_1\,;\eta)^*, (g^* + g'^*)\,;(\iota_2\,;\eta)^*] \tag{354}$$
$$= [(\lambda a.f\, a + f'\, a)^*)\,;(\iota_1\,;\eta)^*, \dots] \tag{355}$$
$$= [((\lambda a.f\, a + f'\, a)\,;(\iota_1\,;\eta)^*)^*, \dots] \tag{356}$$
$$= (f^* \oplus g^*) + (f'^* \oplus g'^*) \tag{357}$$
$$\overline{f \oplus g} = \lambda x.0 \text{ if } [f\,;(\iota_1\,;\eta)^*, g\,;(\iota_2\,;\eta)^*]\, x = 0 \text{ else } \eta\, a \tag{358}$$
$$= [(\lambda a.0 \text{ if } f\, a = 0 \text{ else } \eta\, a)\,;(\iota_1\,;\eta)^*, \dots] \tag{359}$$
$$= \overline{f} \oplus \overline{g} \tag{360}$$

Case Products (WIP) Is there a non-trivial stable, additive and increasing case categoy with a trivial (equality) partial order? Yes.

Lemma 30. *Let $\mathcal{C}$ be restriction category. The category $(\mathcal{C} \times \mathcal{C})_0$ is a stable, additive and increasing case category where*

1. *the objects are the same as $\mathcal{C} \times \mathcal{C}$,*
2. *the morphisms are same as $\mathcal{C} \times \mathcal{C}$ with additional zero-morphisms,*
3. *composition is point-wise as in $\mathcal{C} \times \mathcal{C}$ with zero-morphisms composing to zero-morphisms,*
4. *the partial order is equality,*
5. $(f_L, f_R) + (f'_L, f'_R) = (f_L, f'_R)$,
6. $(f_L, f_R) + 0 = (f_L, f_R) = 0 + (f_L, f_R)$,
7. $0 + 0 = 0$,

8. $\overline{(f_L, f_R)} = (\overline{f_L}, \overline{f_R})$,
9. $\overline{0} = 0$.

Proof. Associativity of the monoid.

$$(f_L, f_R) + ((f'_L, f'_R) + (f''_L, f''_R)) = (f_L, f''_R) \tag{361}$$
$$= ((f_L, f_R) + (f'_L, f'_R)) + (f''_L, f''_R) \tag{362}$$

Compatibility between restriction and monoid.

$$\overline{(f_L, f_R) + (f'_L, f'_R)} = \overline{(f_L, f'_R)} \tag{363}$$
$$= (\overline{f_L}, \overline{f'_R}) \tag{364}$$
$$= (\overline{f_L}, \overline{f_R}) + (\overline{f'_L}, \overline{f'_R}) \tag{365}$$
$$= \overline{(f_L, f_R)} + \overline{(f'_L, f'_R)} \tag{366}$$

Increasingness.

$$(f_L, f_R) ; (g_L, g_R) ; (h_L, h_R) + (f_L, f_R) ; (g'_L, g'_R) ; (h_L, h_R) \tag{367}$$
$$= (f_L ; g_L ; h_L, f_R ; g_R ; h_R) + (f_L ; g'_L ; h_L, f_R ; g'_R ; h_R) \tag{368}$$
$$= (f_L ; g_L ; h_L, f_R ; g'_R ; h_R) \tag{369}$$
$$= (f_L, f_R) ; (g_L, g'_R) ; (h_L, h_R) \tag{370}$$
$$= (f_L, f_R) ; ((g_L, g_R) + (g'_L, g'_R)) ; (h_L, h_R) \tag{371}$$

How to identify "useful" monoids?

Definition 13. *A tensor product* $(\oplus, \emptyset, \mathrm{a}, \mathrm{l}, \mathrm{r})$ *in a case category with zero morphisms is called a* case product *if* $\oplus$ *is a case functor.*

