Polar Shuffles

Mario Román inc. j.w.w. Matt Earnshaw, Chad Nester

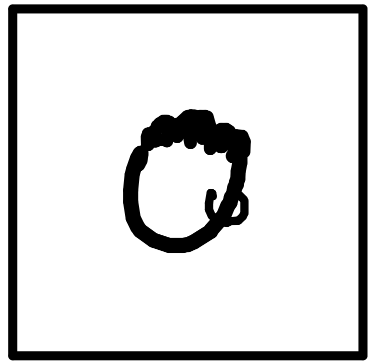
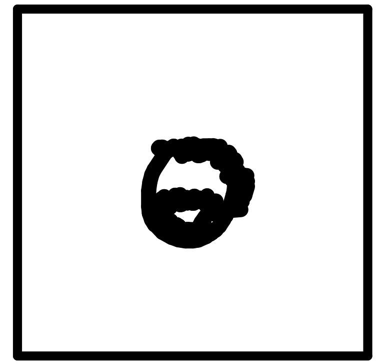
Tallinn University of Technology

March 18th; CHESS.

ERC Blast project.

EU Estonian IT Academy.

Polar Interleavings for Deadlock-Free Message Passing

Matt Earnshaw, Chad Nester, Mario Román

Tallinn University of Technology

University of Oxford

Motivation

What are the fundamental structures of concurrency?

Abramsky.

Motivation

A fundamental structure for message passing: message theories.

1. Message theories can be freely constructed over a sym.mon.cat.

$$\begin{tikzcd} \mathrm{Msg} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \mathrm{SymMonCat} \arrow[l, bend left] \end{tikzcd}$$

2. Message theories are algebras of a universal operad.

   algebras of the freely polarized normal monoidal sym. multicat.

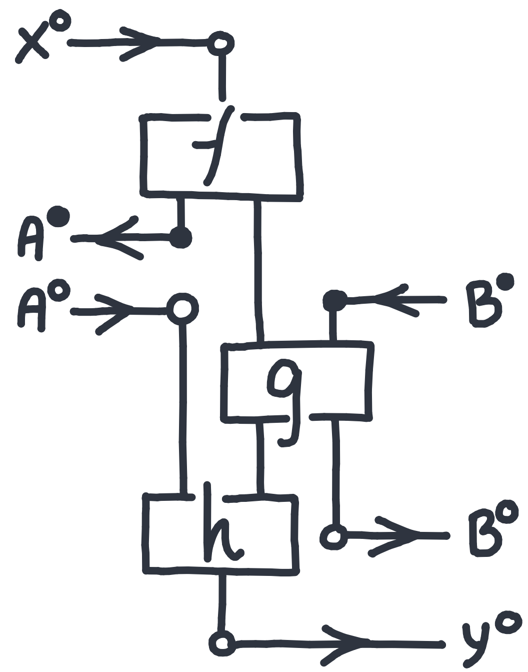
3. Have a concurrency-style internal language.

```
Protocol(request°, login°, get•) {
  Auth(login°, ok•, ack°);
  Serv(ok°, ack•, request°, get•);
}
```

Part 1 : Send/Receive Duality.

CORNERINGS

The horizontal category of the free single object proarrow equipment* represents communicating processes.

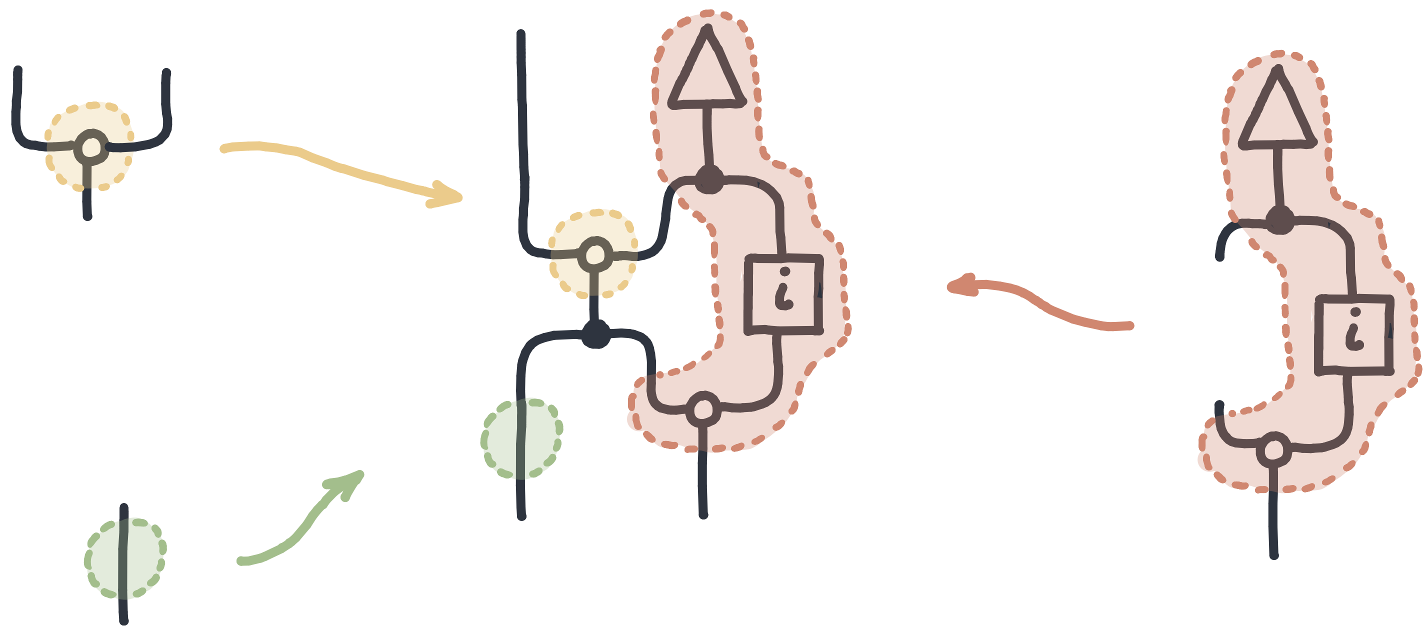


* With pinwheels.

Nester, Voorneveld

CORNERINGS

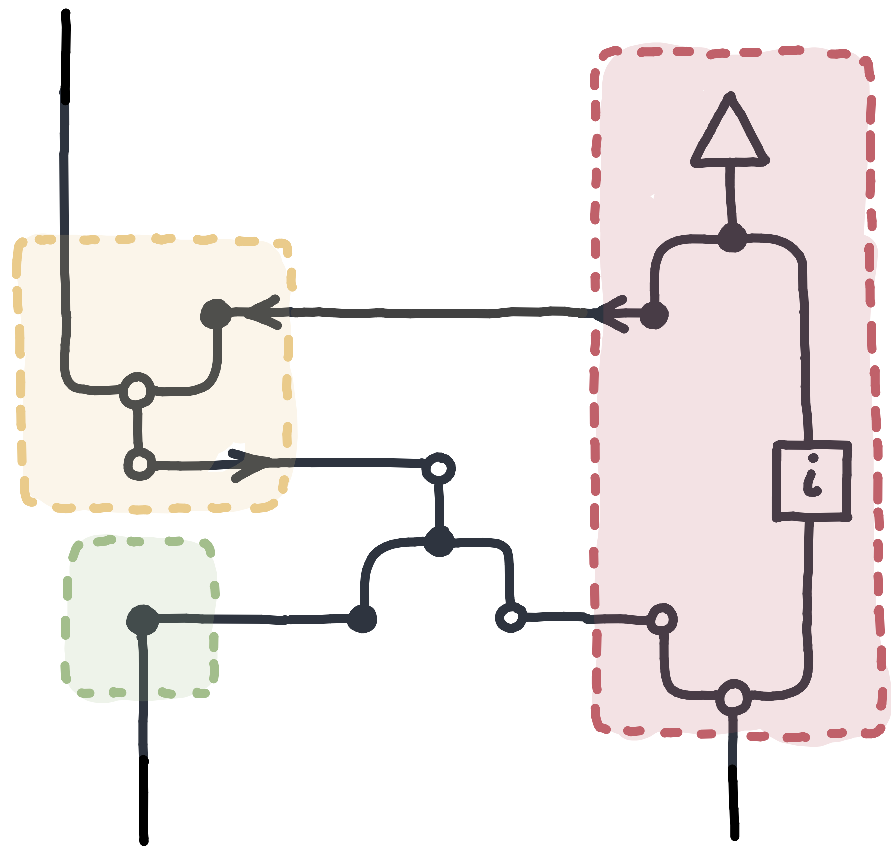
Cornerings allow us to split monoidal morphisms along non-(parallel/sequential) decompositions.



Broadbent, Karvonen.

CORNERINGS

Cornerings allow us to split monoidal morphisms along non-(parallel/sequential) decompositions.

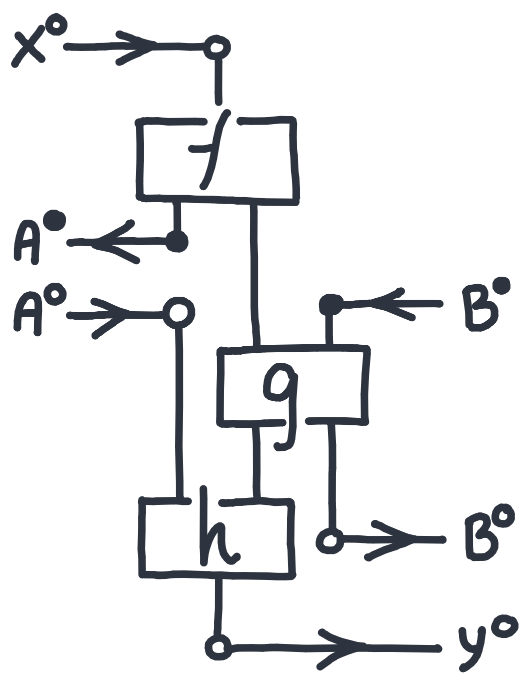


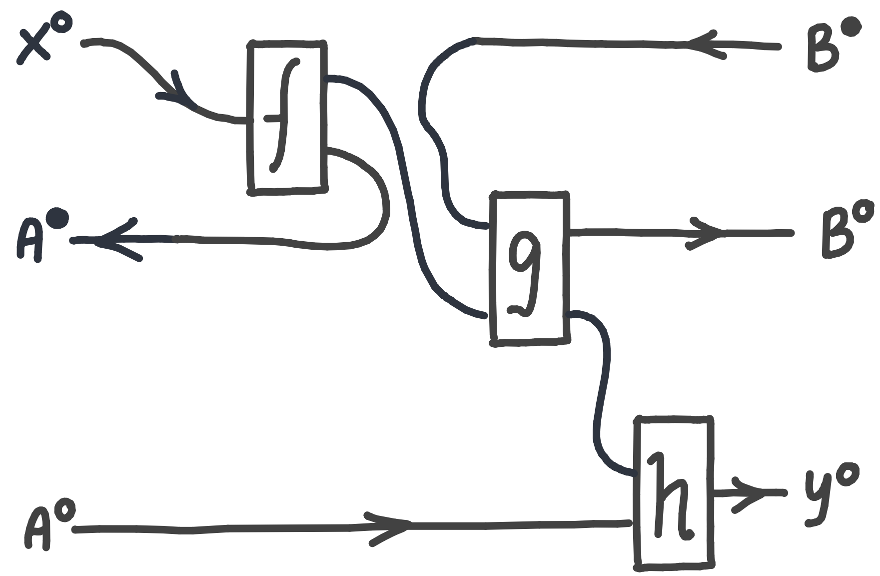
Broadbent, Karvonen.

Nester

SMOOTH CORNERINGS

The horizontal category of the free cornering is the original monoidal category plus freely added dualities.

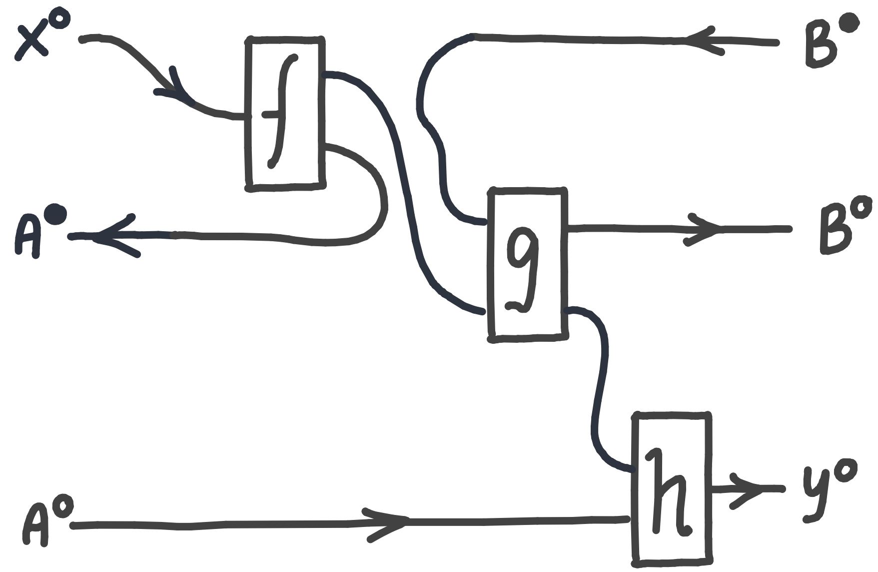




Nester

Smooth Cornerings

The freely polarized monoidal category represents communicating processes.



$$\begin{tikzcd} \mathrm{MonCat}_{str} \arrow[r, bend left, "\mathrm{Polar}"] \arrow[r, phantom, "\perp"] & \mathrm{MonCat}_{str} \arrow[l, bend left, "\mathrm{LAdj}"] \end{tikzcd}$$

"Polarization is left adjoint to taking left adjoints."

Part 2 : Interleaving

Interleaving

Two programs are running concurrently: in how many ways can they interleave?

