Title: Composing causal orderings

Speakers: Aleks Kissinger

Collection: Indefinite Causal Structure

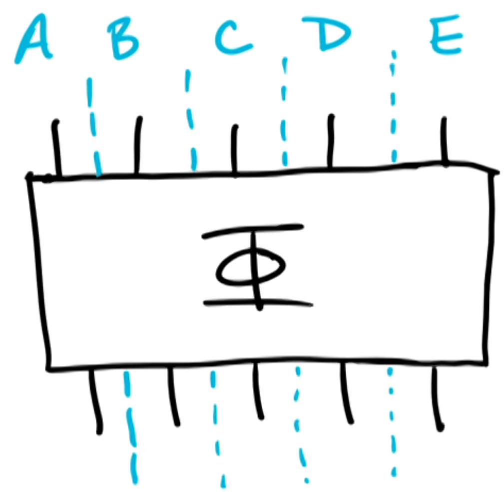
Date: December 09, 2019 - 2:00 PM

URL: http://pirsa.org/19120020

Abstract: When studying (definite or indefinite) causal orderings of processes, it is often useful to consider higher-order processes, i.e. processes which take other processes as their input. However, as a recent no-go result of Guerin et al indicates, our naive first-order notions of "composition" of processes become ill-defined at higher-order. Unlike state spaces, there are multiple non-equivalent notions of "joint system" for process spaces and many different ways one might attempt to plug processes together, with only some giving well-defined (i.e. normalised) processes as outputs. While this starts to look a bit like the Wild West, I'll show in this talk that we can get quite a bit of mileage from considering just two kinds of joint systems: a "non-signalling" tensor product, and a (de Morgan dual) "signalling" product. The interaction between these two products has in fact been well-understood by logicians since the 1980s in a very different disguise: multiplicative linear logic. Using this connection, I'll show how a set of "contractibility" criteria due to Danos and Regnier give a relatively simple, dimension-independent technique for determining whether an arbitrary plugging of higher-order processes is well-defined.

"Black box causal reasoning"

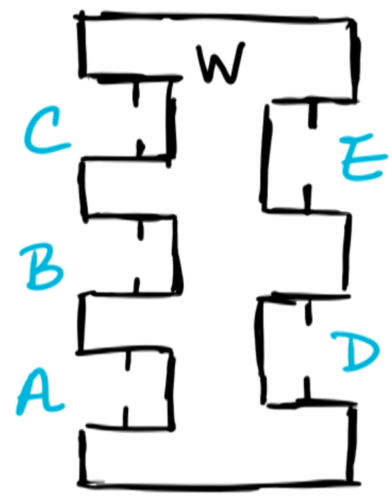
Given:



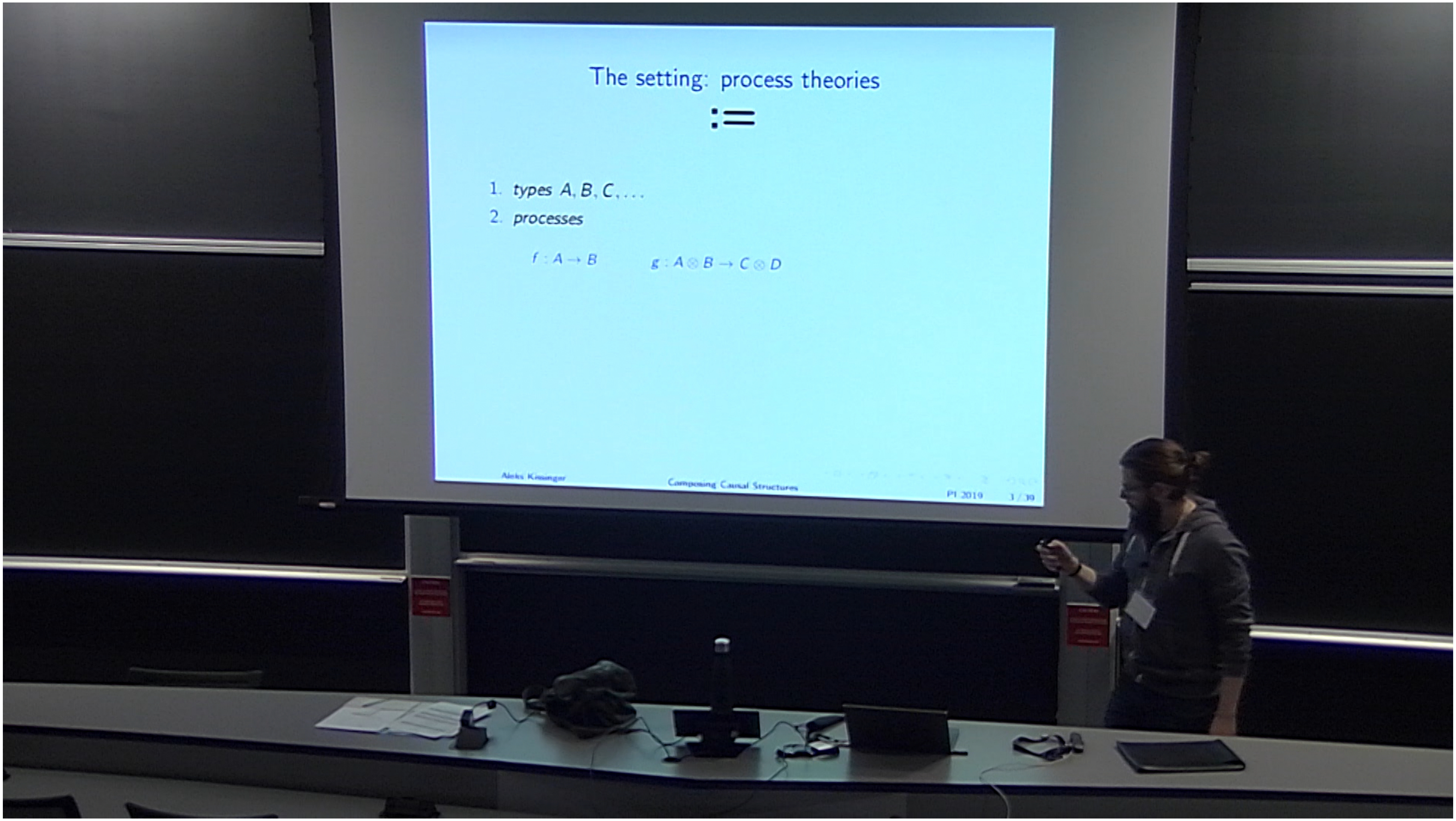
Can we **model**, **observe**, and **prove** properties about causal relationships between **A, B, C, D, E, ...**?

“Black box causal reasoning”

Given:



Can we **model**, **observe**, and **prove** properties about causal relationships between **A, B, C, D, E, ...**?


The setting: process theories
:=
1. types $A, B, C, \ldots$
2. processes
$f : A \to B$
$g : A \otimes B \to C \otimes D$
Aleks Kissinger
Composing Causal Structures
PI 2019

The setting: process theories

$:=$

1. *types* $A, B, C, \ldots$
2. *processes*

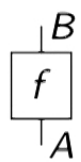
$$f : A \to B \qquad g : A \otimes B \to C \otimes D$$

The setting: process theories

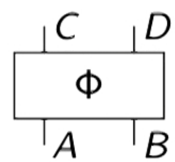
$$:=$$

1. *types* $A, B, C, \ldots$
2. *processes*

$f : A \to B$



$g : A \otimes B \to C \otimes D$



$\rho : I \to A$

states

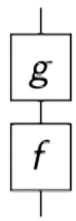
$\pi : A \to I$

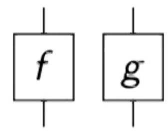
effects

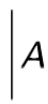
The setting: process theories

$:=$

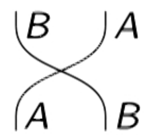
3. *composition* operations:

$g \circ f :=$ 

$f \otimes g :=$ 

$1_A :=$ 

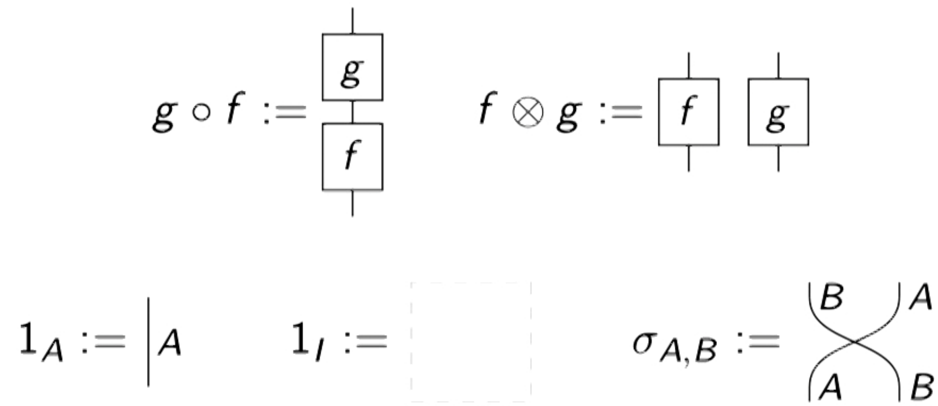
$1_I :=$

$\sigma_{A,B} :=$ 

The setting: process theories

$$:=$$

3. *composition* operations:

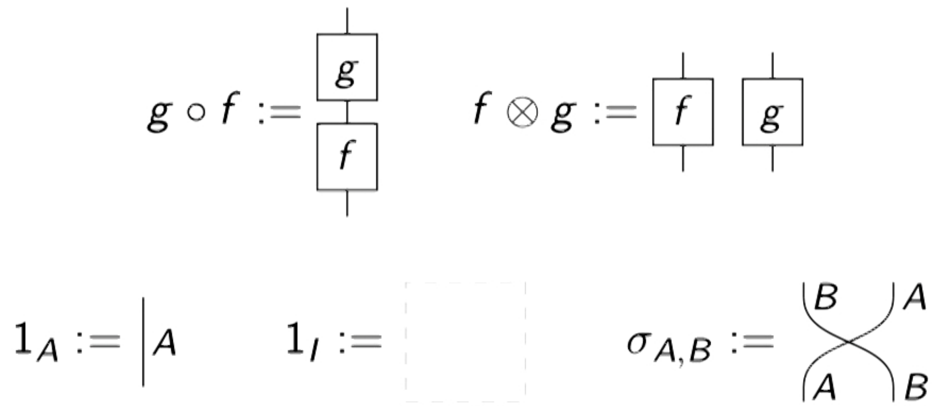


4. some rules...

The setting: process theories

$$:=$$

3. *composition* operations:



4. some rules... (symmetric monoidal category / "only connectivity matters")

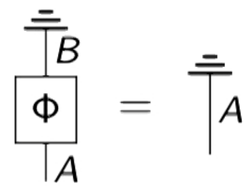
The setting: process theories

5. a special effect for every A:

$\longleftarrow$ discarding

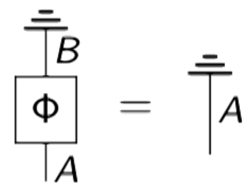
Causality

This enables us to say when a process is *causal*:



Causality

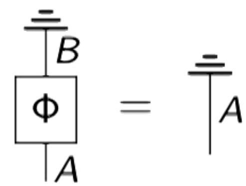
This enables us to say when a process is *causal*:



"If the outputs of a process are ignored, it doesn't matter which process happened."

Causality

This enables us to say when a process is *causal*:



“If the outputs of a process are ignored, it doesn’t matter which process happened.”

Consequence: A causal process only affects other processes which consume its outputs (i.e. those that lie in its *causal future*).

The classical case

$\mathbf{Mat}(\mathbb{R}_+)$ is the process theory whose objects are natural numbers and morphisms are *matrices of positive numbers*. Then:

 $= \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}$

 $= \sum_i \rho^i = 1$

The classical case

Mat$(\mathbb{R}_+)$ is the process theory whose objects are natural numbers and morphisms are *matrices of positive numbers*. Then:

 $= \begin{pmatrix} 1 & 1 & \cdots & 1 \end{pmatrix}$

 $= \sum_i \rho^i = 1$

Causal states $=$ probability distributions
Causal processes $=$ stochastic maps

The quantum case

CPM is the process theory whose objects are Hilbert spaces and morphisms are *completely postive maps*. Then:

 $= \mathrm{Tr}(-)$

$= \mathrm{Tr}(\rho) = 1$

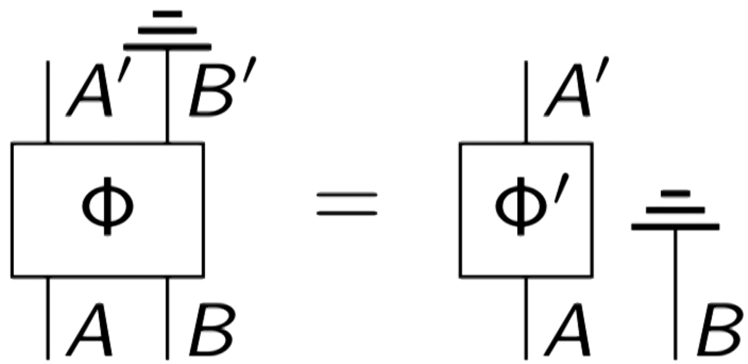
The quantum case

CPM is the process theory whose objects are Hilbert spaces and morphisms are *completely postive maps*. Then:

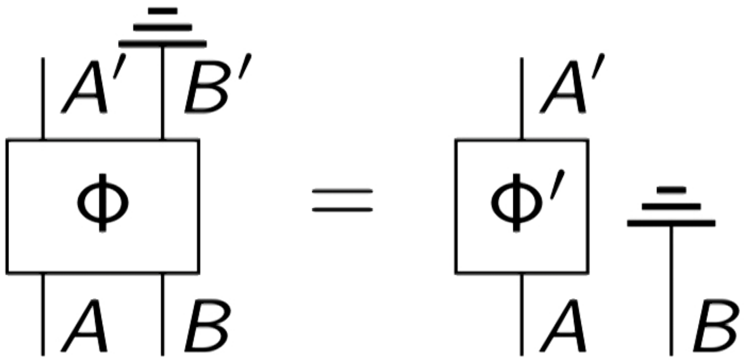
 $= \mathrm{Tr}(-)$  $= \mathrm{Tr}(\rho) = 1$

Causal states $=$ density operators
Causal processes $=$ CPTPs

One-way signalling

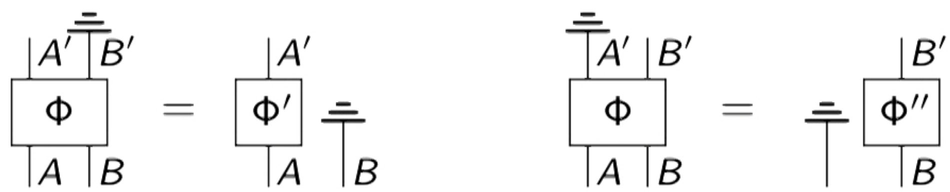


One-way signalling

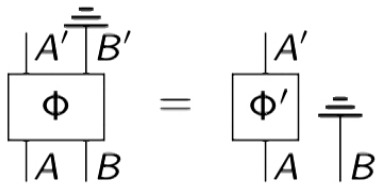


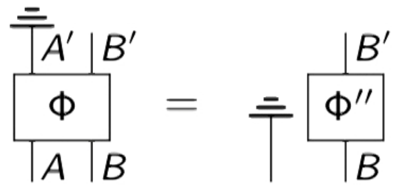
$P(A'|AB) = P(A'|A)$

Non-signalling



Non-signalling





$P(A'|AB) = P(A'|A)$ $\qquad$ $P(B'|AB) = P(B'|B)$

Q: What about higher-order structure?

$$(\textbf{2-COMBS}) : \begin{pmatrix} \textbf{CAUSAL} \\ \textbf{PROCS} \end{pmatrix} \longmapsto \begin{pmatrix} \textbf{CAUSAL} \\ \textbf{PROCS} \end{pmatrix}$$

Q: What about higher-order structure?

$$(\textbf{2-COMBS}) : \begin{pmatrix} \textbf{CAUSAL} \\ \textbf{PROCS} \end{pmatrix} \longmapsto \begin{pmatrix} \textbf{CAUSAL} \\ \textbf{PROCS} \end{pmatrix}$$

$$(\textbf{3-COMBS}) : (\textbf{2-COMBS}) \longmapsto \begin{pmatrix} \textbf{CAUSAL} \\ \textbf{PROCS} \end{pmatrix}$$

...

$$\begin{pmatrix} \textbf{PROCESS} \\ \textbf{MATRICES} \end{pmatrix} : \begin{pmatrix} \textbf{CAUSAL} \\ \textbf{PROCS} \end{pmatrix} \times \begin{pmatrix} \textbf{CAUSAL} \\ \textbf{PROCS} \end{pmatrix} \longmapsto (\mathbf{1})$$

...