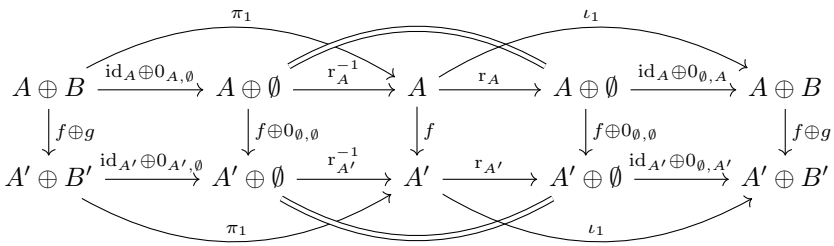
Lemma 31. *In a case category with a case product we have*

$$f \oplus f' = \pi_1 ; f ; \iota_1 + \pi_2 ; f' ; \iota_2$$

where

$$\pi_1 = (\mathrm{id}_A \oplus 0_{B,\emptyset}) ; \mathrm{r}_A \qquad \pi_2 = (0_{A,\emptyset} \oplus \mathrm{id}_B) ; \mathrm{l}_B \tag{372}$$
$$\iota_1 = \mathrm{r}_A^{-1} ; (\mathrm{id}_A \oplus 0_{\emptyset,B}) \qquad \iota_2 = \mathrm{l}_B^{-1} ; (0_{\emptyset,A} \oplus \mathrm{id}_B) \tag{373}$$

Proof. The following diagram commutes, because $\oplus$ is a functor, 0 is a zero morphism and r is a natural isomorphism.



$$
\begin{aligned}
f \oplus g &= (f + 0_{A,A'}) \oplus (0_{B,B'} + g) && (374)\\
&= (f \oplus 0_{B,B'}) + (0_{A,A'} \oplus g) && (375)\\
&= \quad (\mathrm{id}_A \oplus 0_{B,\emptyset})\,;(f \oplus 0_{\emptyset,\emptyset})\,;(\mathrm{id}_{A'} \oplus 0_{\emptyset,B'}) && (376)\\
&\quad + (0_{A,\emptyset} \oplus \mathrm{id}_B)\,;(0_{\emptyset,\emptyset} \oplus g)\,;(0_{\emptyset,B'} \oplus \mathrm{id}_{B'}) && (377)\\
&= \pi_1\,;f\,;\iota_1 + \pi_2\,;f'\,;\iota_2 && (378)
\end{aligned}
$$

Lemma 32. *Let T be a case monad on a cartesian closed category with coproduct $(- \oplus -, \emptyset, [-,-], \mathrm{inj}_1, \mathrm{inj}_2)$ such that*

$$
\begin{aligned}
f^* \le f'^* \wedge g^* \le g'^* \iff [f,g]^* \le [f',g']^* && (379)\\
[f^* + f'^*, g^* + g'^*] = [f^*, g^*] + [f'^*, g'^*] && (380)\\
\overline{[f,g]^*} = [\overline{f}, \overline{g}]^* && (381)
\end{aligned}
$$

then $((T-) \oplus (T-), T\,\emptyset, ([[\mathrm{inj}_1, \mathrm{inj}_2\,;\mathrm{inj}_1], \mathrm{inj}_2\,;\mathrm{inj}_2]\,;\eta)^*, (\mathrm{inj}_2\,;\eta)^*, (\mathrm{inj}_1\,;\eta)^*)$ is a case product in the Kleisli-category.

Proof.

$$
\begin{aligned}
& (f^*, g^*) \le (f'^*, g'^*) && (382)\\
\iff & f^* \le f'^* \wedge g^* \le g'^* && (383)\\
\iff & [f\,;(\mathrm{inj}_1\,;\eta)^*, g\,;(\mathrm{inj}_2\,;\eta)^*]^* \le [f'\,;(\mathrm{inj}_1\,;\eta)^*, g'\,;(\mathrm{inj}_2\,;\eta)^*]^* && (384)\\
\iff & (f^* \oplus g^*) \le f'^* \oplus g'^*) && (385)
\end{aligned}
$$

Lemma 33. *The cartesian product and the coalesced sum are case products in the case category of Lemma 27 where $T = \mathrm{Id}$, i.e, the category of join-semilattices and semilattice homomorphisms.*

Proof. Cartesian product:

$$
\begin{aligned}
\pi_1 &= (\mathrm{id}_A \times \bot_{B,1})\,;(\lambda(a,\bot).a) && (386)\\
\iota_1 &= (\lambda a.(a,\bot))\,;(\mathrm{id}_A \times \bot_{1,B}) && (387)\\
\pi_1\,;\iota_1 &= (\mathrm{id}_A \times \bot_{B,1})\,;(\mathrm{id}_A \times \bot_{1,1})\,;(\mathrm{id}_A \times \bot_{1,B}) && (388)\\
&= \mathrm{id}_A \times \bot_{B,B} && (389)\\
\pi_1\,;\iota_1 \sqcup \pi_2\,;\iota_2 &= (\mathrm{id}_A \times \bot_{B,B}) \sqcup (\bot_{A,A} \times \mathrm{id}_B) && (390)\\
&= \mathrm{id}_A \times \mathrm{id}_B && (391)
\end{aligned}
$$