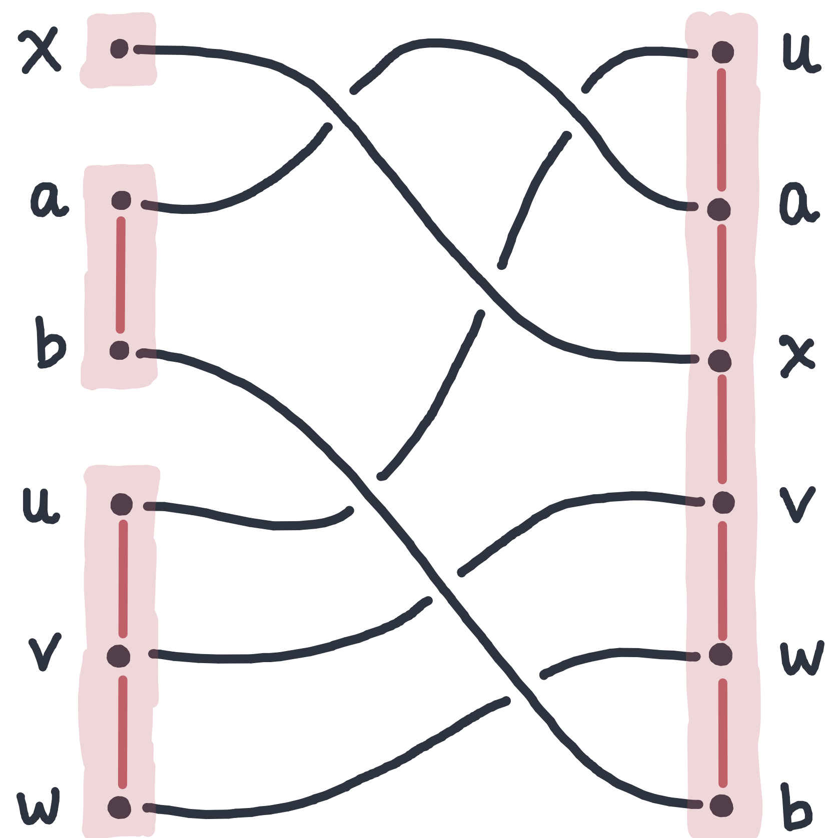
```
alice = do
  x ;
  y ;
  z ;
```

```
bob = do
  p ;
  q ;
```

xyzpq	xpqyz
xypzq	pxyzq
xypqz	pxyqz
xpyzq	pxqyz
xpyqz	pqxyz

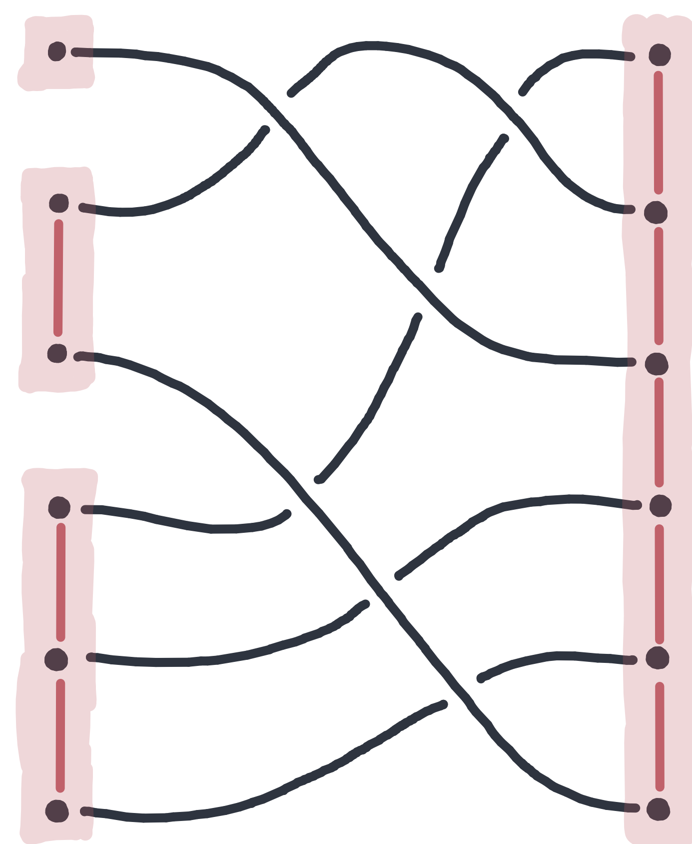
Eilenberg, MacLane.

Shuffles



Mix some words, preserving the relative order inside the words.

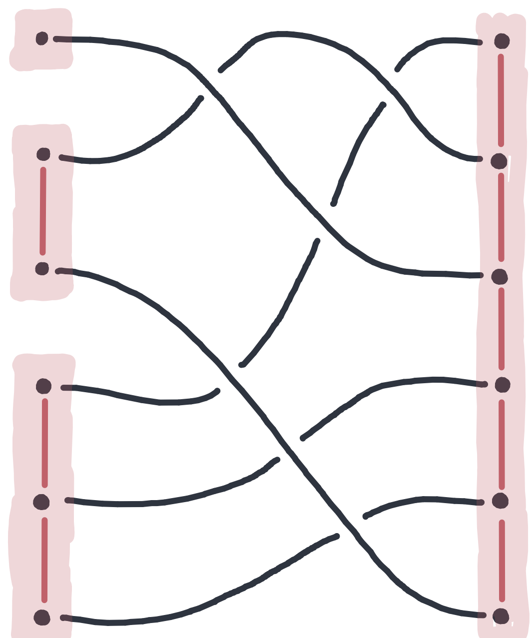
SHUFFLES

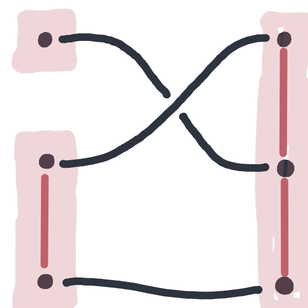


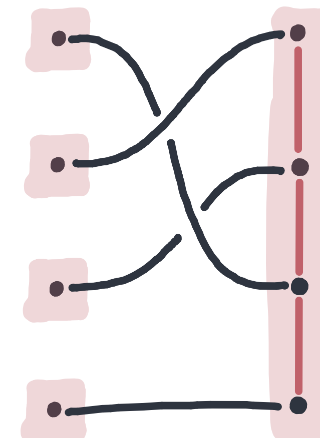
Invariant to renaming, defined up to α-equivalence.

SHUFFLES

$$\mathrm{Shuf}(n_1, \ldots, n_k) = \frac{(n_1 + \ldots + n_k)!}{n_1! \cdots n_k!} \ ;$$







Shuffles are a rich combinatorial structure.

PART 3 : Interlude - Duoidals

Duoidal Categories

Duoidal categories have two tensors and one distributes over the other. We interpret

- $X \triangleleft Y$, sequential tensor, "X, and then Y";
- $X \otimes Y$, parallel tensor, "X and Y at the same time";

$$(X \triangleleft Y) \otimes (U \triangleleft V) \longrightarrow (X \otimes U) \triangleleft (Y \otimes V).$$

When the unit distributor is an isomorphism, $I \xrightarrow{\sim} N$, it is normal. A normal, $\otimes$-symmetric duoidal, is a physical duoidal.

Aguiar & Mahajan '10, Shapiro & Spivak '23.

Duoidal Categories

Duoidal categories are not coherent, there are two formal maps

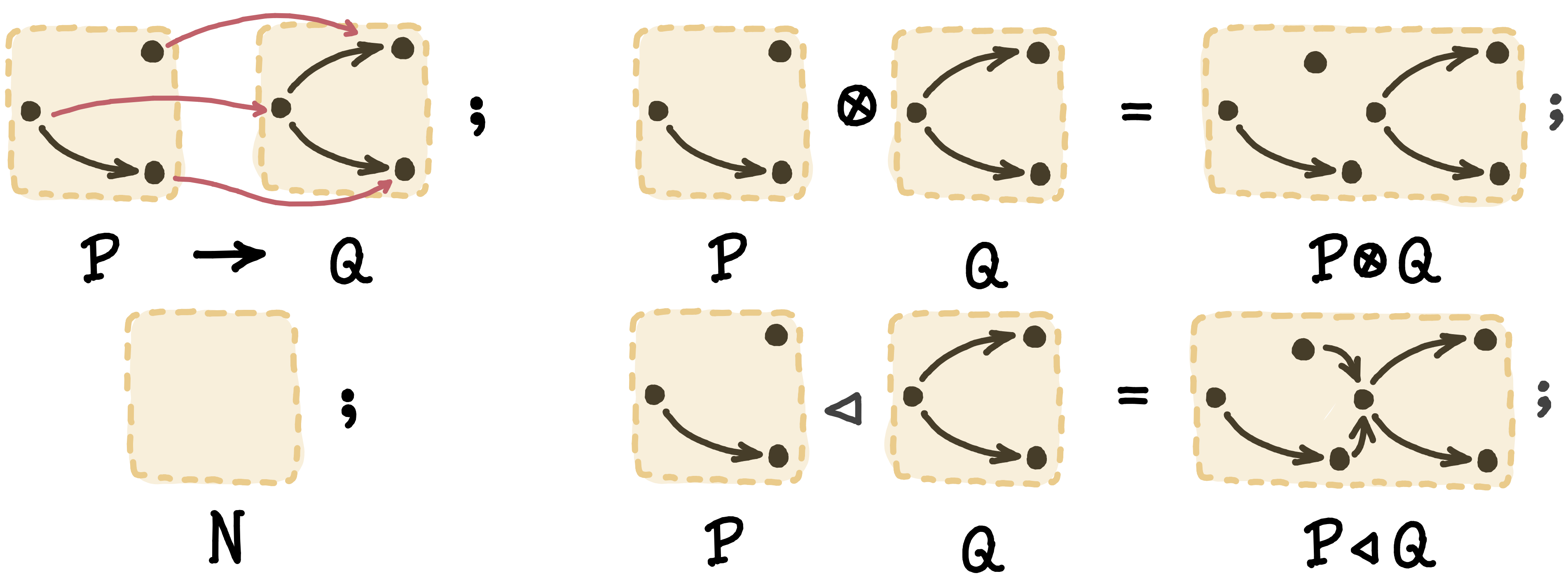
$$I \triangleleft I \longrightarrow I.$$

However, physical duoidal categories are coherent. There is at most a single morphism between any two objects where every type appears exactly once with each variance.

$$\begin{tikzcd} A \otimes (B \triangleleft C) \arrow[r, bend left] \arrow[r, bend right] \arrow[r, phantom, "="] & (A \otimes B) \triangleleft C. \end{tikzcd}$$

Physical Duoidal Categories

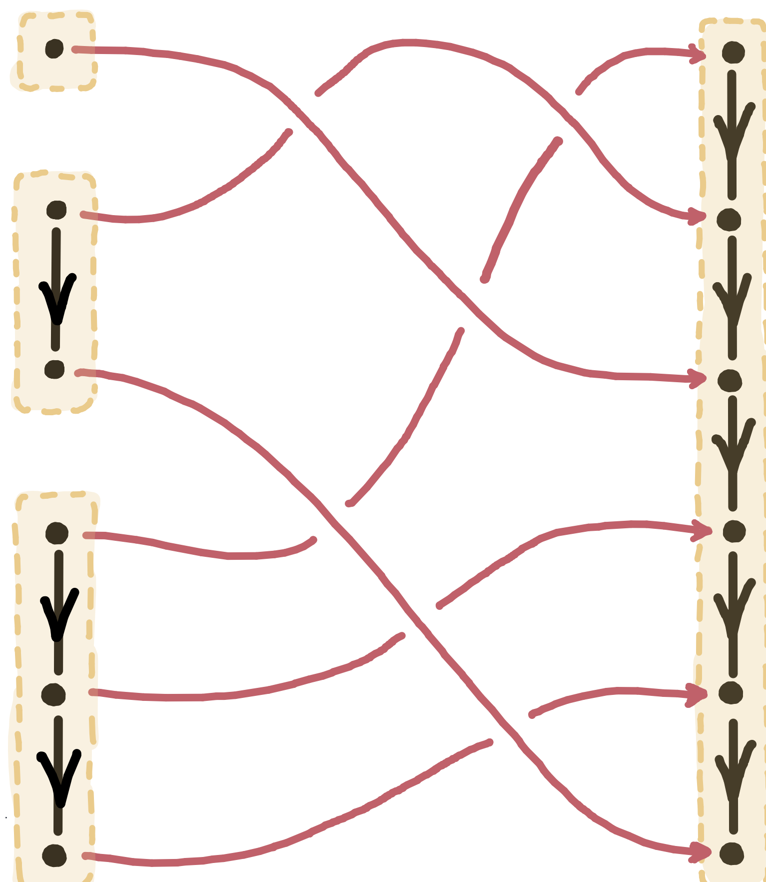
The free physical duoidal category over a single object is fully-faith. into posets and bijective-on-objects inclusions.



Grabowski '81, Gischer '88, Shapiro & Spivak '23

Physical Monoidal Multicategories

When only sequential composition is representable, we recover shuffles as poset inclusions.



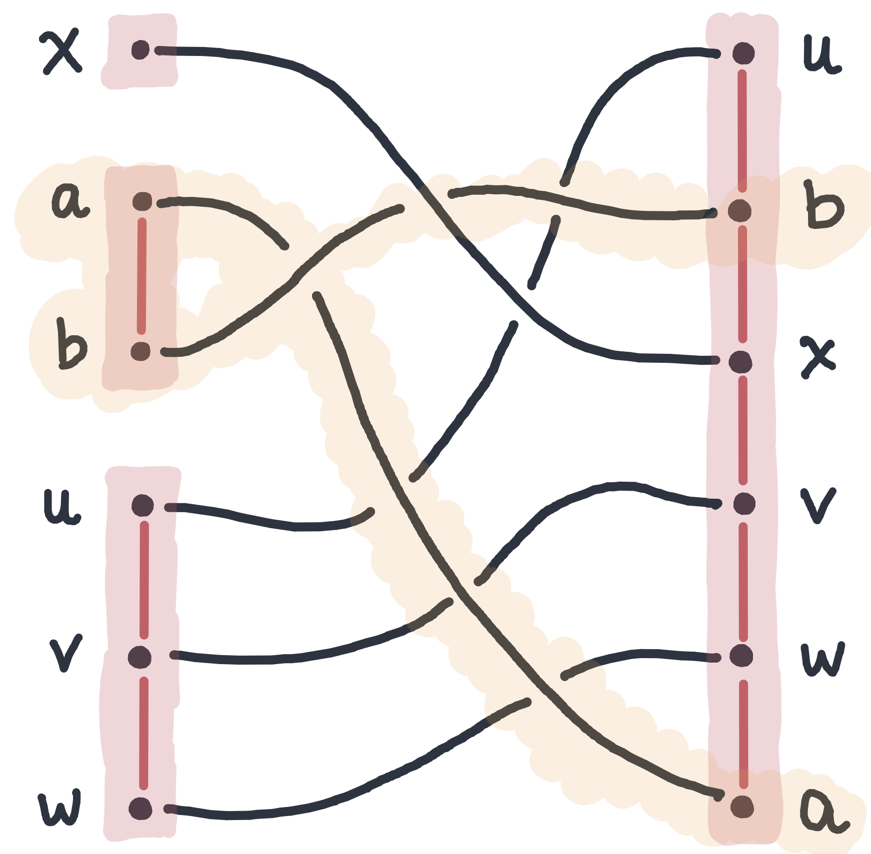
Theorem. Shuffles form the free physical monoidal multicategory over a single object.

