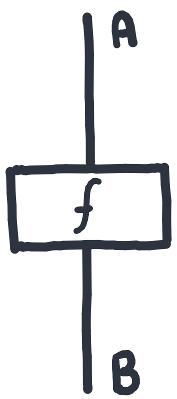
TIMING PROCESSES

joint with
Elena Di Lavore

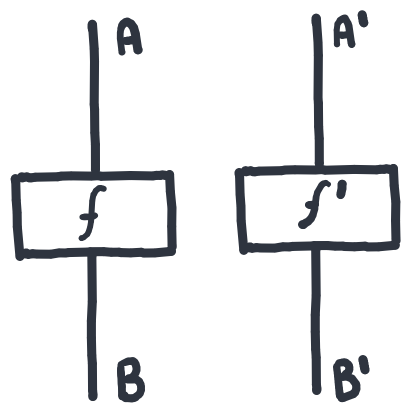
Walters Tribute.
Tallinn, 17th July 2023

Monoidal Categories: Process Theories

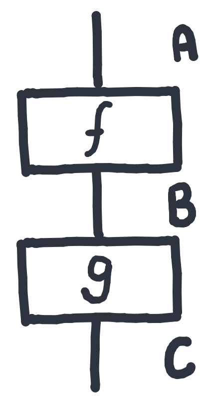
Monoidal categories are an algebra of parallel and sequential composition.
String diagrams are an internal language of monoidal categories.



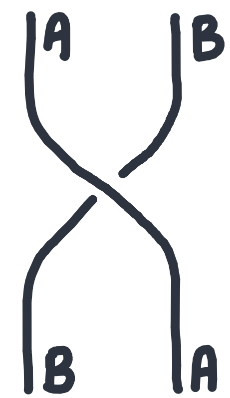
Process



Parallel composition



Sequential composition

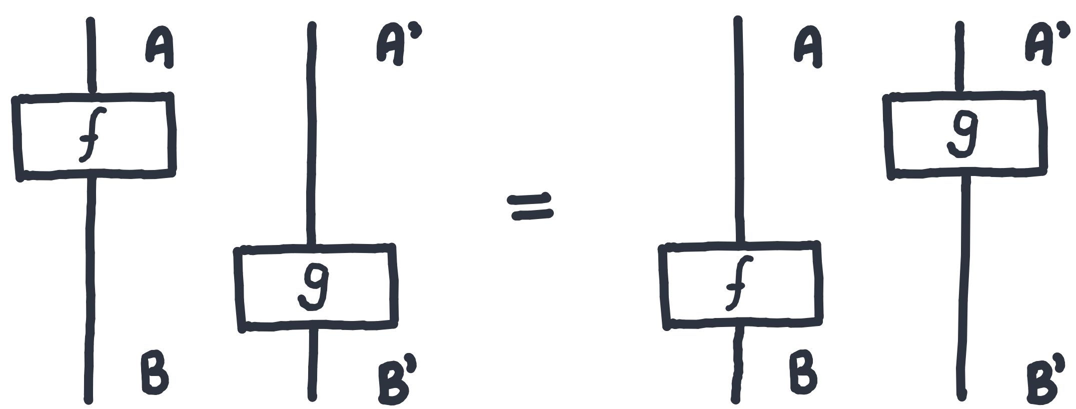


Swap

Bènabou

Monoidal Categories: Process Theories

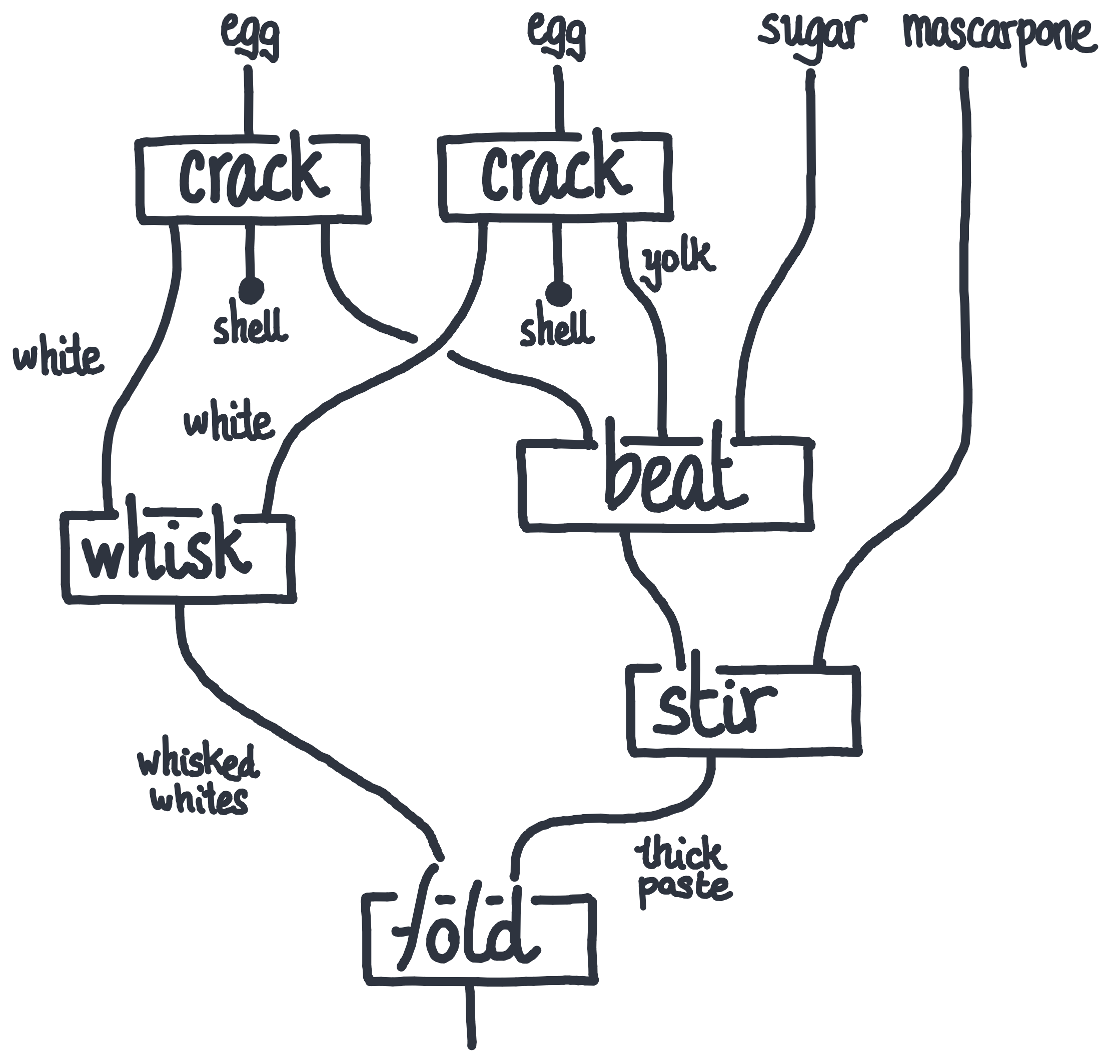
Monoidal categories are an algebra of parallel and sequential composition.
String diagrams are an internal language of monoidal categories.