Coalesced sum:

$$
\begin{aligned}
\pi_1 &= (\mathrm{id}_A \oplus \bot_{B,0})\,;(\lambda \mathrm{inj}_1\, a.a) && (392)\\
\iota_1 &= (\lambda a.\mathrm{inj}_1\, a)\,;(\mathrm{id}_A \oplus \bot_{0,B}) && (393)\\
\pi_1\,;\iota_1 &= (\mathrm{id}_A \oplus \bot_{B,0})\,;(\mathrm{id}_A \oplus \bot_{0,0})\,;(\mathrm{id}_A \oplus \bot_{0,B}) && (394)\\
&= \mathrm{id}_A \oplus \bot_{B,B} && (395)\\
\pi_1\,;\iota_1 \sqcup \pi_2\,;\iota_2 &= (\mathrm{id}_A \oplus \bot_{B,B}) \sqcup (\bot_{A,A} \oplus \mathrm{id}_B) && (396)\\
&= \mathrm{id}_A \oplus \mathrm{id}_B && (397)
\end{aligned}
$$

Lemma 34. *Set^* is a stable, additive, increasing case-monad.*

4 Universally and existentially quantified types

4.1 10.2.

For syntactic sugar like $[\![\texttt{if}\, e_1\, \texttt{then}\, e_2\, \texttt{else}\, e_3]\!]_{\mathsf{S}} = \mathsf{sel}\,(\lambda().e_2, \lambda().e_3)\, e_1\, ()$ does sel need to have the type $\forall A.A \times A \to [2 \to A]$ or is there a way around it?

5 Implementation

5.1 10.2.

Functional language (System F) implemented in Haskell : `https://github.com/stefanbohne/reversible`

Other idea: Template Haskell. Convert

`[| \(a, b) -> let s = a + b; d = b * (-2) + s in (s, d) |]`

into a pair of functions:

`\(a, b) -> let s = a + b; d = b * (-2) + s in (s, d)` and

`\(s, d) -> let b = (d - s) / (-2); a = s - b in (a, b)`

or: turn it into a point-free form which can already be handled by libraries.Other (minor) problem: case can fail in expression as well as pattern. Not typically how functional languages work.

6 Lenses

Lemma 35. *Existentially quantified pairs of inverse morphisms* $\exists R.(A \to B \times R) \times (B \times R \to A)$ *are isomorphic to well-behaved lenses between* A *and* B*.*

Lemma 36. *Morphism pairs of the form* $\exists R.(A \to B \times R) \times (B \times R \to A)$ *form a symmetric monoidal category.*

Proof. Composition is given by a morphism in the janus category

$$\begin{aligned} -\,;- &= [\![\boldsymbol{f}, \boldsymbol{g}; \boldsymbol{a} \vdash \texttt{let}\,(\texttt{b}, \boldsymbol{rf}) = \boldsymbol{f}\ \boldsymbol{a}\ \texttt{in let}\,(\texttt{c}, \boldsymbol{rg}) = \boldsymbol{g}\ \boldsymbol{b}\ \texttt{in}\,(\texttt{c}, (\boldsymbol{rf}, \boldsymbol{rg}))]\!]_{\dagger} \\ &= \lambda(f, g).f\,;^{\mathsf{J}\mathcal{C}}\,(g \otimes^{\mathsf{J}\mathcal{C}} \mathrm{id}) \\ &= \lambda(f, g).\begin{pmatrix} \lambda a.\mathsf{let}\,(b, r_f) = f \nearrow a\ \mathsf{in let}\,(c, r_g) = g \nearrow b\ \mathsf{in}\,(c, (r_f, r_g)) \\ \lambda(c, (r_f, r_g)).f \searrow (g \searrow (c, r_g), r_f) \end{pmatrix} \end{aligned}$$

so $R_{f\,;g} = R_f \otimes R_g$. Associativity is proven by associativity of $-\,;-$ and $-\otimes-$.

For isomorphisms in $\mathcal{C}$ we find that we can set $R = I$. So the identities and the symmetric monoidal structure is isomorphic to that in $\mathsf{J}\mathcal{C}$.