Part 4: Polar Shuffles

Send
Receive

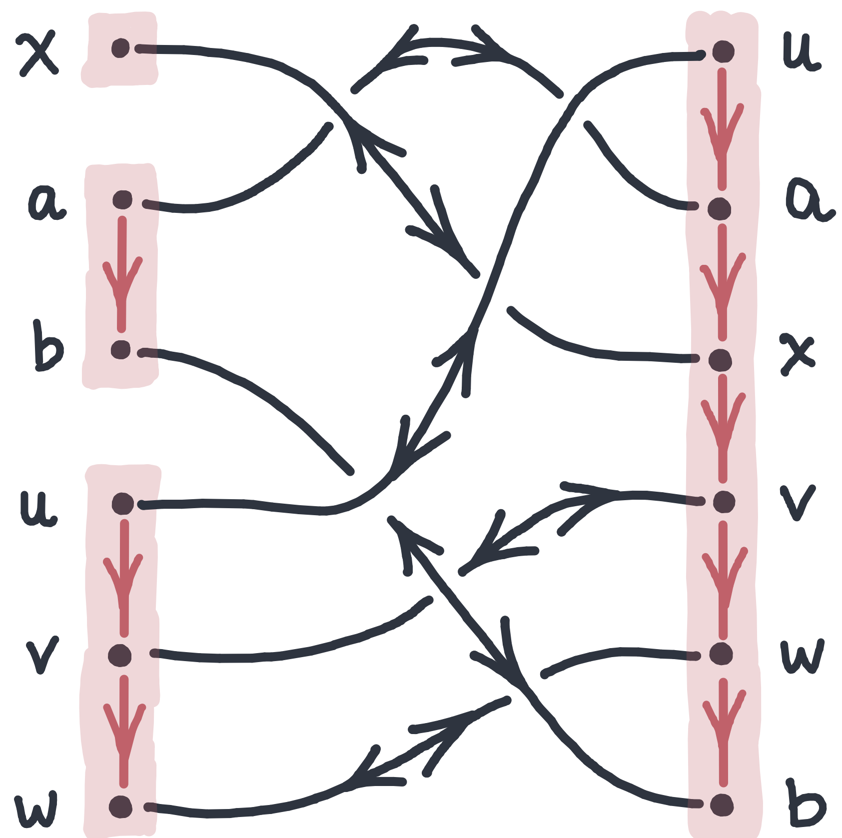
Interleave

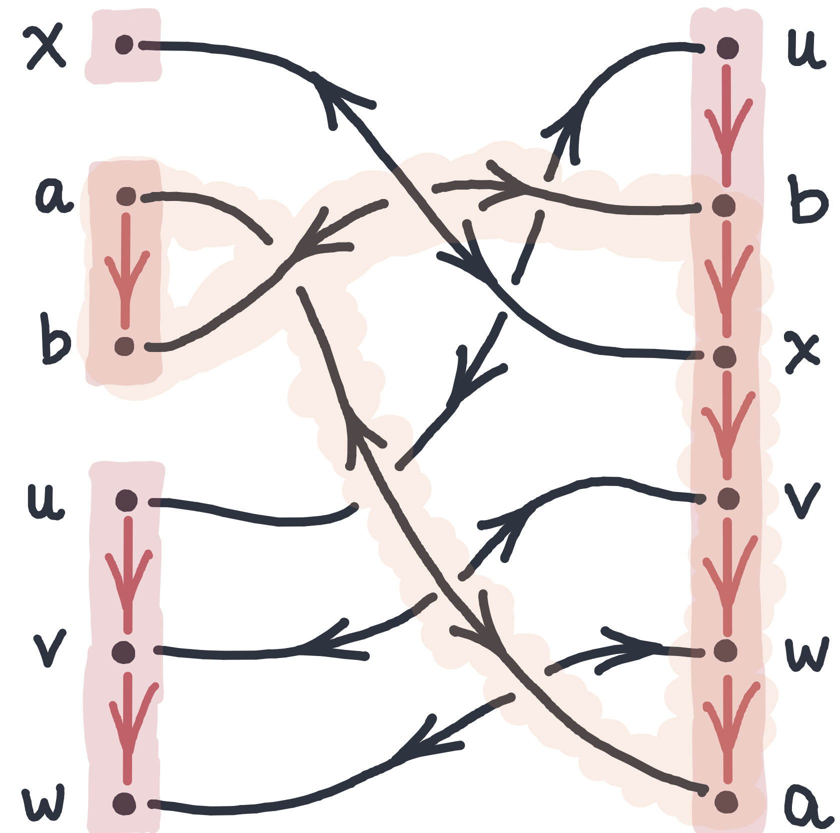
Shuffles



This is not a shuffle: "a" and "b" change order.

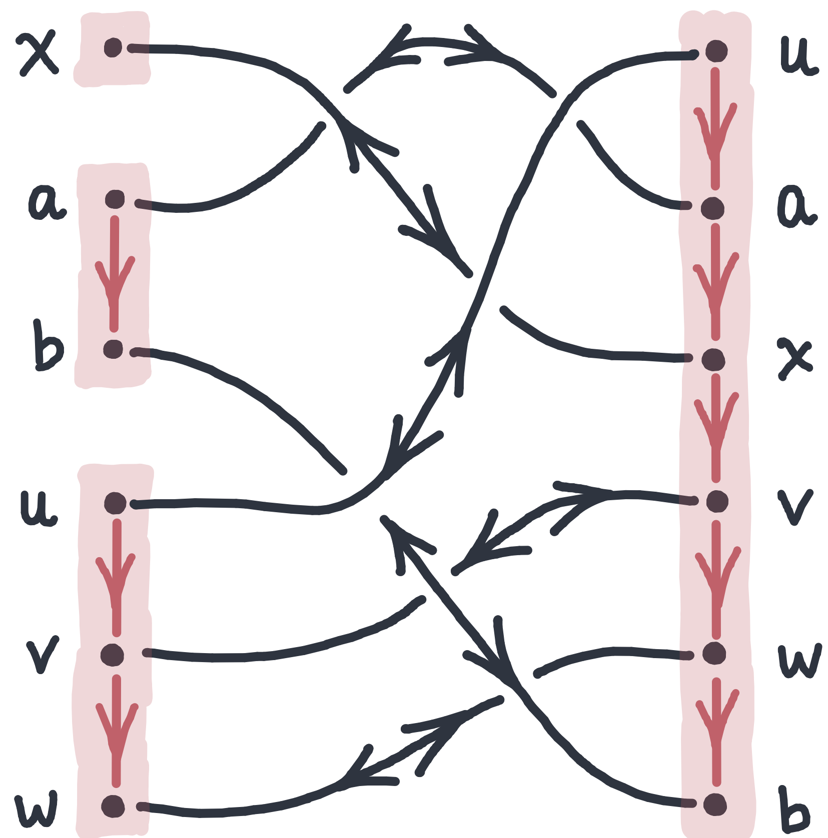
SHUFFLES





Valid shuffles are acyclic. Invalid shuffles contain cycles.

SHUFFLES



DEFINITION. A shuffle is a bijection

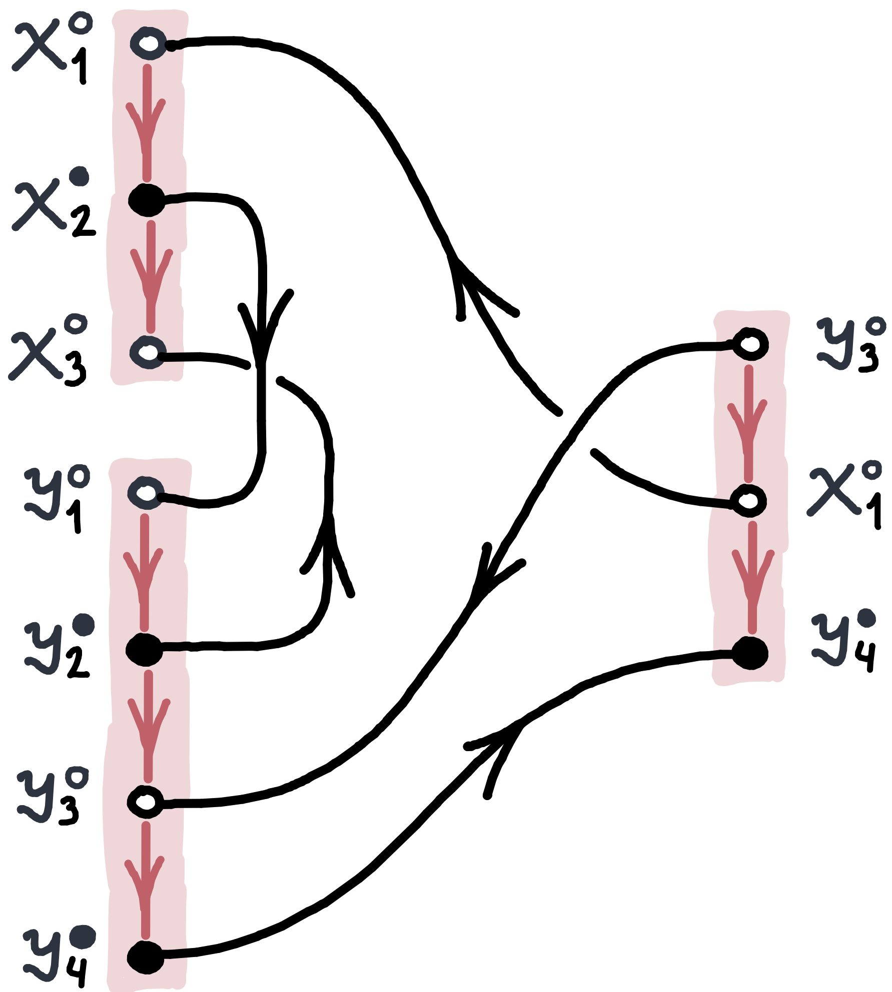
$$f : A_1 + \cdots + A_n \longrightarrow X$$

such that the edges from relative orders ($\downarrow$) and the edges from the bijection

$$(x \bullet \!\!\longrightarrow\!\! \bullet f(x)) \qquad (f(x) \bullet \!\!\longrightarrow\!\! \bullet\, x)$$

form an acyclic graph.

Polar Shuffles



DEFINITION. Polar shuffles are bijections

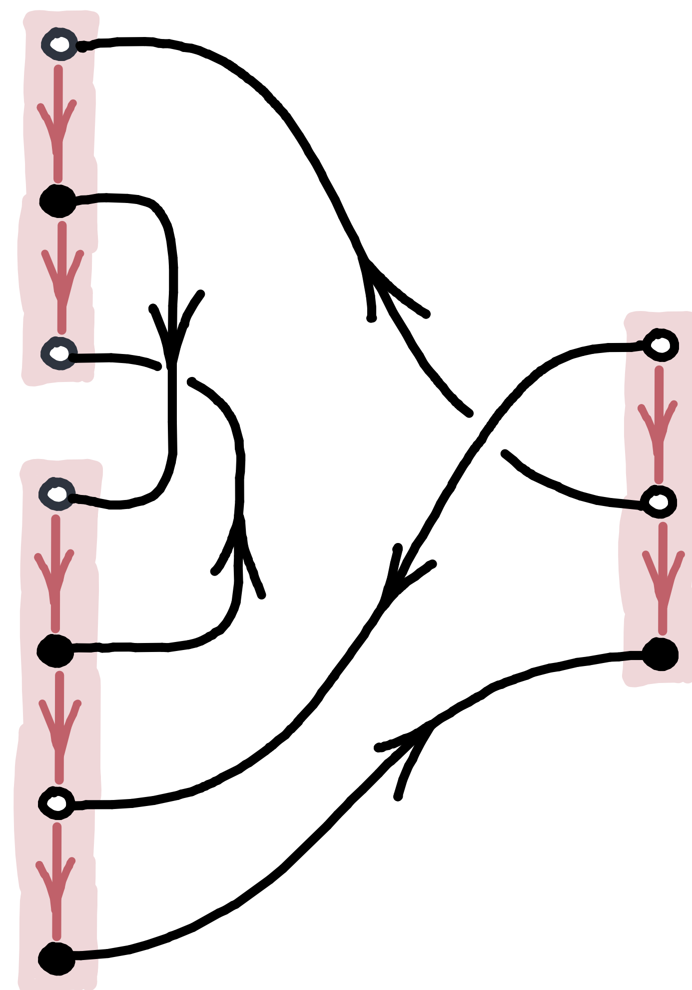
$$A_1^\bullet + \cdots + A_n^\bullet + X^\circ \longrightarrow A_0^\circ + \cdots + A_n^\circ + X^\bullet$$