Interchange Law

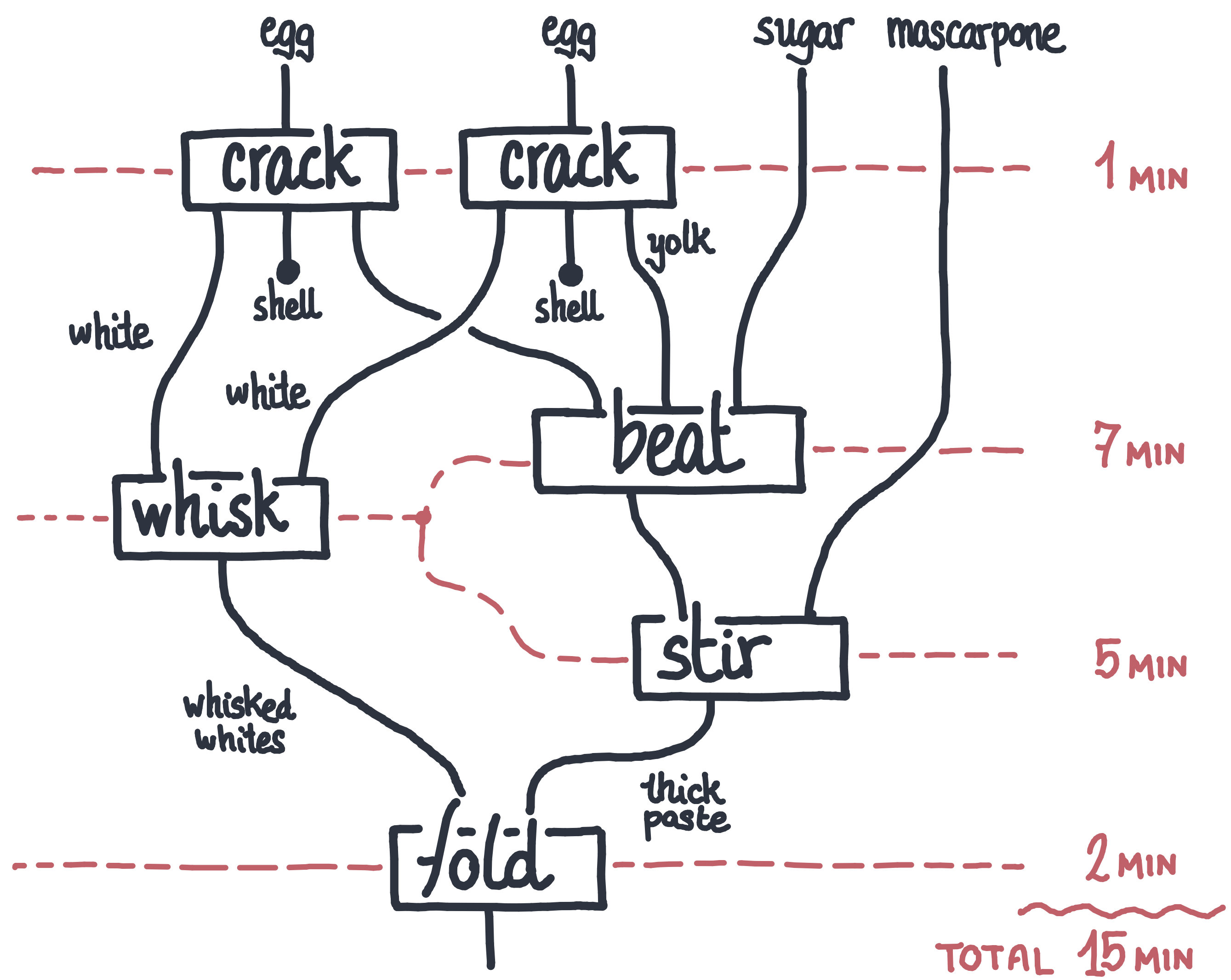
Bènabou

Crema di Mascarpone



Sobociński

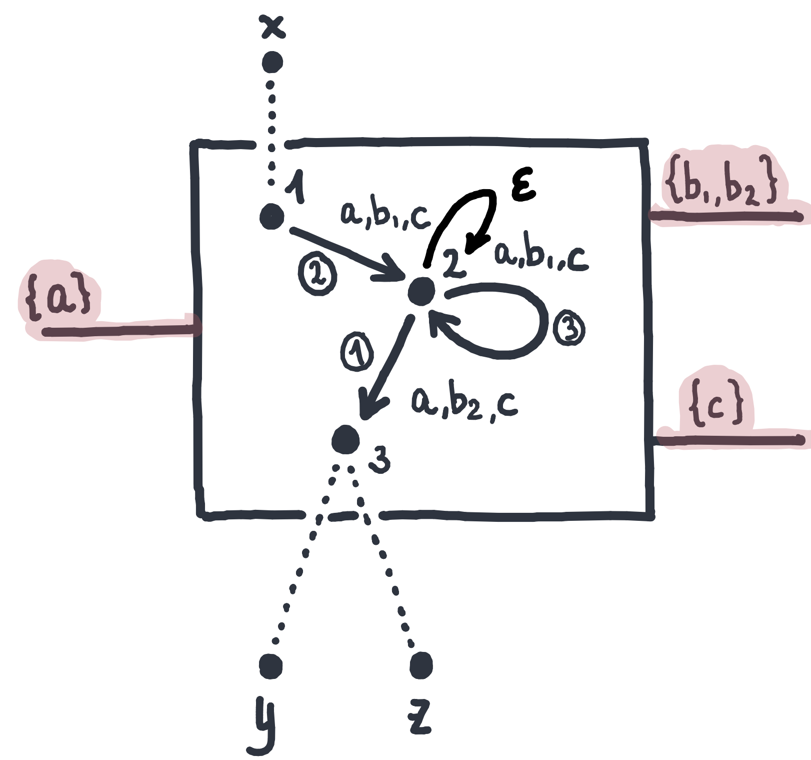
Timing Crema di Mascarpone



TIME?

Three ideas from Timing in Span(Graph).

1. Waiting, ε-transitions, help synchronizing.
2. Two boundaries: synchronizing vs sequential.
3. Synchronization is not I/O.