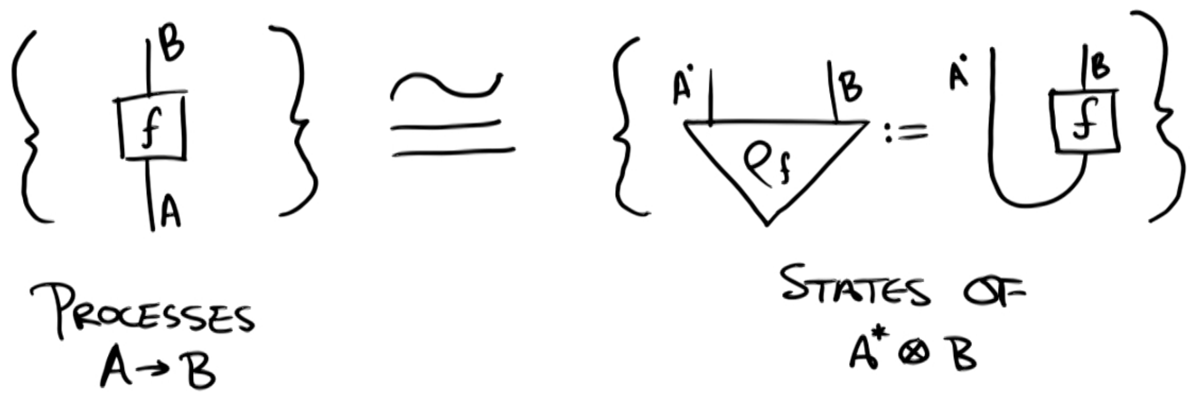
Compact closed categories

:= Process theories + 'Abstract Choi-Jamiołkowski'

For every A, $\exists A^*$ and two processes:

Such that:

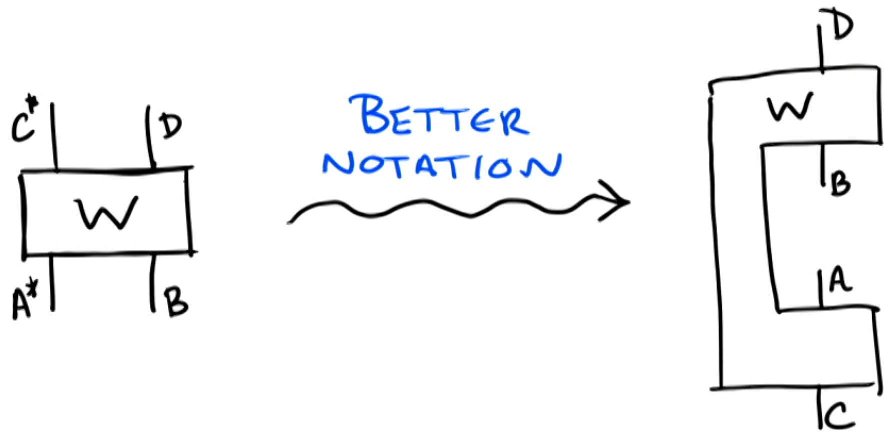
:= Process theories + 'Abstract Choi-Jamiołkowski'



Higher-order processes

Let $A \multimap B := A^* \otimes B$.

$$W : (A \multimap B) \to (C \multimap D)$$



A blessing and a curse...

In a compact closed category:

$$(A \otimes B)^* = A^* \otimes B^*$$

Which gives:

$$(A \multimap B) \multimap C$$

A blessing and a curse...

In a compact closed category:

$$(A \otimes B)^* = A^* \otimes B^*$$

Which gives:

$$\begin{aligned}(A \multimap B) \multimap C &\cong (A \multimap B)^* \otimes C \\ &\cong (A^* \otimes B)^* \otimes C\end{aligned}$$

A blessing and a curse...