such that the edges from relative orders (↓) and the edges from the bijection

$$(x \bullet \!\!\longrightarrow\!\! \bullet f(x))$$

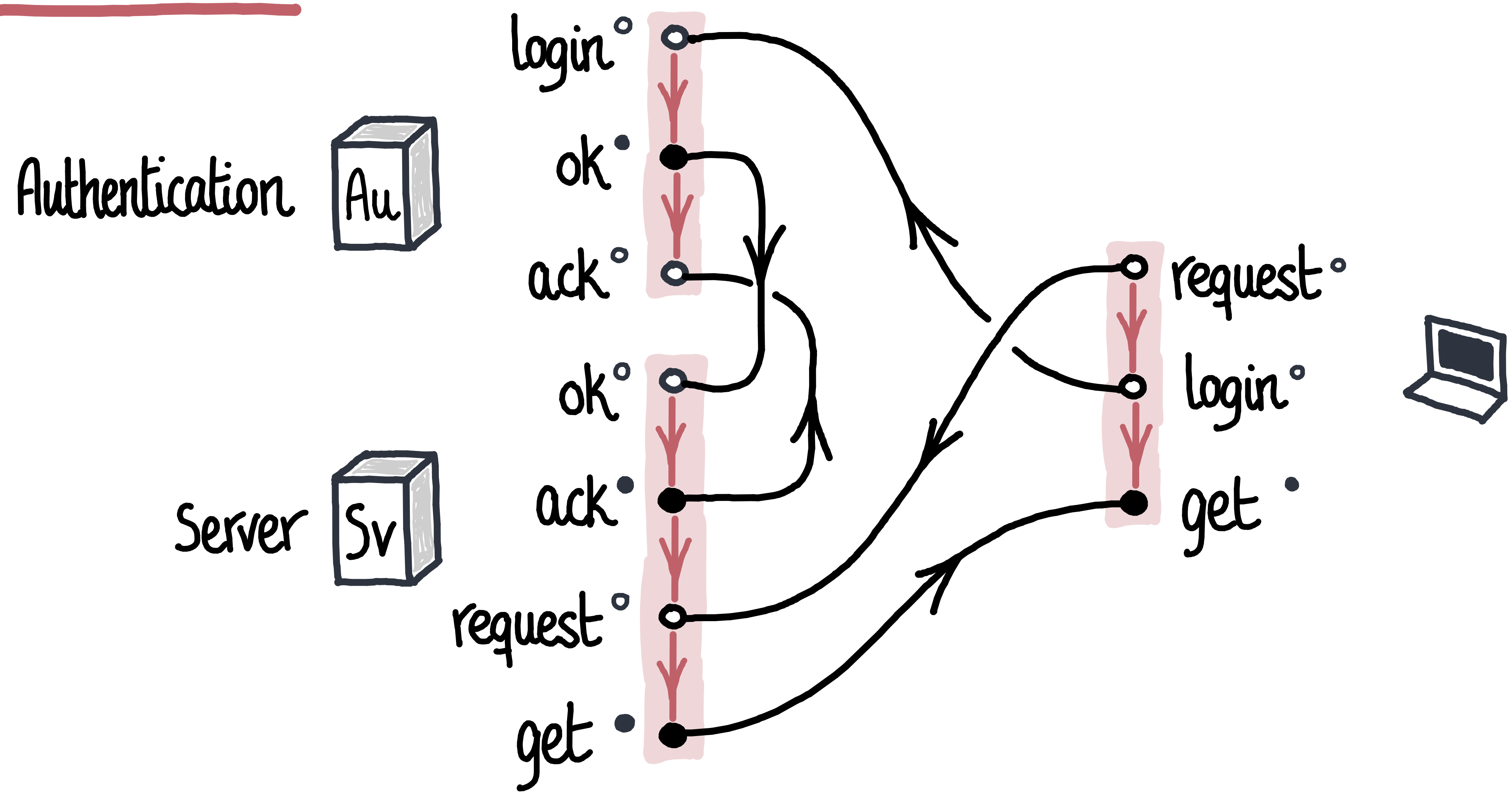
induce an acyclic graph.

Polar Shuffles



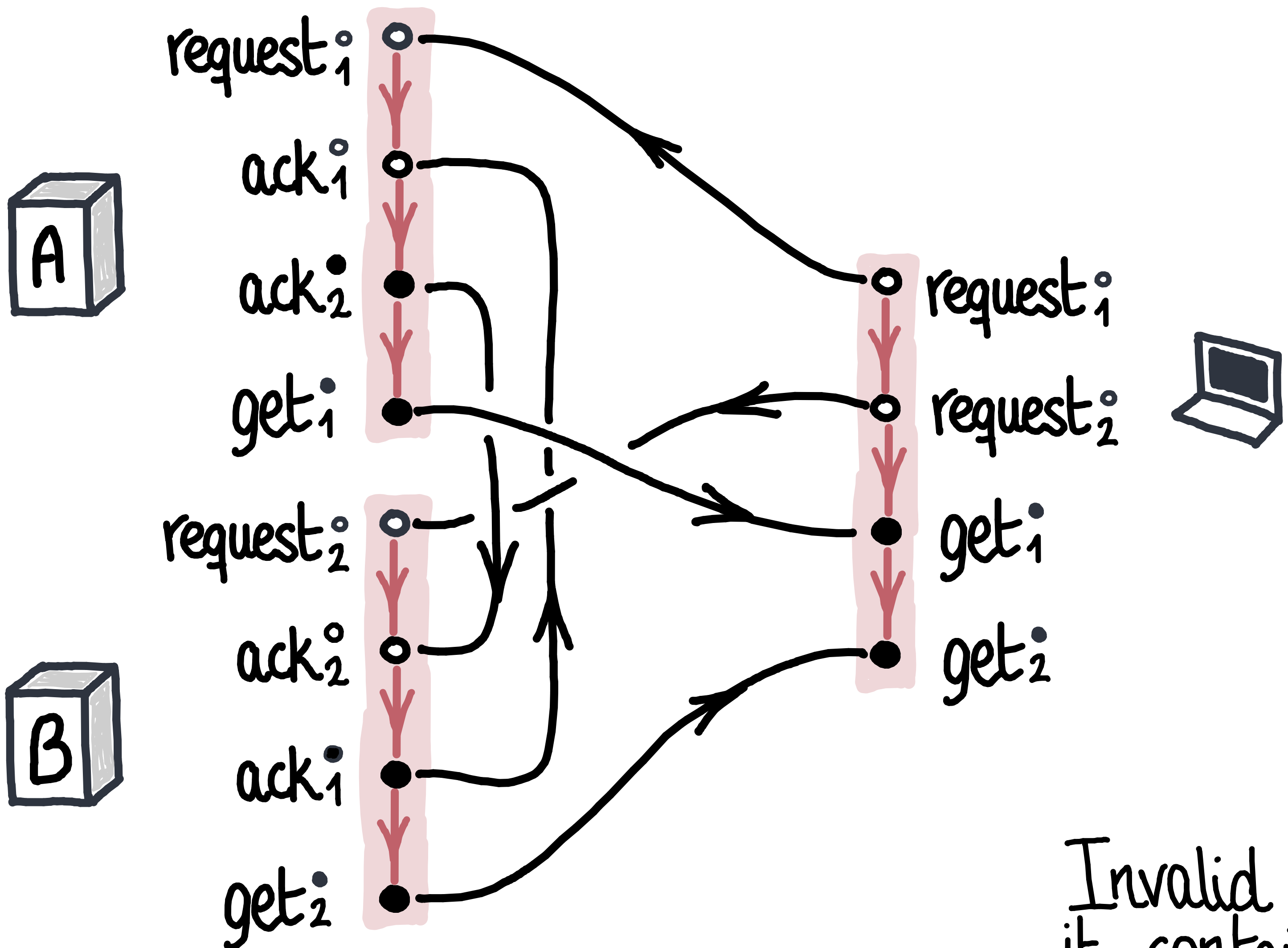
Polar shuffles mix programs that communicate sending• and receiving°.

Polar Shuffles



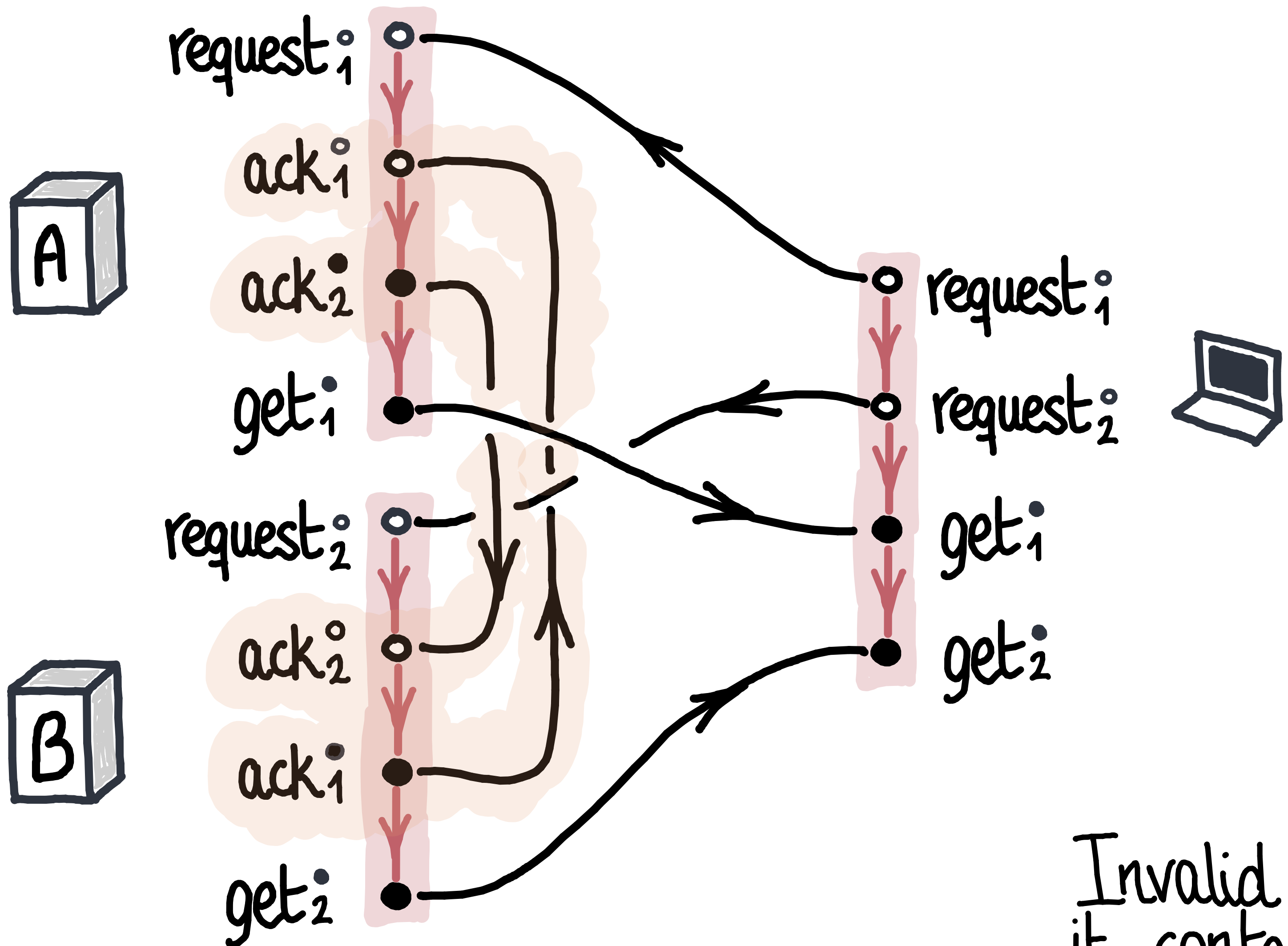
Each polar shuffle determines a way programs could interleave their messages.

Polar Shuffles



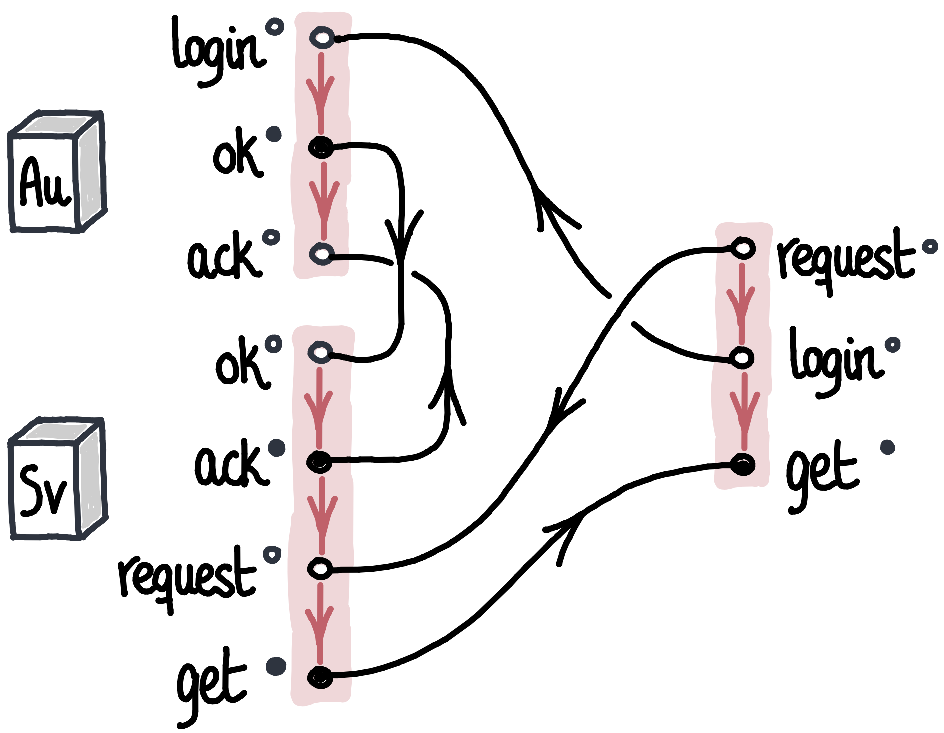
Invalid if and only if it contains a deadlock.

Polar Shuffles



Invalid if and only if it contains a deadlock.

Polar Shuffles




```
Protocol(request°, login°, get•) {
  Auth(login°, ok•, ack°);
  Serv(ok°, ack•, request°, get•);
}
```

Part 5: Message Theories

Message Theories

Set of types, representing resource types: $X, Y, Z, \dots$

Two actions for each resource: send and receive,

$X^{\bullet}$ means "send X".
X° means "receive X".

Lists of actions represent sequencing of the actions.

$\Gamma = X^{\circ}, Y^{\bullet}, Z^{\bullet}, W^{\circ}$ means "ask for X; send Y and then Z;
finally, receive W".

Message Theories, a monoidal multicategory.

$$\frac{}{\varepsilon}\,\textsc{nop} \qquad \frac{}{X^\circ, X^\bullet}\,\textsc{echo} \qquad \frac{\Gamma, X^\bullet, X^\circ, \Delta}{\Gamma, \Delta}\,\textsc{lnk} \qquad \frac{\Gamma_1 \quad \ldots \quad \Gamma_n}{[\Gamma_1, \ldots, \Gamma_n]_\sigma}\,\textsc{shf}_\sigma$$

1. Doing nothing is a session.
2. We can create a receive-send "echo" session.
3. We can link a Send to a Receive port.
4. Events can be interleaved in any order.