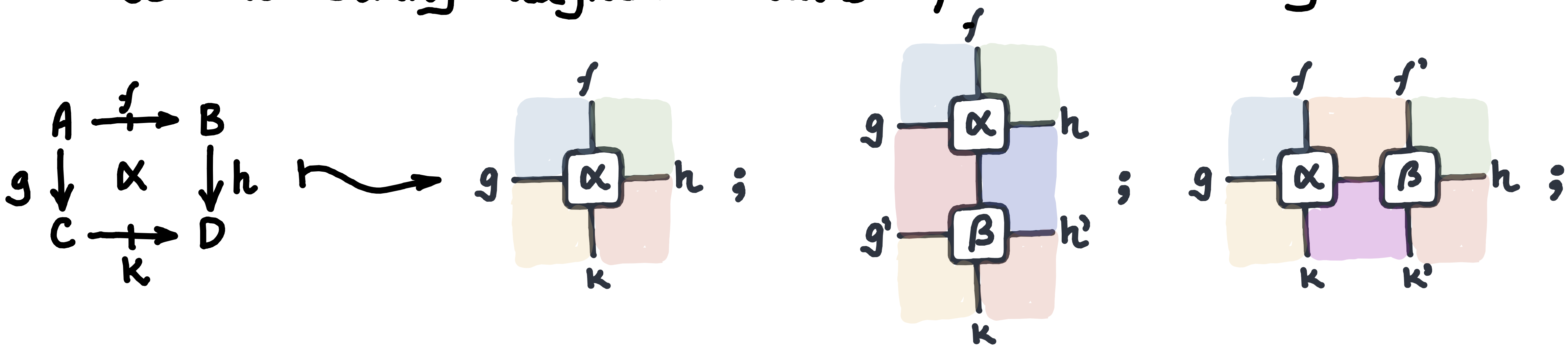


Cherubini, Sabadini, Walters. Timing in the Caspan-Span model.

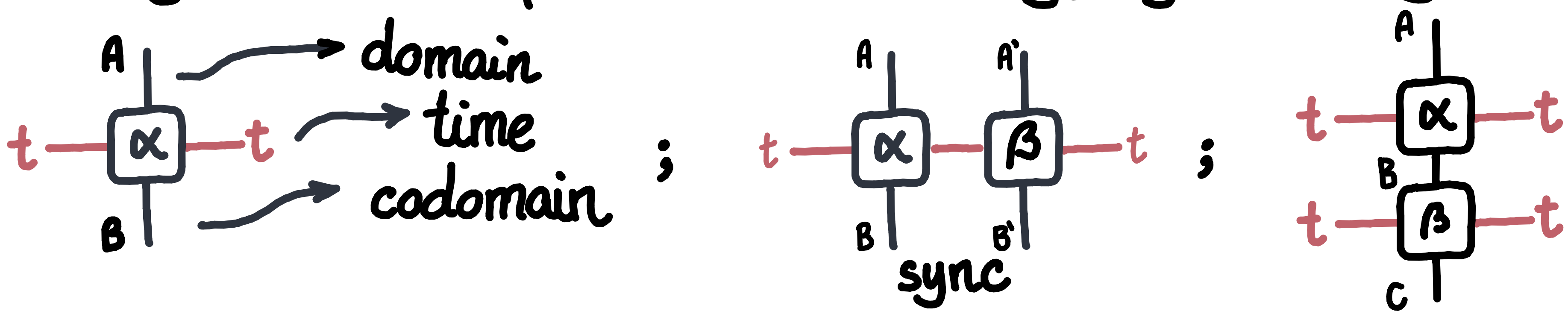
Part 1 : Timing in Double Categories

IDEA: SYNCHRONIZATION

We use the string diagram calculus of double categories.



Vertically, usual morphisms; horizontally, synchronizing time.