In a compact closed category:

$$(A \otimes B)^* = A^* \otimes B^*$$

Which gives:

$$\begin{aligned}(A \multimap B) \multimap C &\cong (A \multimap B)^* \otimes C \\ &\cong (A^* \otimes B)^* \otimes C \\ &\cong A \otimes B^* \otimes C\end{aligned}$$

A blessing and a curse...

In a compact closed category:

$$(A \otimes B)^* = A^* \otimes B^*$$

Which gives:

$$\begin{aligned}(A \multimap B) \multimap C &\cong (A \multimap B)^* \otimes C \\ &\cong (A^* \otimes B)^* \otimes C \\ &\cong A \otimes B^* \otimes C \\ &\cong B^* \otimes A \otimes C\end{aligned}$$

A blessing and a curse...

In a compact closed category:

$$(A \otimes B)^* = A^* \otimes B^*$$

Which gives:

$$\begin{aligned}
(A \multimap B) \multimap C &\cong (A \multimap B)^* \otimes C \\
&\cong (A^* \otimes B)^* \otimes C \\
&\cong A \otimes B^* \otimes C \\
&\cong B^* \otimes A \otimes C \\
&\cong B \multimap A \otimes C
\end{aligned}$$

*-autonomous categories

...are almost the same as compact closed except:

$$(A^* \otimes B^*)^* \ncong A \otimes B$$

*-autonomous categories

...are almost the same as compact closed except:

$$(A^* \otimes B^*)^* =: A \parr B$$

The recipe

‘Raw materials’ $\mathcal{C}$ $\mapsto$ $\mathrm{Caus}[\mathcal{C}]$

compact closed category *+ well-behaved* $\bar{\top}$

$\ast$*-autonomous category capturing ‘logic of causality’*

The recipe

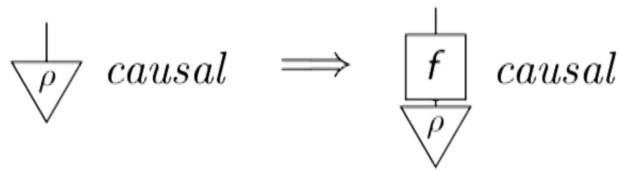
'Raw materials' $\mathcal{C}$	$\mapsto$	$\mathrm{Caus}[\mathcal{C}]$
compact closed category + *well-behaved*		$*$*-autonomous category* *capturing 'logic of causality'*
Mat$(\mathbb{R}_+)$	$\mapsto$	higher-order stochastic maps
CPM	$\mapsto$	higher-order quantum channels

The recipe

'Raw materials' $\mathcal{C}$	$\mapsto$	$\mathrm{Caus}[\mathcal{C}]$
compact closed category + *well-behaved* $\overline{\top}$		$*$*-autonomous category* *capturing 'logic of causality'*
Mat$(\mathbb{R}_+)$	$\mapsto$	higher-order stochastic maps
CPM	$\mapsto$	higher-order quantum channels
???	$\mapsto$	non-standard causal models

Building Caus[$\mathcal{C}$]

A process is causal, a.k.a. *first order causal*, if and only if it preserves the set of causal states:



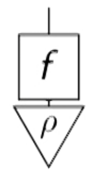
That is, it preserves:

$$c = \left\{ \rho : I \to A \;\middle|\;$$

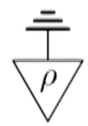

$$= 1 \right\} \subseteq \mathbf{st}(A)$$

Building $\mathrm{Caus}[\mathcal{C}]$

A process is causal, a.k.a. *first order causal*, if and only if it preserves the set of causal states:

 *causal* $\Longrightarrow$  *causal*

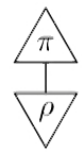
That is, it preserves:

$c = \Big\{ \rho : I \to A \;\Big|$  $= 1 \Big\} \subseteq \mathbf{st}(A)$

We define $\mathrm{Caus}[\mathcal{C}]$ by equipping each object with *a generalisation of* the set c, and requiring processes to preserve it.