MESSAGE THEORIES

DEFINITION. A message theory $\mathbb{M}$ consists of a set of types, $\mathbb{M}_{obj}$, and, for each list of polarized types, a set "of sessions" typed by that list.

$$\mathbb{M}(X_1^{\ominus_1}, \dots, X_n^{\ominus_n}), \text{ for any list of objects } X_i \in \mathbb{M}_{obj} \text{ and } \ominus_i \in \{\circ, \bullet\}.$$

A message theory must contain (reasonably axiomatized) operations for

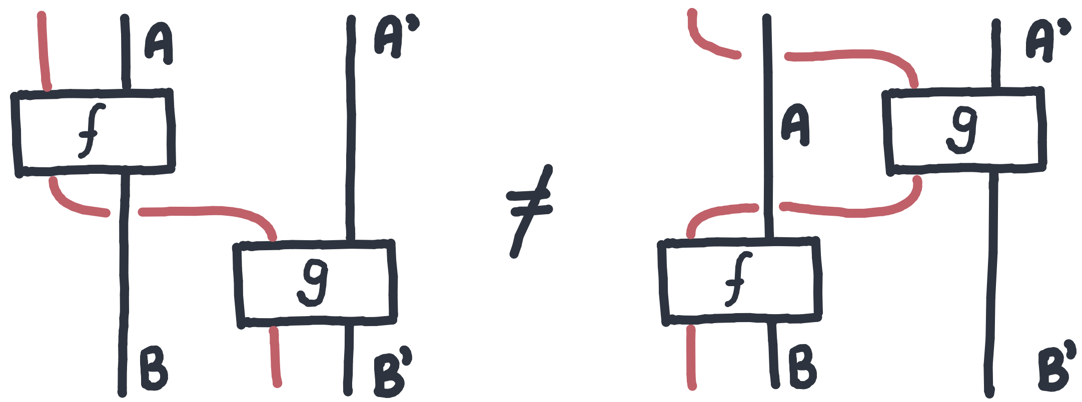
i. binary shuffling, $\textsc{shf}_\sigma \colon \mathbb{M}(\Gamma) \times \mathbb{M}(\Delta) \to \mathbb{M}(\sigma(\Gamma, \Delta))$;
ii. a no-operation, $\textsc{nop} \colon \mathbb{M}()$;
iii. a receive to send channel, $\textsc{spw}_X^\Gamma \colon \mathbb{M}(\Gamma, \Delta) \to \mathbb{M}(\Gamma, X^\circ, X^\bullet, \Delta)$;
iv. linking a send to receive channel, $\textsc{lnk}^\Gamma \colon \mathbb{M}(\Gamma, X^\bullet, X^\circ, \Delta) \to \mathbb{M}(\Gamma, \Delta)$.

THM. Message theories are algebras for the monoidal operad of shuffles.

Part 6: Process/Message Adj.

PreMonoidal Categories

Premonoidal categories extend monoidal categories with effects.



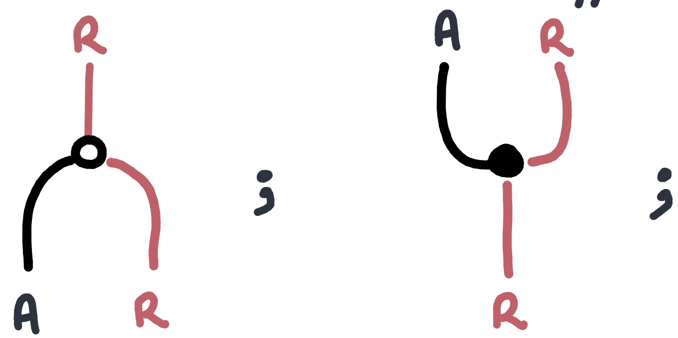
Failure of Interchange

THEOREM. String diagrams with runtime are the internal language of premonoidal categories.

Jeffrey Román

Message Theories vs Process Theories

The free message theory contains string diagrams extended with "send" and "receive" effects.



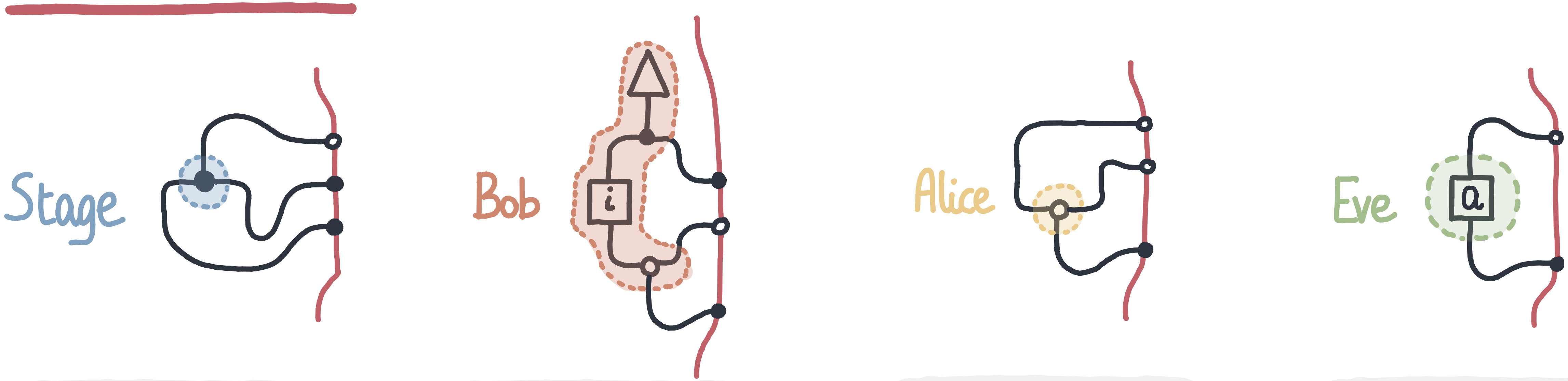
The cofree process theory on a message theory has as morphisms the "receive-then-send" sessions:

$$M(X^\circ, Y^\bullet).$$

THEOREM.

$$\begin{tikzcd} \mathsf{SymMon} \arrow[r, bend left, "\text{Sessions}"] \arrow[r, phantom, "\perp"] & \mathsf{Msg} \arrow[l, bend left, "\text{Send-Receive}"] \end{tikzcd}$$

String Diagrams + Send/Receive




```
stage = do
    x ← get
    send(x)
    send(x)
```

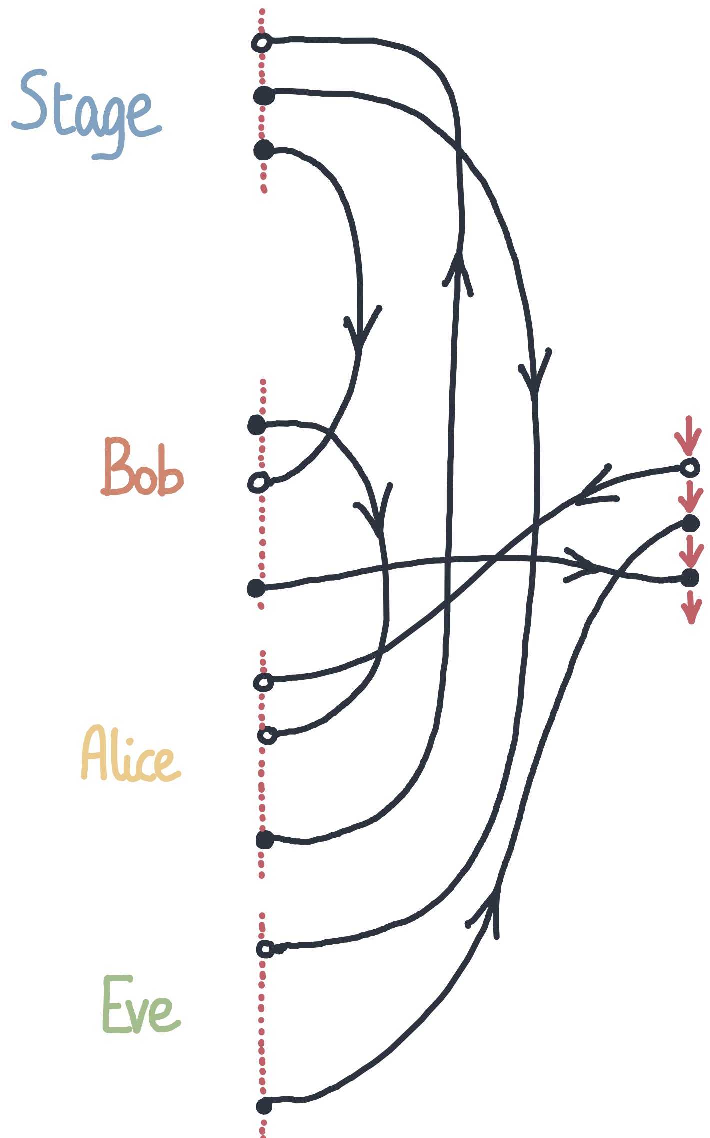
```
bob = do
    k ← random
    send(k)
    c ← get
    send(i(k)⊕c)
```

```
alice = do
    m ← get
    k ← get
    send(m⊕k)
```

```
eve = do
    c ← get
    attack(c)
```

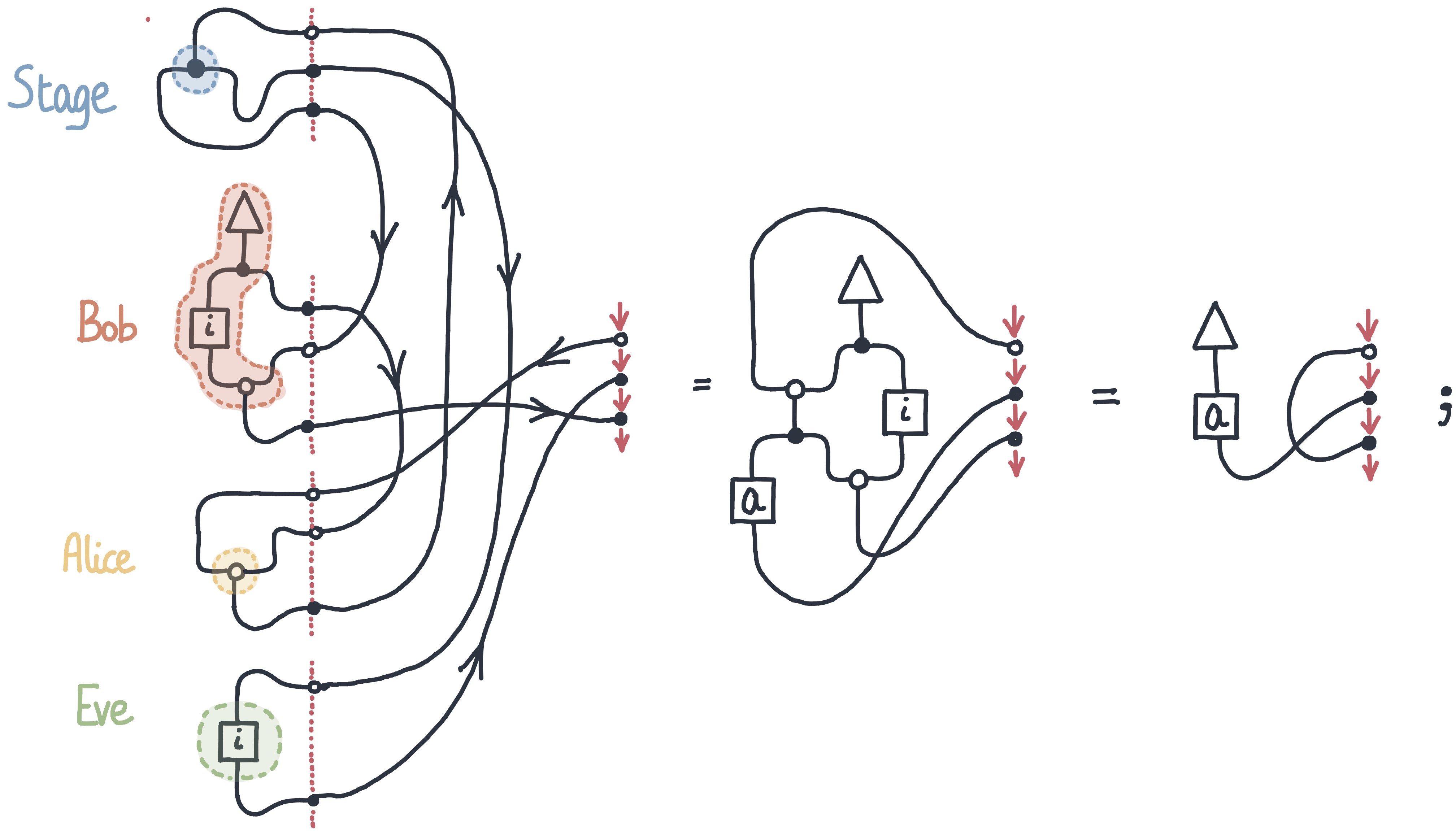
We split a protocol into multiple agents.

Polar Shuffles for Protocols



$$
\begin{array}{l}
\mathrm{otp}\,(\mathrm{msg}^{\circ},\ \mathrm{attack}^{\bullet},\ \mathrm{decrypt}^{\bullet})\ \{\\
\quad \mathrm{stage}\,(\mathrm{crypt}^{\circ}, \mathrm{crypt}_1^{\bullet}, \mathrm{crypt}_2^{\bullet});\\
\quad \mathrm{bob}\,(\mathrm{key}^{\bullet},\ \mathrm{crypt}_1^{\circ},\ \mathrm{decrypt}^{\bullet});\\
\quad \mathrm{alice}\,(\mathrm{msg}^{\circ},\ \mathrm{key}^{\circ},\ \mathrm{crypt}^{\bullet});\\
\quad \mathrm{eve}\,(\mathrm{crypt}_2^{\circ},\ \mathrm{attack}^{\bullet});\ \}
\end{array}
$$

Polar Shuffles for Protocols



Polar Shuffles for Protocols

```
stage = do
    x ← get
    send(x)
    send(x)
```

```
eve = do
    c ← get
    attack(c)
```

```
bob = do
    k ← random
    send(k)
    c ← get
    send(i(k)⊕c)
```

```
alice = do
    m ← get
    k ← get
    send(m⊕k)
```

```
otp (msg°, attack•, decrypt•) {
    stage(crypt°, crypt_1•, crypt_2•);
    bob(key•, crypt_1°, decrypt•);
    alice(msg°, key°, crypt•);
    eve(crypt_2°, attack•); }
```

MOTIVATION

A fundamental structure for message passing: **message theories.**

1. Message theories can be freely constructed over a sym. mon. cat.

$$\begin{tikzcd} \mathsf{Msg} \arrow[r, bend left] \arrow[r, phantom, "\top"] & \mathsf{SymMonCat} \arrow[l, bend left] \end{tikzcd}$$

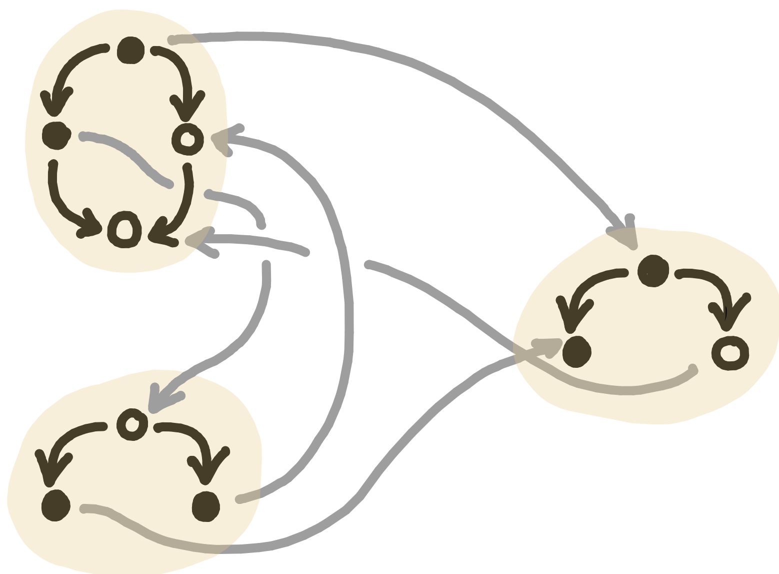
2. Message theories are algebras of a universal operad.

   algebras of the freely polarized normal monoidal sym. multicat.