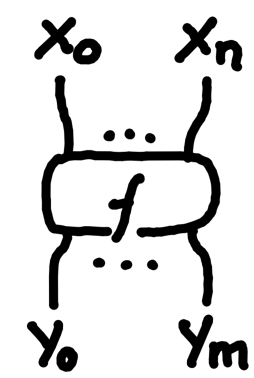


Timing in Double Categories

A polygraph $\mathcal{P}$ is a signature for a monoidal category:

- a set of objects $\mathcal{G}_{obj}$; and
- a set of generators for each two words of objects.

$$\mathcal{G}(X_0, \dots, X_n ; Y_0, \dots, Y_m).$$



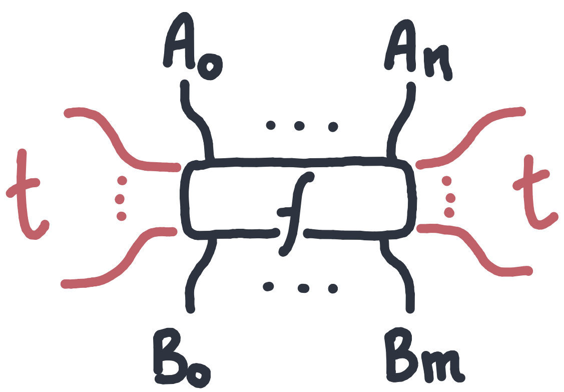
Additionally, we assign a weight to each generator; we define a weighted polygraph,

$$w : \mathcal{G}(X_0, \dots, X_n ; Y_0, \dots, Y_m) \longrightarrow \mathbb{N}.$$

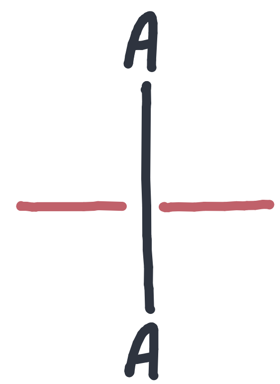
Timing in Double Categories

Timed processes. Time(P). Single-object double category with

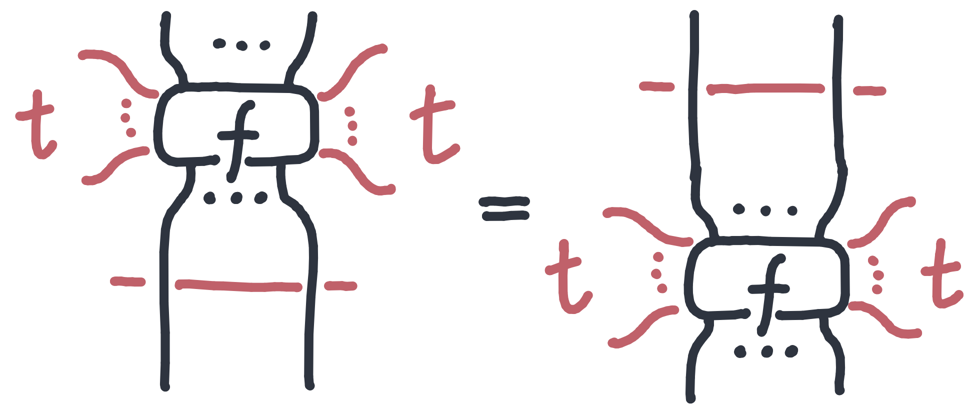
- arrows objects generated by the polygraph;
- proarrows the monoid of the natural numbers;
- cells freely generated by a weighted polygraph, plus 'waiting'.



;



;



;

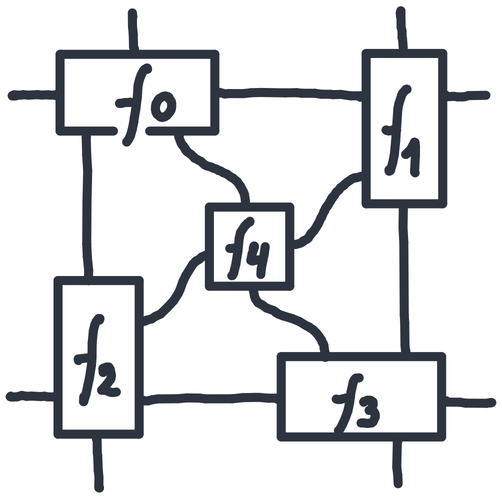
$f: A_0 \otimes \cdots \otimes A_n \to B_0 \otimes \cdots \otimes B_m$ in t units of time

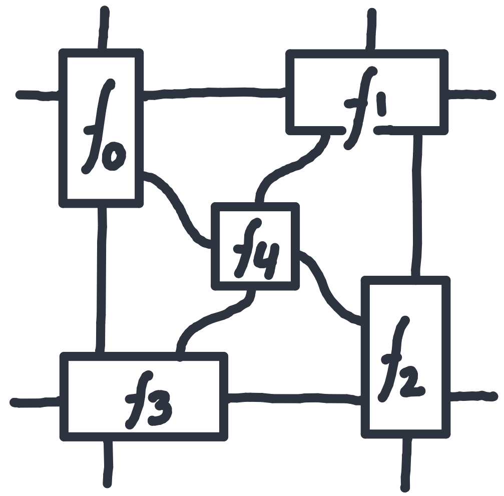
Waiting for 1 unit of time

waiting commutes

Pinwheel Double Categories

The pinwheel and the reverse pinwheel are the only obstacles to the string diagrams for double categories.