Building $\mathrm{Caus}[\mathcal{C}]$

Note *any* set of states $c \subseteq \mathcal{C}(I, A)$ admits a *dual*, which is a set of effects:

$$c^* \; := \; \left\{ \pi : A^* \;\middle|\; \forall \rho \in c \;.\right.$$


$$\left. = \; 1 \right\}$$

Building Caus[$\mathcal{C}$]

Note *any* set of states $c \subseteq \mathcal{C}(I, A)$ admits a *dual*, which is a set of effects:

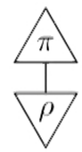
$$c^* \;:=\; \left\{ \pi : A^* \;\middle|\; \forall \rho \in c \,.\, \begin{array}{c} \overset{\pi}{\triangle} \\ | \\ \underset{\rho}{\triangledown} \end{array} \;=\; 1 \right\}$$

The double-dual c^{**} is a set of states again.

Building $\mathrm{Caus}[\mathcal{C}]$

Note *any* set of states $c \subseteq \mathcal{C}(I, A)$ admits a *dual*, which is a set of effects:

$$c^* \;:=\; \left\{ \pi : A^* \;\middle|\; \forall \rho \in c \;.\right.$$



$$\left. = \; 1 \right\}$$

The double-dual c^{**} is a set of states again.

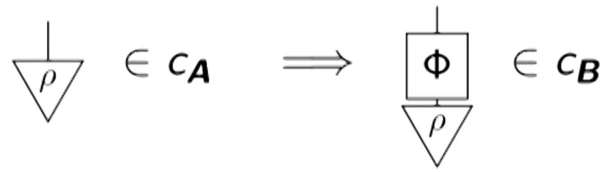
Definition

A set of states $c \subseteq \mathbf{st}(A)$ is *closed* if $c = c^{**}$.

Building Caus[$\mathcal{C}$]

$\mathrm{Caus}[\mathcal{C}]$ is a new process theory where:

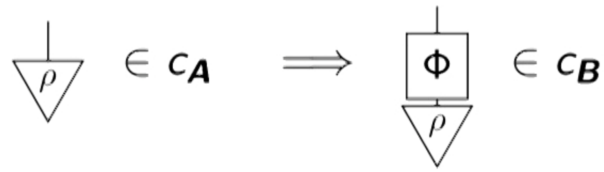
- types are pairs $\boldsymbol{A} := (A, c_{\boldsymbol{A}} \subseteq \mathbf{st}(A))$
- processes $\Phi : \boldsymbol{A} \to \boldsymbol{B}$ are Φ from $\mathcal{C}$ where:



Building $\mathrm{Caus}[\mathcal{C}]$

$\mathrm{Caus}[\mathcal{C}]$ is a new process theory where:

- types are pairs $\boldsymbol{A} := (A, c_{\boldsymbol{A}} \subseteq \mathbf{st}(A))$
- processes $\Phi : \boldsymbol{A} \to \boldsymbol{B}$ are Φ from $\mathcal{C}$ where:



- ...and letting:

$$\boldsymbol{A}^* := (A^*, c_{\boldsymbol{A}}^*) \qquad\qquad \boldsymbol{A} \otimes \boldsymbol{B} := (A \otimes B, (c_{\boldsymbol{A}} \otimes c_{\boldsymbol{B}})^{**})$$

makes $\mathrm{Caus}[\mathcal{C}]$ into a *-autonomous category.

Connectives

One connective $\otimes$ becomes 3 interrelated ones:

$$\mathbf{A} \otimes \mathbf{B}$$
$$\mathbf{A} \parr \mathbf{B} := (\mathbf{A}^* \otimes \mathbf{B}^*)^*$$
$$\mathbf{A} \multimap \mathbf{B} := \mathbf{A}^* \parr \mathbf{B} \cong (\mathbf{A} \otimes \mathbf{B}^*)^*$$

Connectives

One connective $\otimes$ becomes 3 interrelated ones:

$$\begin{aligned} &\boldsymbol{A} \otimes \boldsymbol{B} \\ &\boldsymbol{A} \parr \boldsymbol{B} := (\boldsymbol{A}^* \otimes \boldsymbol{B}^*)^* \\ &\boldsymbol{A} \multimap \boldsymbol{B} := \boldsymbol{A}^* \parr \boldsymbol{B} \cong (\boldsymbol{A} \otimes \boldsymbol{B}^*)^* \end{aligned}$$

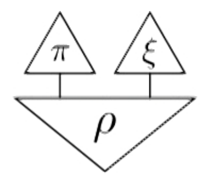
- $\otimes$ is the smallest joint state space that contains all product states