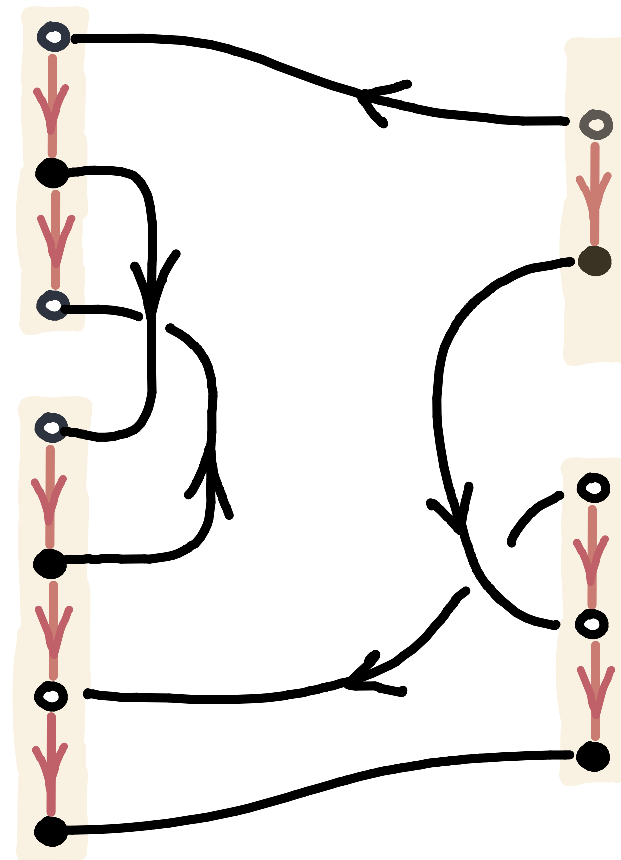
3. Have a concurrency-style internal language.

```
Protocol(request°, login°, get•) {
  Auth(login°, ok•, ack°);
  Serv(ok°, ack•, request°, get•);
}
```

Next Steps



Polar Posets
(c.f. Event Strs.)



*-Polycategory
via Chu Construction

END

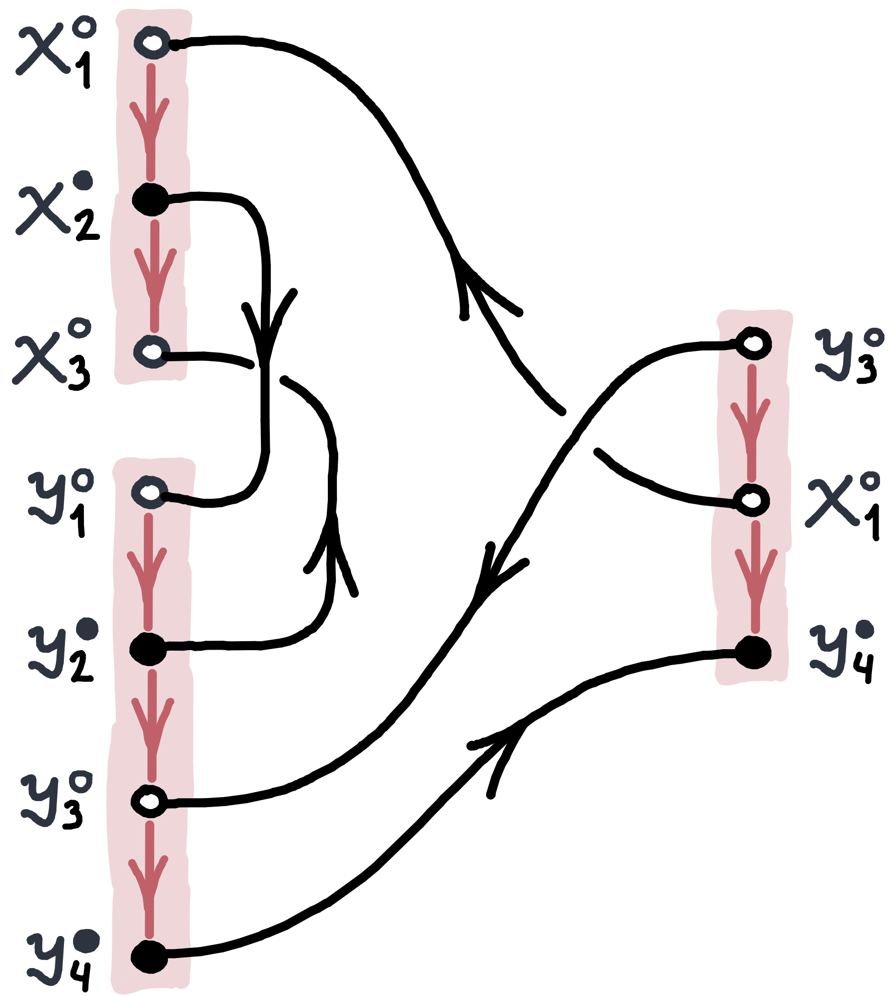
Shuffles

All good if the programs are independent.

What if they communicate?

How can we add message passing?

Polar Shuffles



Polar shuffles mix programs that communicate sending$^\bullet$ and receiving$^\circ$.

MOTIVATION

What is a fundamental structure of concurrency?

What is a fundamental structure of higher-order functional programming? Cartesian-closed categories? Untyped λ-calculi?

What is a fundamental structure of probability? Markov categories.

Abramsky.

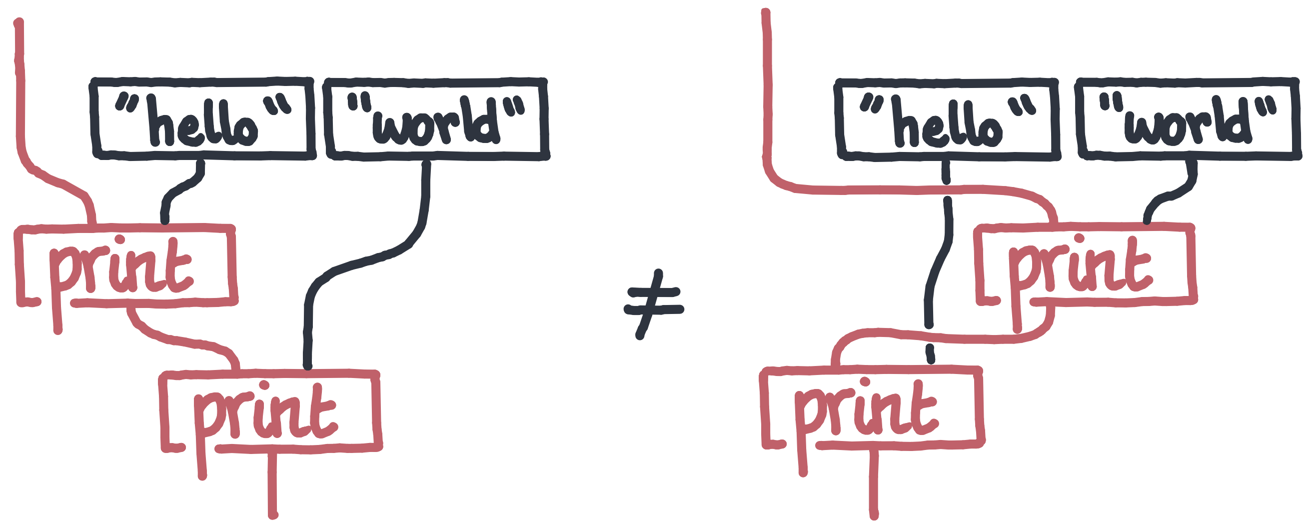
Outline

0. Shuffles
1. Structure: Polar Shuffles.
2. Framework: Monoidal Categories.

Nester

Nester, Voorneveld

STRING DIAGRAMS



Power, Robinson. Premonoidal Categories and Notions of Computation.
Jeffrey. A Graphical View of Programs.
Román. Promonads and String Diagrams for Effectful Categories.
Staton, Møgelberg. Linear Usage of State.

String Diagrams

```
f() = do
  let a = "hello"
  let b = "world"
  print(a)
  print(b)
```

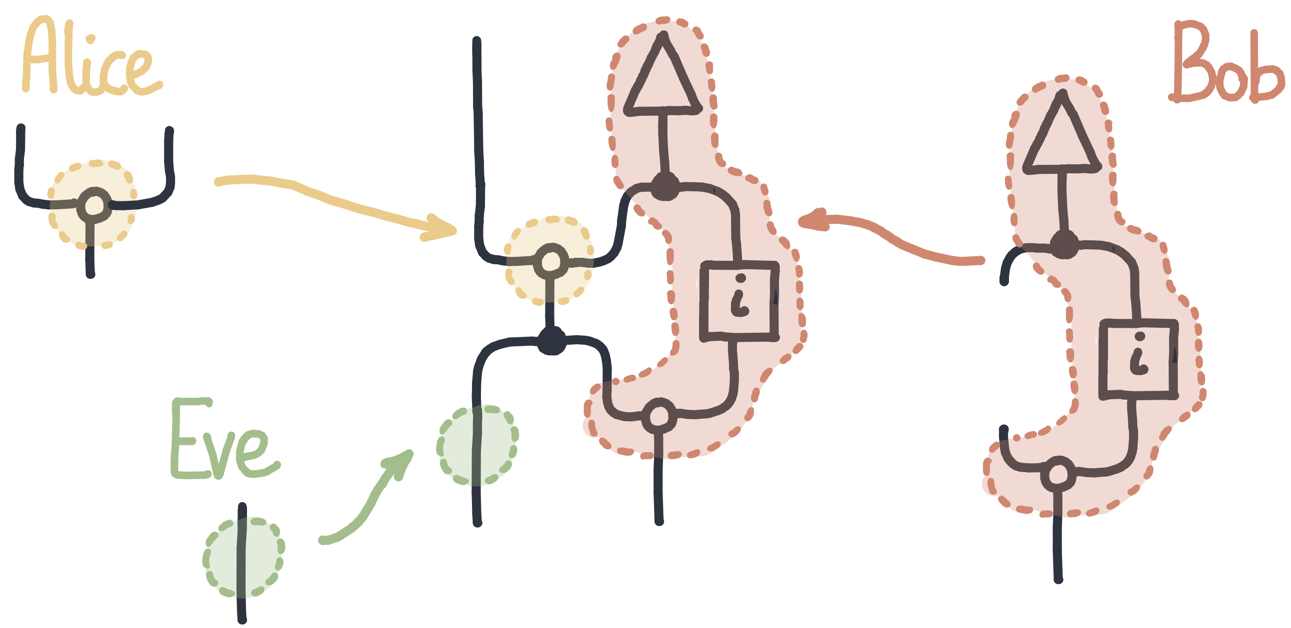
≠

```
f() = do
  let b = "world"
  let a = "hello"
  print(b)
  print(a)
```

;

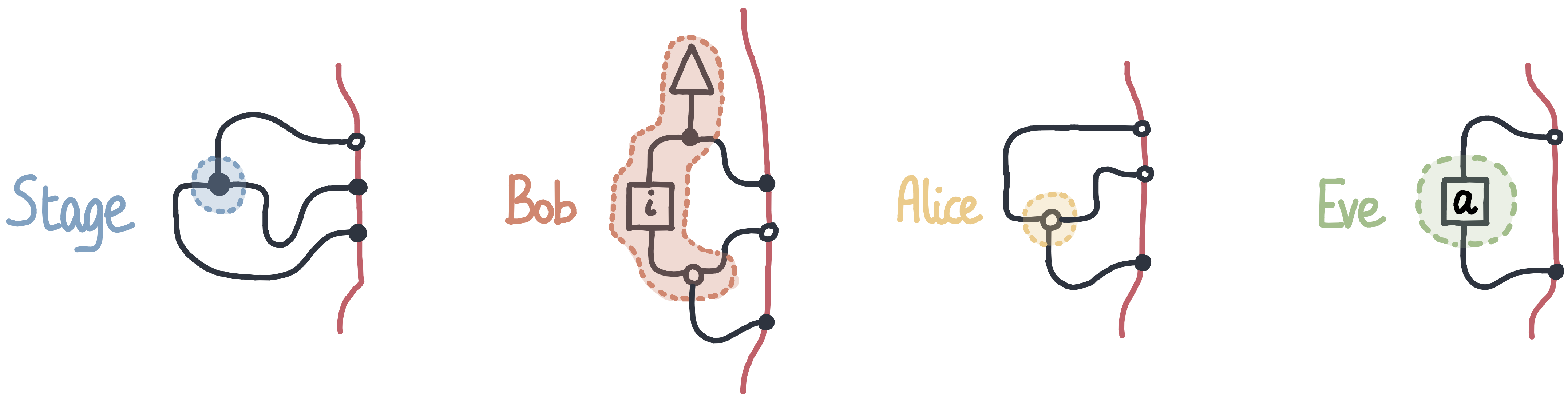
- Staton, Levy. Universal Properties of Impure Programming Languages.
- Jacobs, Hasuo. Freyd is Kleisli, for Arrows.
- Power, Robinson. Premonoidal Categories and Notions of Computation.
- Hughes. Generalising monads to arrows.