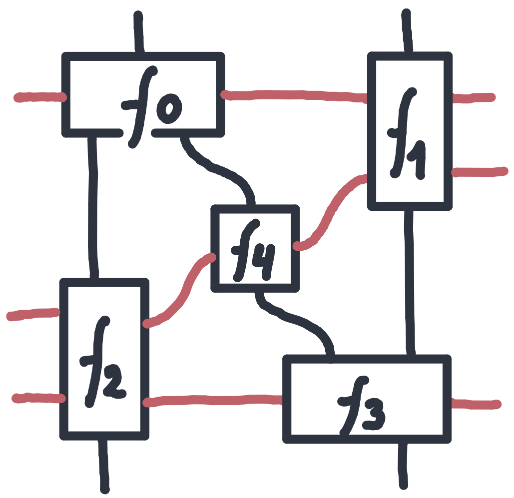


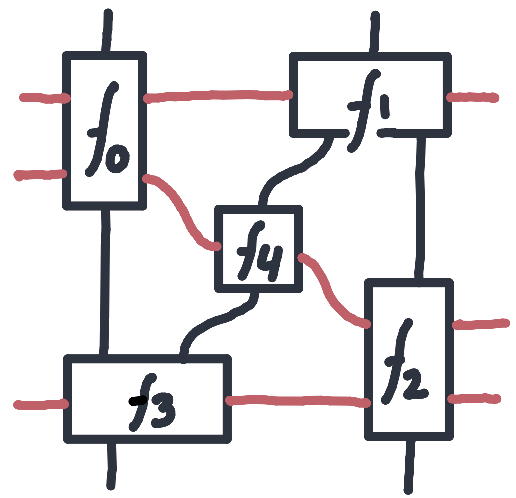
Either define the notion of "double category with pinwheels" or work with bicategories instead.

Delpeuch, Vicary
Dawson

Pinwheel Double Categories

The pinwheel and the reverse pinwheel are the only obstacles to the string diagrams for double categories.





Either define the notion of "double category with pinwheels" or work with bicategories instead.

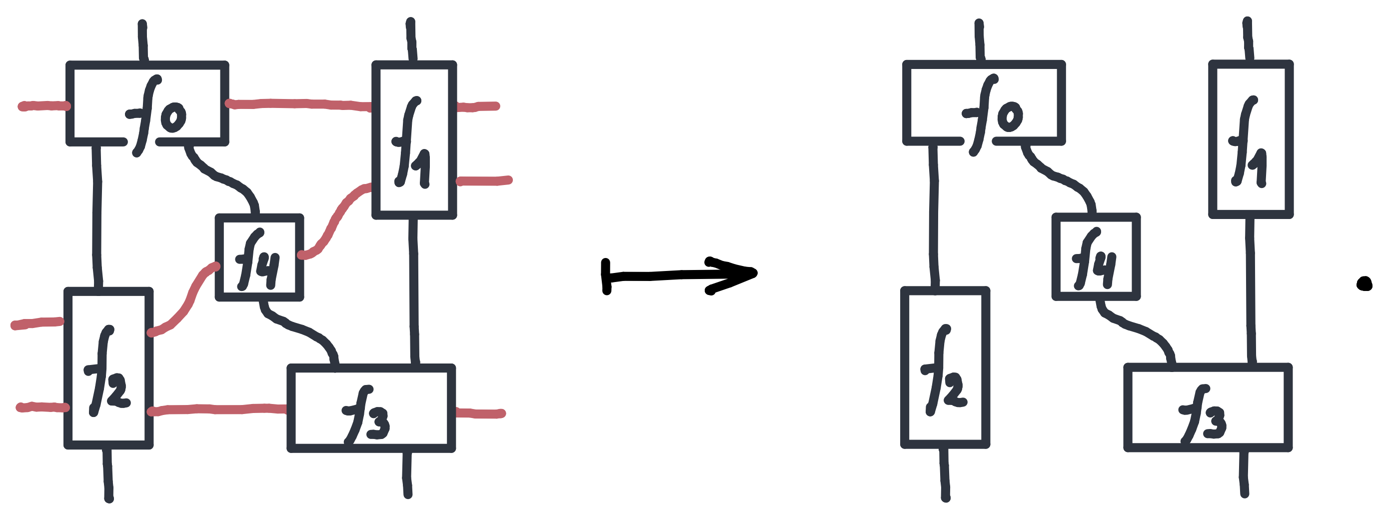
Delpeuch, Vicary
Dawson

These are perfectly valid timings.

Timing in Double Categories

Forgetting about time is a pinwheel double functor; forgetting about everithing but time is also a double functor.

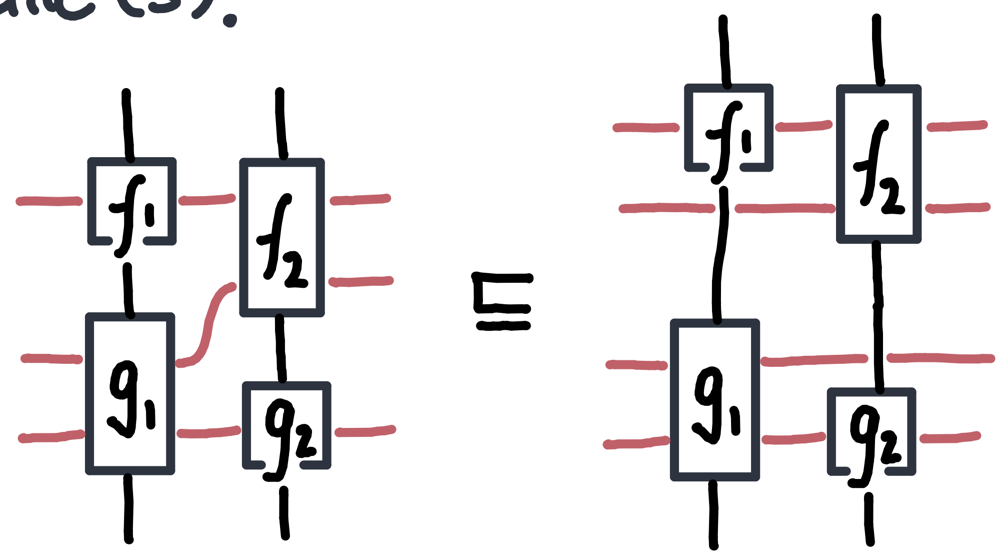
$$\mathrm{morph}: \mathrm{Time}(\mathcal{G}) \longrightarrow \mathrm{Free}(U_{time}\,\mathcal{G})$$
$$\mathrm{time}: \mathrm{Time}(\mathcal{G}) \longrightarrow \mathbb{N}$$



Timing in Double Categories

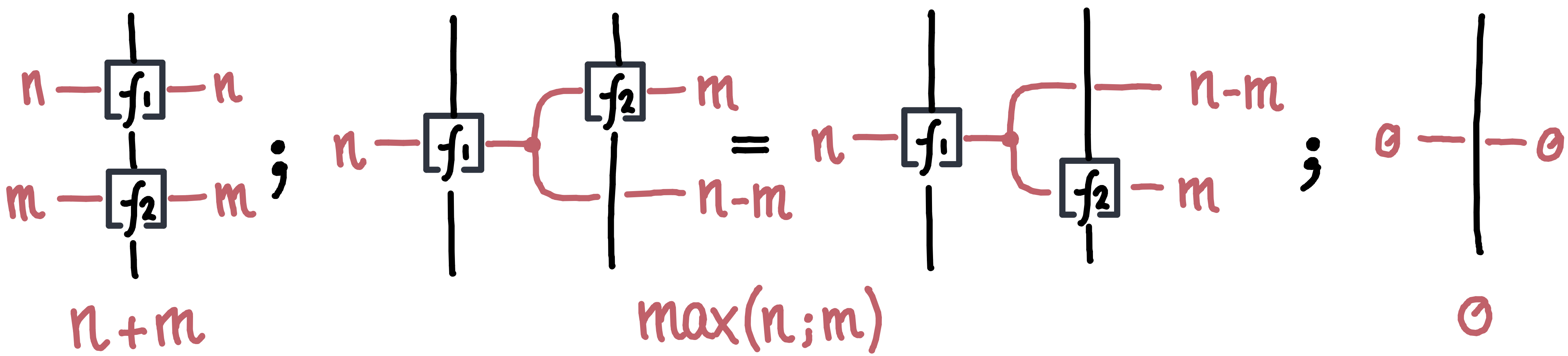
We impose an order relation between **timed processes**: $r \sqsubseteq s$ whenever

- $\mathrm{morph}(r) = \mathrm{morph}(s)$, and
- $\mathrm{time}(r) \leq \mathrm{time}(s)$.



Timing in Double Categories

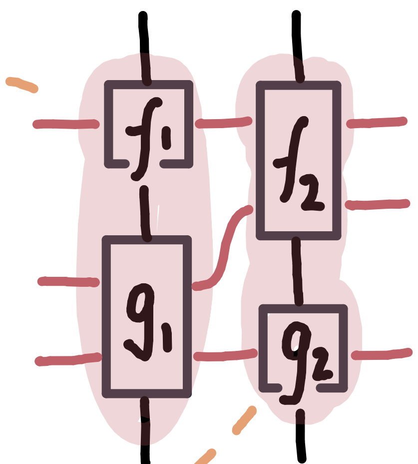
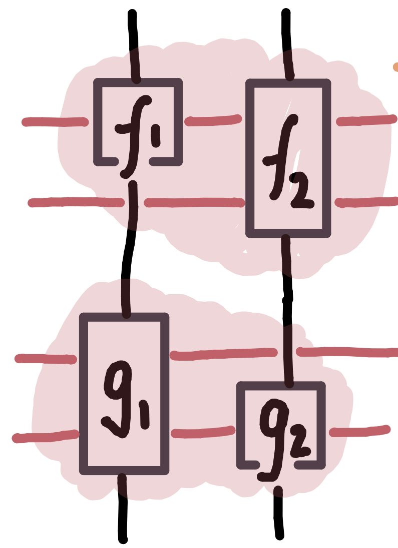
Timed processes of arbitrary time boundary, $\Sigma_{n \in \mathbb{N}} \,\mathrm{Time}(n; {}^{A}_{B}; n)$, almost form a monoidal category.



Sequential Parallel Identity

Timing in Double Categories

The interchange law is only an inequality.

$(f_1 ⨾ g_1) \otimes (f_2 ⨾ g_2) =$  $\sqsubseteq$  $= (f_1 \otimes f_2) ⨾ (g_1 \otimes g_2)$

PART 2 : DUOIDAL

Duoidal Categories

A duoidal category is a category $\mathbb{C}$ with two monoidal structures $(\mathbb{C}, \otimes, I)$ and $(\mathbb{C}, \triangleleft, N)$ such that the latter distributes over the former

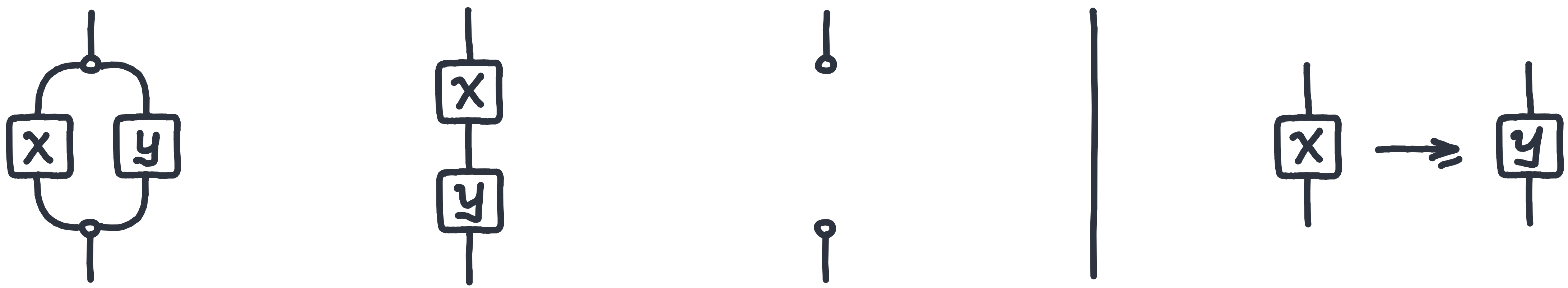
$$
\begin{aligned}
\ell &: (A \triangleleft B) \otimes (C \triangleleft D) \longrightarrow (A \otimes C) \triangleleft (B \otimes D), \\
m &: N \otimes N \longrightarrow N, \\
u &: I \triangleleft I \longrightarrow I, \\
c &: I \longrightarrow N.
\end{aligned}
$$

These are subject to some "coherence conditions" as lax monoidal functors.