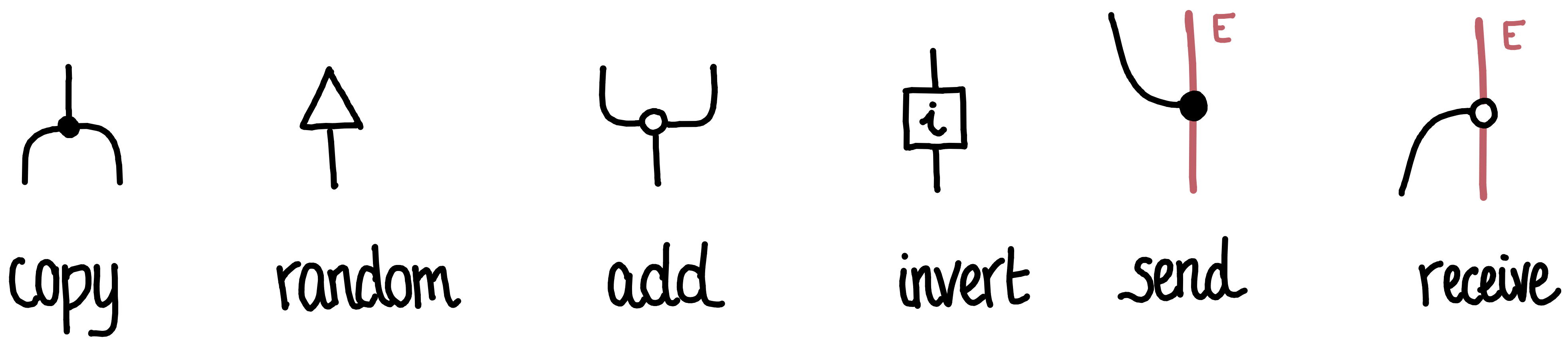
Multi-Party Processes



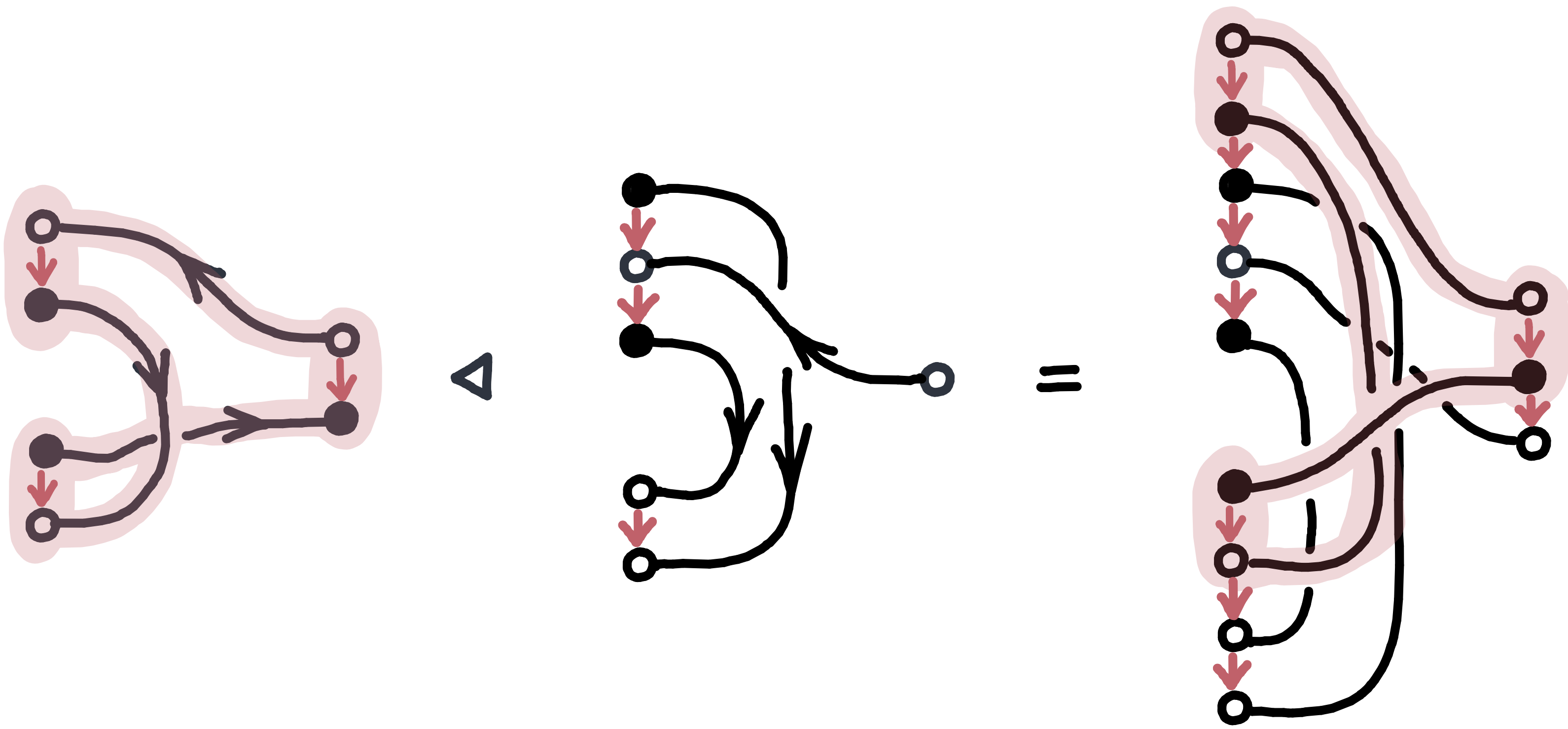
We want to split the morphism into different agents: Alice does not control the broadcast; Eve can only attack at the end; Bob keeps a bit in memory.

String Diagrams + Send/Receive



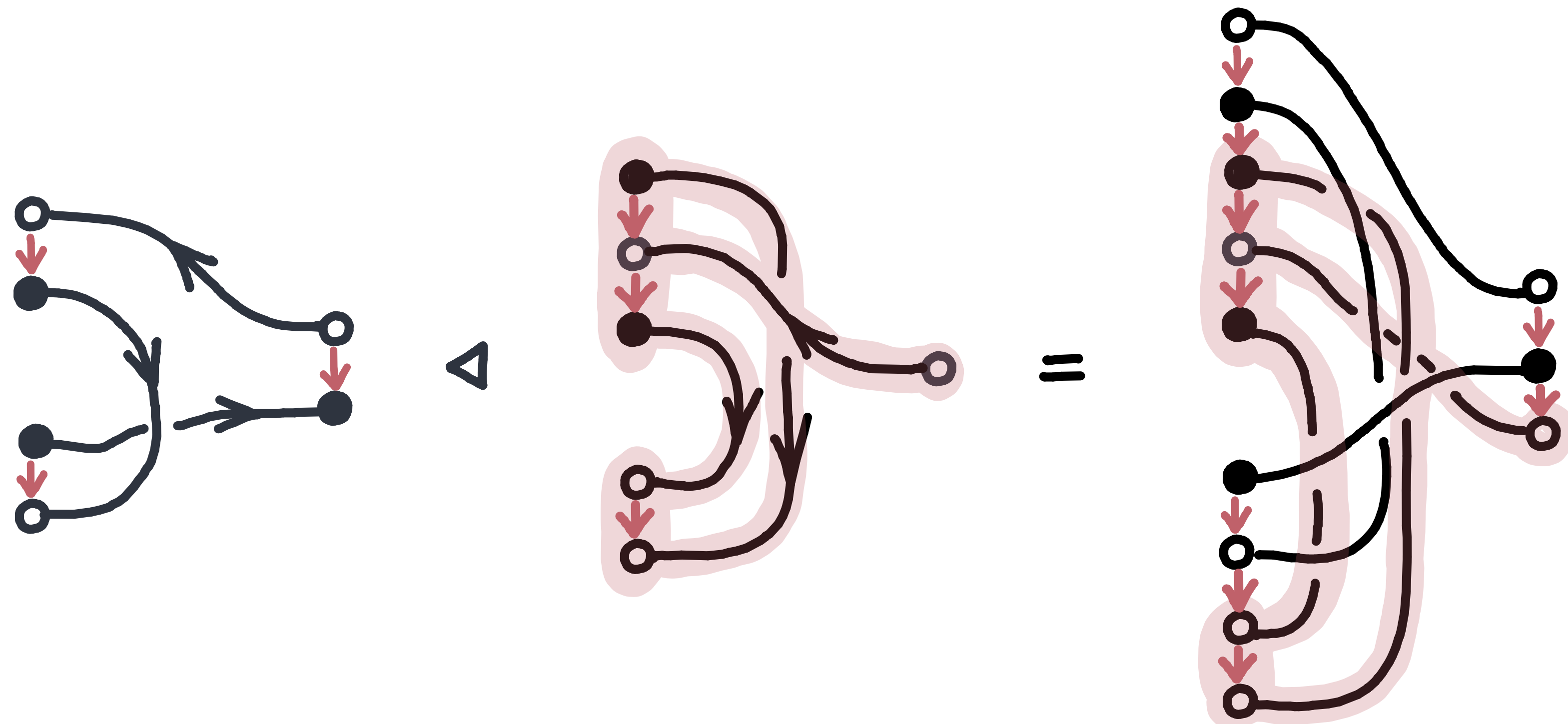


Polar Shuffles



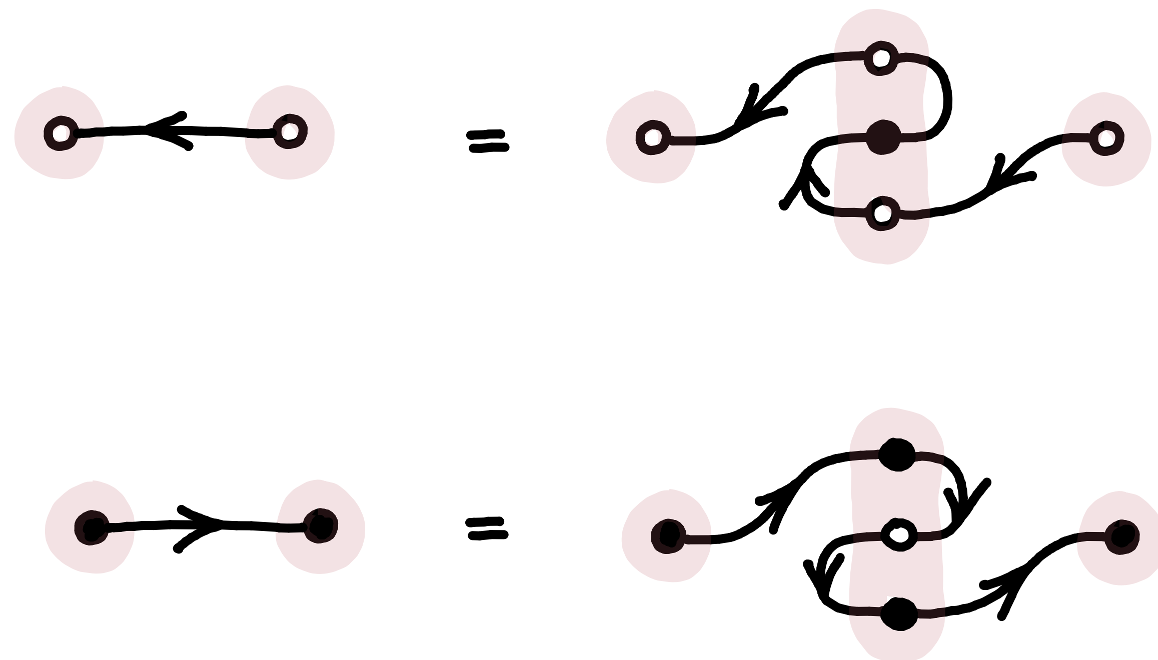
THEOREM. Polar shuffles form the free polar normal monoidal symmetric multicategory.

POLAR SHUFFLES



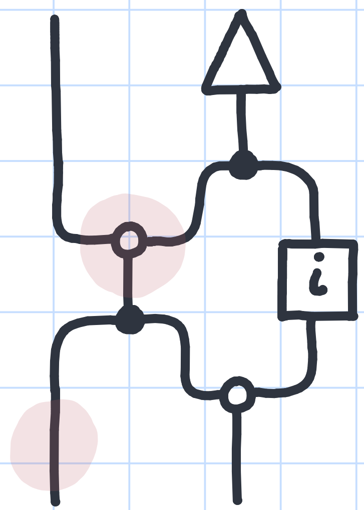
THEOREM. Polar shuffles form the free polar normal monoidal symmetric multicategory.

Polar Shuffles



THEOREM. Polar shuffles form the free polar normal monoidal symmetric multicategory.

ONE-TIME PAD



n w p t	message
1 0 -2 -4	random key
m w n p	crypted message

Message Theories

Shuffles

$$\frac{\Gamma_1 \quad \dots \quad \Gamma_n}{[\Gamma_1, \dots, \Gamma_n]_\sigma}\,\mathsf{SHF}_\sigma \qquad \frac{}{X^\circ, X^\bullet}\,\mathsf{SPW} \qquad \frac{\Gamma, X^\bullet, X^\circ, \Delta}{\Gamma, \Delta}\,\mathsf{LNK}$$

1. We can create a receive-send "echo" session.
2. We can receive what we just sent.
3. Events can be interleaved in any order.

This is the free polarized physical monoidal multicategory.

Message Theories

Dualities

$$\frac{\Gamma_1 \quad \dots \quad \Gamma_n}{[\Gamma_1, \dots, \Gamma_n]_\sigma}\,\mathsf{SHF}_\sigma \qquad \frac{}{X^\circ, X^\bullet}\,\mathsf{SPW} \qquad \frac{\Gamma, X^\bullet, X^\circ, \Delta}{\Gamma, \Delta}\,\mathsf{LNK}$$

1. We can create a receive-send "echo" session.
2. We can receive what we just sent.
3. Events can be interleaved in any order.

This is the free polarized physical monoidal multicategory.