- Avoid Eckmann-Hilton by being lax.

Aguiar, Mahajan.

Duoidal Categories, Intuition



$x \otimes y$ — parallel independent composition

$x \triangleleft y$ — sequential dependent composition

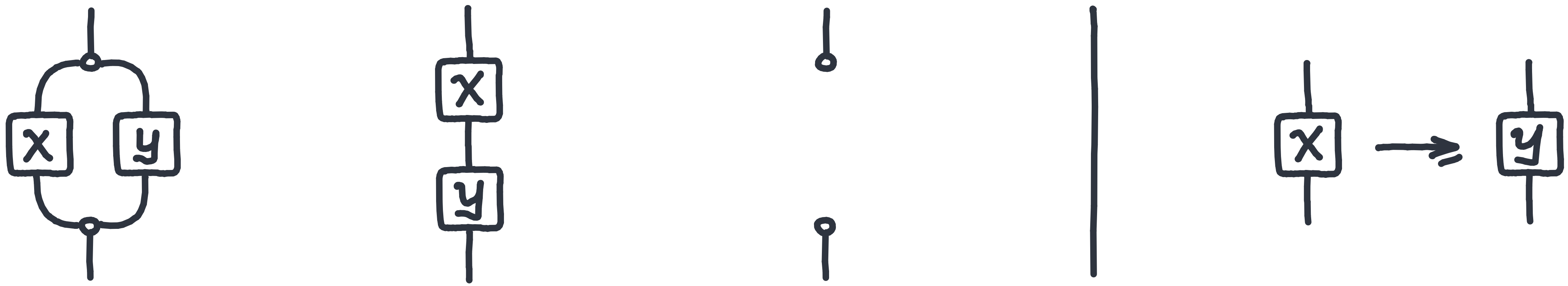
I — interrupt

N — doing nothing

$x \twoheadrightarrow y$ — more dependencies

Duoidals in Adjoint Monoids

THEOREM. The endocells of an adjoint pseudomonoid form a duoidal category with convolution and composition.



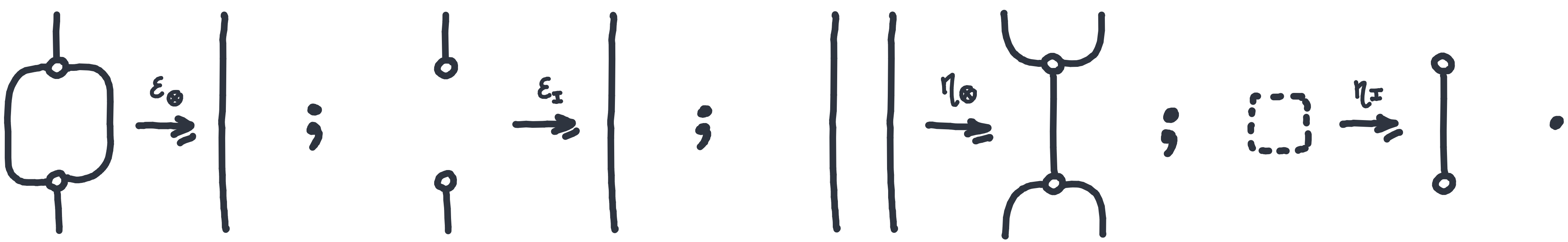
$X \otimes Y$ $\qquad$ $X \triangleleft Y$ $\qquad$ I $\qquad$ N $\qquad$ $X \to Y$

Garner, López Franco

Adjoint Monoids or Map Monoids

In a monoidal bicategory, an adjoint monoid is a monoid-comonoid adjoint pair.

This is to say:

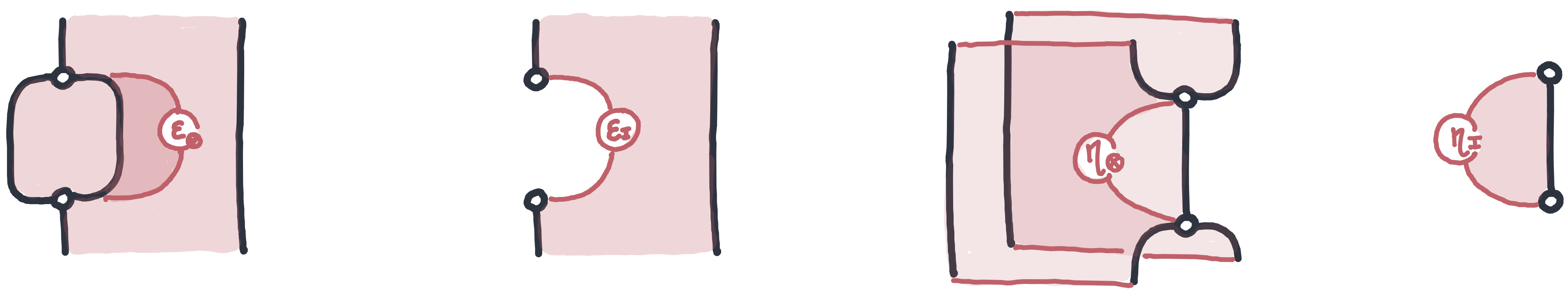


Carboni, Walters

Adjoint Monoids or Map Monoids

In a monoidal bicategory, an adjoint monoid is a monoid-comonoid adjoint pair.

This is to say:



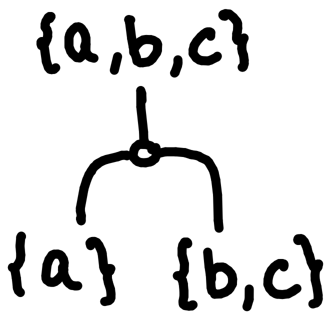
Morehouse   Carboni, Walters

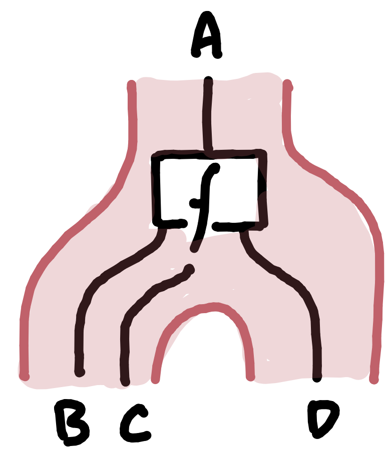
EXAMPLE

Natural numbers with sum and maximum are a posetal duoidal

$$\max(a+c,\, b+d) \;\leq\; \max(a,b) + \max(c,d).$$

Endorelations over a commutative monoid.





Endoprofunctors over a monoidal category.

DUOIDAL CATEGORIES: COHERENCE

Coherence: any two parallel maps constructed out of structure maps and formally typed commute.

In the literature, this is well-known. Appears in print.

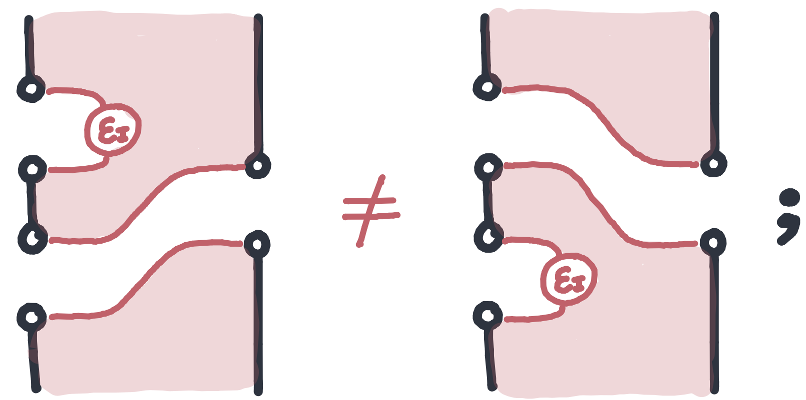
Duoidal Categories: Coherence

Coherence: any two parallel maps constructed out of structure maps and formally typed commute.

In the literature, this is well-known. Appears in print.

PROPOSITION. That coherence does NOT hold.

$$\begin{tikzcd} I \triangleleft I \arrow[r] \arrow[d] \arrow[dr, phantom, "\neq"] & I \triangleleft N \arrow[d] \\ N \triangleleft I \arrow[r] & I \end{tikzcd} \quad ;$$



Duoidal Categories

- We do not need morphisms now, posetal is enough
- We want $(\triangleleft)$ to have a typed composition.
- The duoidal needs to occur between

  $(\otimes)$ and $(\fatsemi) = (\triangleleft)$.

PART 3 : LAXLY MONOIDAL

LAX FUNCTOR

DEFINITION. Let $(\mathbb{C},\leq)$ and $(\mathbb{D},\leq)$ be two posetally-enriched monoidal categories. A lax functor $F : \mathbb{C} \to \mathbb{D}$ is an assignment on objects and morphisms that moreover satisfies

$$F(f) \fatsemi F(g) \leq F(f \fatsemi g) \qquad \text{and} \qquad id \leq F(id).$$

We may consider natural transformations between them.

- Normal lax functors preserve identities.

Walters. Sheaves and Cauchy-complete categories.

Lax-Interchanging Monoidals

Definition. A *laxly interchanging monoidal category* is a locally posetal category $(\mathbb{C}, \geq)$ endowed with two lax functors,

$$(\otimes) : \mathbb{C} \times \mathbb{C} \longrightarrow \mathbb{C} \quad \text{and} \quad I : 1 \longrightarrow \mathbb{C},$$

that, for now, we assume to be strictly associative and unital.

Laxity implies:

$$\otimes(f,g) \mathbin{;} \otimes(f',g') \geq \otimes(f \mathbin{;} f' ; g \mathbin{;} g')$$
$$id \geq \otimes(id, id)$$
$$id_I \mathbin{;} id_I \geq id_I$$
$$id_A \geq id_I$$

Many of these are bureaucratic; we can just pick normal lax functors.

Lax-Interchanging Monoidals

Definition. A *laxly interchanging monoidal category* is a locally posetal category $(\mathbb{C}, \geq)$ endowed with two lax functors,

$$(\otimes) : \mathbb{C} \times \mathbb{C} \to \mathbb{C} \quad \text{and} \quad I : 1 \to \mathbb{C},$$

that, for now, we assume to be strictly associative and unital.

Laxity implies:

$$(f \otimes g) \,;\, (f' \otimes g') \geq (f \,;\, f') \otimes (g \,;\, g')$$
$$id \geq \otimes(id, id)$$
$$id_I \,;\, id_I \geq id_I$$
$$id_A \geq id_I$$

Many of these are bureaucratic; we can just pick normal lax functors.

Future Work

$$
\begin{tikzcd}
\omega\mathrm{PolyGr} \arrow[r, bend left, "\mathrm{Time}"] \arrow[r, phantom, "\perp"] \arrow[d, "U"'] & \mathrm{TimingDouble} \arrow[l, bend left] \arrow[r] \arrow[d, "\mathrm{Vert}"] & \mathrm{LaxIntMon} \\
\mathrm{PolyGr} \arrow[r, bend left] \arrow[r, phantom, "\perp"] & \mathrm{MonCat} \arrow[l, bend left] &
\end{tikzcd}
$$

End