Message Theories: Axioms

Linking is natural with respect to shufflings.

$$
\dfrac{\dfrac{\begin{array}{c}\vdots\, m_1\\ \Gamma_1, X^\bullet, X^\circ, \Gamma_2\end{array}}{\Gamma_1, \Gamma_2} \qquad \begin{array}{c}\vdots\, m_2\\ \Delta_1, \Delta_2\end{array}}{[\Gamma_1 ; \Delta_1]_\sigma, [\Gamma_2 ; \Delta_2]_\tau}
\;=\;
\dfrac{\dfrac{\begin{array}{c}\vdots\, m_1\\ \Gamma_1, X^\bullet, X^\circ, \Gamma_2\end{array} \qquad \begin{array}{c}\vdots\, m_2\\ \Delta_1, \Delta_2\end{array}}{[\Gamma_1 ; \Delta_1]_\sigma, X^\bullet, X^\circ, [\Gamma_2 ; \Delta_2]_\tau}}{[\Gamma_1 ; \Delta_1]_\sigma, [\Gamma_2 ; \Delta_2]_\tau} \;;
$$

Spawning is natural with respect to shufflings.

$$
\dfrac{\dfrac{\begin{array}{c}\vdots\, m_1\\ \Gamma_1, \Gamma_2\end{array}}{\Gamma_1, X^\circ, X^\bullet, \Gamma_2} \qquad \begin{array}{c}\vdots\, m_2\\ \Delta_1, \Delta_2\end{array}}{[\Gamma_1 ; \Delta_1]_\sigma, X^\circ, X^\bullet, [\Gamma_2 ; \Delta_2]_\tau}
\;=\;
\dfrac{\dfrac{\begin{array}{c}\vdots\, m_1\\ \Gamma_1, \Gamma_2\end{array} \quad \begin{array}{c}\vdots\, m_2\\ \Delta_1, \Delta_2\end{array}}{[\Gamma_1 ; \Delta_1]_\sigma, [\Gamma_2 ; \Delta_2]_\tau}}{[\Gamma_1 ; \Delta_1]_\sigma, X^\circ, X^\bullet, [\Gamma_2 ; \Delta_2]_\tau} \;;
$$

Message Theories: Axioms

Spawning and linking are duals.

$$\begin{array}{c} \vdots\, m \\ \Gamma, X^{\circ}, \Delta \\ \hline \Gamma, X^{\circ}, X^{\bullet}, X^{\circ}, \Delta \\ \hline \Gamma, X^{\circ}, \Delta \end{array} \overset{(4_L)}{=} \vdots\, m \quad ; \qquad \begin{array}{c} \vdots\, m \\ \Gamma, X^{\bullet}, \Delta \\ \hline \Gamma, X^{\bullet}, X^{\circ}, X^{\bullet}, \Delta \\ \hline \Gamma, X^{\bullet}, \Delta \end{array} \overset{(4_R)}{=} \vdots\, m \quad ;$$

Spawning and linking interchange.

$$\begin{array}{c} \vdots\, m \\ \Gamma_1, X^{\bullet}, X^{\circ}, \Gamma_2, \Gamma_3 \\ \hline \Gamma_1, X^{\bullet}, X^{\circ}, \Gamma_2, Y^{\circ}, Y^{\bullet}, \Gamma_3 \\ \hline \Gamma_1, \Gamma_2, Y^{\circ}, Y^{\bullet}, \Gamma_3 \end{array} \overset{(5_A)}{=} \begin{array}{c} \vdots\, m \\ \Gamma_1, X^{\bullet}, X^{\circ}, \Gamma_2, \Gamma_3 \\ \hline \Gamma_1, \Gamma_2, \Gamma_3 \\ \hline \Gamma_1, \Gamma_2, Y^{\circ}, Y^{\bullet}, \Gamma_3 \end{array} \; ;$$

This is the free polarized normal monoidal multicategory on a set of types.

MESSAGE THEORIES: AXIOMS

Shuffles compose as in the multicategory of shuffles.

$$\frac{\dfrac{\overset{\vdots\, m_1}{\Gamma_1} \qquad \overset{\vdots\, m_2}{\Gamma_2}}{[\Gamma_1;\Gamma_2]_\sigma} \qquad \overset{\vdots\, m_3}{\Gamma_3}}{[[\Gamma_1;\Gamma_2]_\sigma;\Gamma_3]_\tau} \overset{(1A)}{=} \frac{\overset{\vdots\, m_1}{\Gamma_1} \qquad \dfrac{\overset{\vdots\, m_2}{\Gamma_2} \qquad \overset{\vdots\, m_3}{\Gamma_3}}{[\Gamma_2;\Gamma_3]_{\tau'}}}{[\Gamma_1;[\Gamma_2;\Gamma_3]_{\tau'}]_{\sigma'}} \;;$$

$$\frac{\overset{\vdots\, m}{\Gamma_1} \qquad \overline{}}{[\Gamma_1;]_*} \overset{(1B)}{=} \vdots\, m \;;$$

$$\frac{\overset{\vdots\, m_1}{\Gamma} \qquad \overset{\vdots\, m_2}{\Delta}}{[\Gamma;\Delta]_\sigma} \overset{(1C)}{=} \frac{\overset{\vdots\, m_2}{\Delta} \qquad \overset{\vdots\, m_1}{\Gamma}}{[\Gamma;\Delta]_{\tilde{\sigma}}} \;;$$

This is the free **normal monoidal multicategory** on a set of types.

PROOF: Polar shuffles are Message derivations

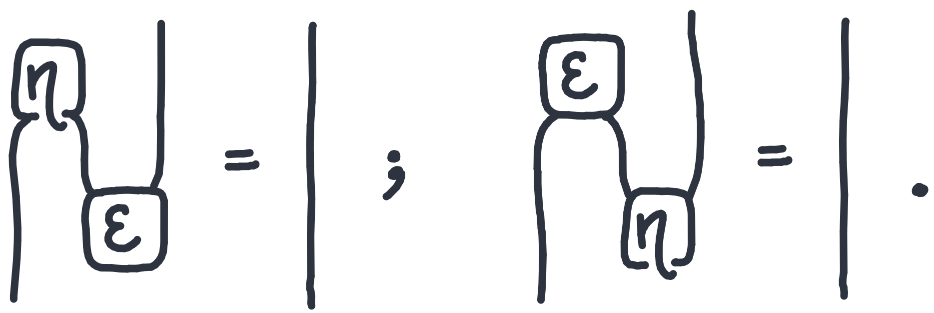
1. Shuffles form the free physical monoidal multicategory. Prop 4.2.9
2. Message theories are shuffles with duals, by definition.
3. Message theories are the free polarized physical monoidal multicategory.
4. Message theories are coherent, by finding their normal form. Thm 4.1.8
5. Polar shuffles are coherent, by definition. Prop 4.4.4
6. A polar shuffle between some types exists iif a message theory derivation exists.
7. Polar shuffles are message theory derivations. Prop 4.4.9
8. Polar shuffles form the free polarized physical mon. multicategory. Thm 4.4.11

PROOF: String diagrams for effectful categories

1. Braiding runtime forms cliques.
2. Braid cliques on the runtime monoidal form a effectful, $\mathrm{Eff}(\mathbb{V}, \mathbb{G})$. Lem 1.7.5
3. There exists an id. on objs. $\mathrm{Mon}(\mathbb{V}) \to \mathrm{Eff}(\mathbb{V}, \mathbb{G})$ preserving mon. structure. Lem 1.7.6
4. There exists a unique effectful functor out of $\mathrm{Mon}(\mathbb{V}) \to \mathrm{Eff}(\mathbb{V}, \mathbb{G})$. Lem 1.7.7
5. The free effectful has morphisms $A \to B$ the $R \otimes A \to R \otimes B$ of the runtime monoidal. Thm 1.7.8
6. String diagrams with runtime are a language for effectful categories. Cor 1.7.9

POLARIZATION

A duality is a pair of objects with two morphisms $(L \dashv R, \varepsilon, \eta)$ such that

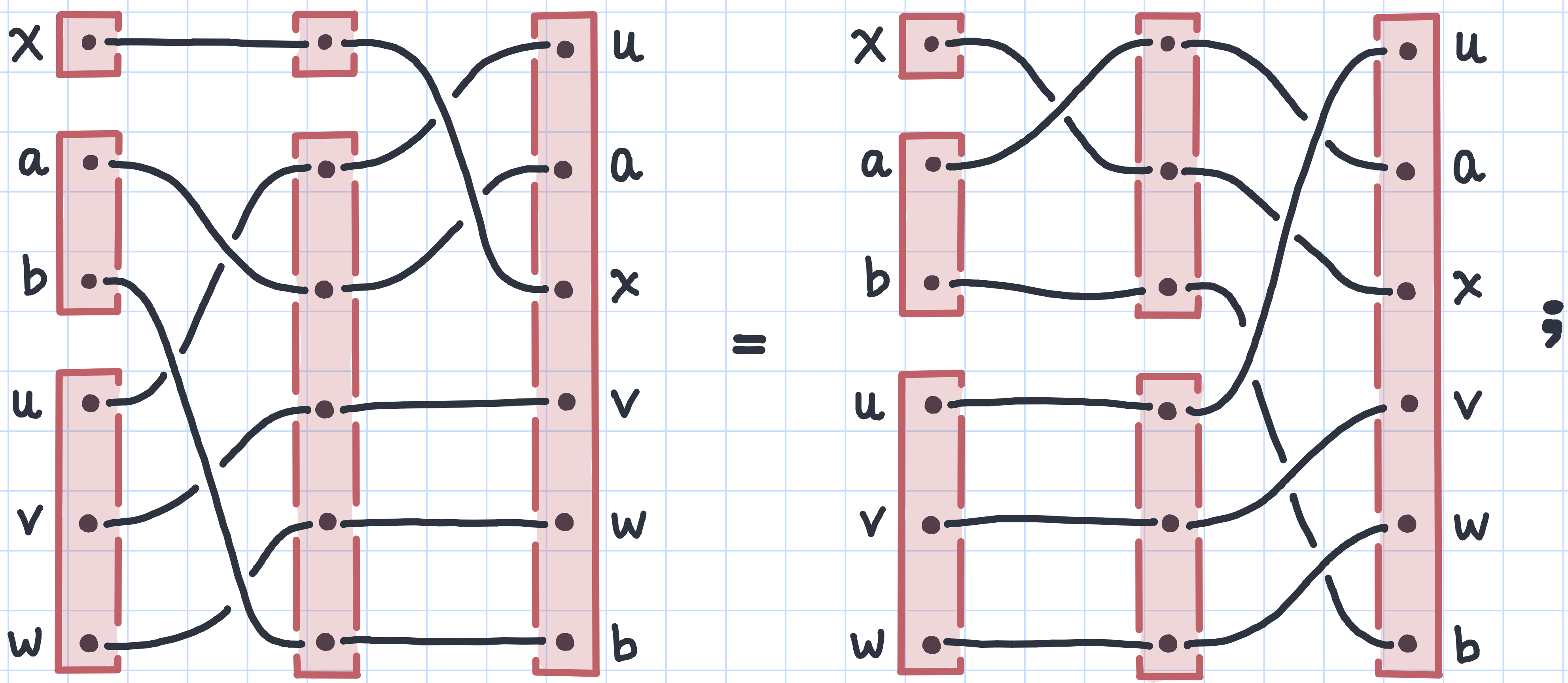


"Polarization is left adjoint to taking left adjoints."

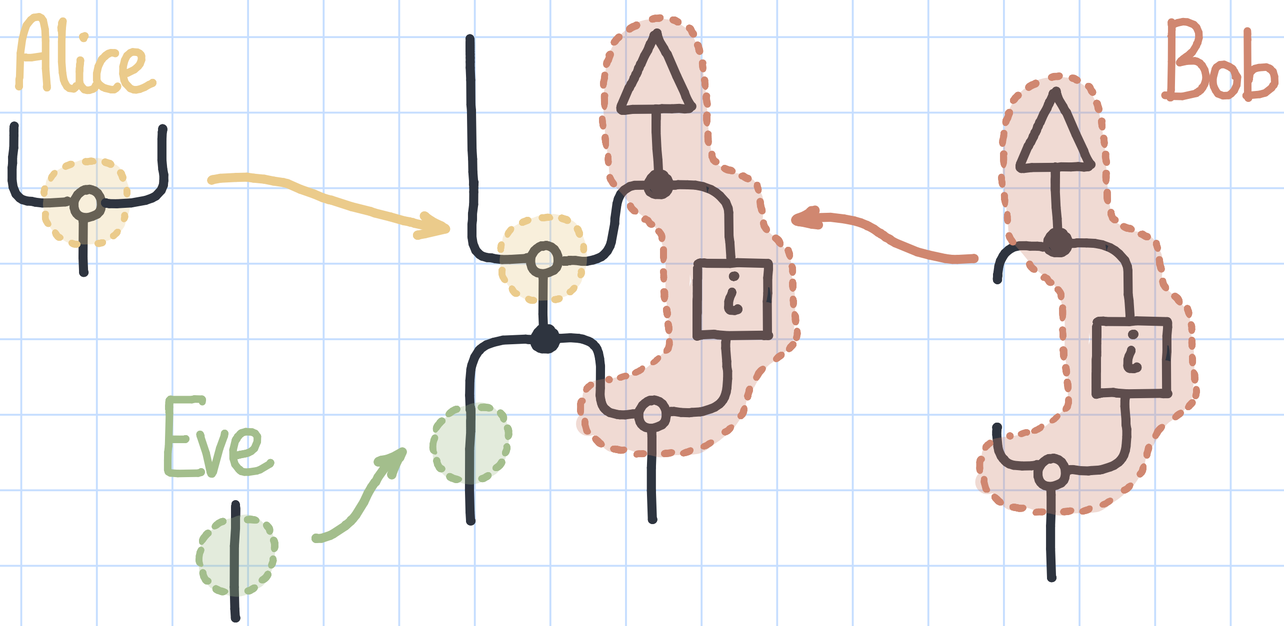
The free polarized monoidal category over a monoidal category.

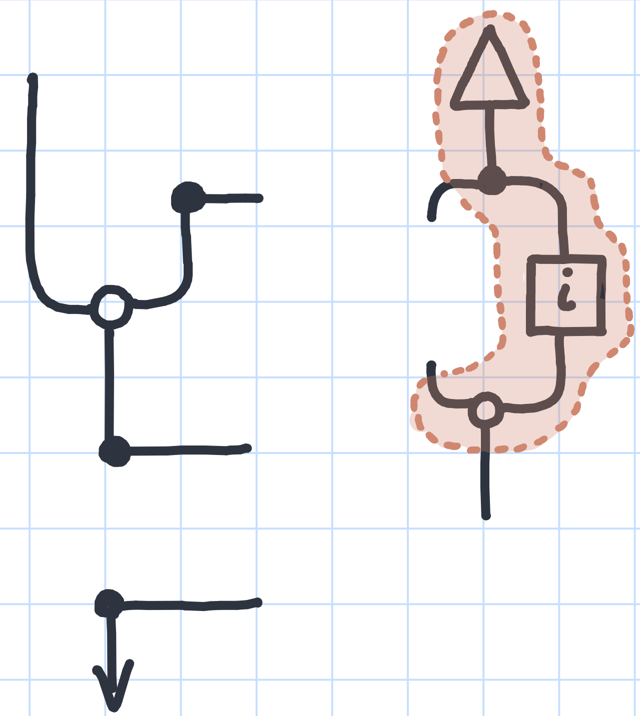
$$\begin{tikzcd} \mathrm{MonCat} \arrow[r, bend left, "\mathrm{LeftAdj}"] \arrow[r, phantom, "\perp"] & \mathrm{MonCat} \arrow[l, bend left, "\mathrm{Polar}"] \end{tikzcd}$$

Shuffles



THEOREM. Shuffles form the free physical monoidal multicategory.


Alice
Bob
Eve
i
i


i