Connectives

One connective $\otimes$ becomes 3 interrelated ones:

$$
\begin{aligned}
&\boldsymbol{A} \otimes \boldsymbol{B} \\
&\boldsymbol{A} \parr \boldsymbol{B} := (\boldsymbol{A}^* \otimes \boldsymbol{B}^*)^* \\
&\boldsymbol{A} \multimap \boldsymbol{B} := \boldsymbol{A}^* \parr \boldsymbol{B} \cong (\boldsymbol{A} \otimes \boldsymbol{B}^*)^*
\end{aligned}
$$

- $\otimes$ is the smallest joint state space that contains all product states
- $\parr$ is the biggest joint state space normalised on all product effects:

$c_{\boldsymbol{A} \parr \boldsymbol{B}} = \left\{ \rho : A \otimes B \;\middle|\; \forall \pi \in c_{\boldsymbol{A}}^*, \xi \in c_{\boldsymbol{B}}^* \,.\right.$  $\left. = 1 \right\}$

Example: first-order systems

First order := systems of the form $\boldsymbol{A} = (A, \{\,\overset{=}{\top}\,\}^*)$

Example: first-order systems

First order := systems of the form $\boldsymbol{A} = (A, \{\top\}^*)$

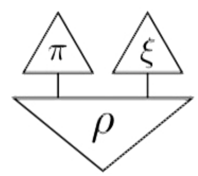
$c_{\boldsymbol{A} \otimes \boldsymbol{B}} := (c_{\boldsymbol{A}} \otimes c_{\boldsymbol{B}})^{**} = (\top \;\; \top)^* =$ all causal states

Example: first-order systems

First order := systems of the form $\boldsymbol{A} = (A, \{ \,\bar{\top}\, \}^*)$

$$c_{\boldsymbol{A}\otimes\boldsymbol{B}} := (c_{\boldsymbol{A}} \otimes c_{\boldsymbol{B}})^{**} = (\,\bar{\top}\quad\bar{\top}\,)^* = \text{all causal states}$$

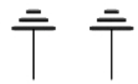
$$c_{\boldsymbol{A}\parr\boldsymbol{B}} := \left\{ \rho : A \otimes B \;\middle|\; \forall \pi \in c^*_{\boldsymbol{A}}, \xi \in c^*_{\boldsymbol{B}} \,.\; \right.$$

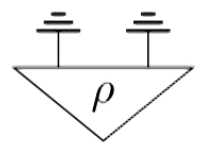


$$\left. = 1 \right\}$$

Example: first-order systems

First order := systems of the form $\boldsymbol{A} = (A, \{$  $\}^*)$

$c_{\boldsymbol{A}\otimes\boldsymbol{B}} := (c_{\boldsymbol{A}} \otimes c_{\boldsymbol{B}})^{**} = ($  $)^* =$ all causal states

$c_{\boldsymbol{A}\parr\boldsymbol{B}} := \Big\{ \rho : A \otimes B \;\Big|$  $= 1 \Big\}$ = all causal states

Theorem

For first order systems, $\boldsymbol{A} \otimes \boldsymbol{B} \cong \boldsymbol{A} \parr \boldsymbol{B}$.

When $\otimes \neq \parr$

For f.o. $\boldsymbol{A}, \boldsymbol{A}', \boldsymbol{B}, \boldsymbol{B}'$:

$$(\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}')$$

When $\otimes \neq \parr$

For f.o. $\boldsymbol{A}, \boldsymbol{A}', \boldsymbol{B}, \boldsymbol{B}'$:

$$\begin{aligned}(\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}') &\cong \boldsymbol{A}^* \parr \boldsymbol{A}' \parr \boldsymbol{B}^* \parr \boldsymbol{B}' \\ &\cong \boldsymbol{A}^* \parr \boldsymbol{B}^* \parr \boldsymbol{A}' \parr \boldsymbol{B}' \\ &\cong (\boldsymbol{A} \otimes \boldsymbol{B})^* \parr \boldsymbol{A}' \parr \boldsymbol{B}'\end{aligned}$$

When $\otimes \neq \parr$

For f.o. $\boldsymbol{A}, \boldsymbol{A}', \boldsymbol{B}, \boldsymbol{B}'$:

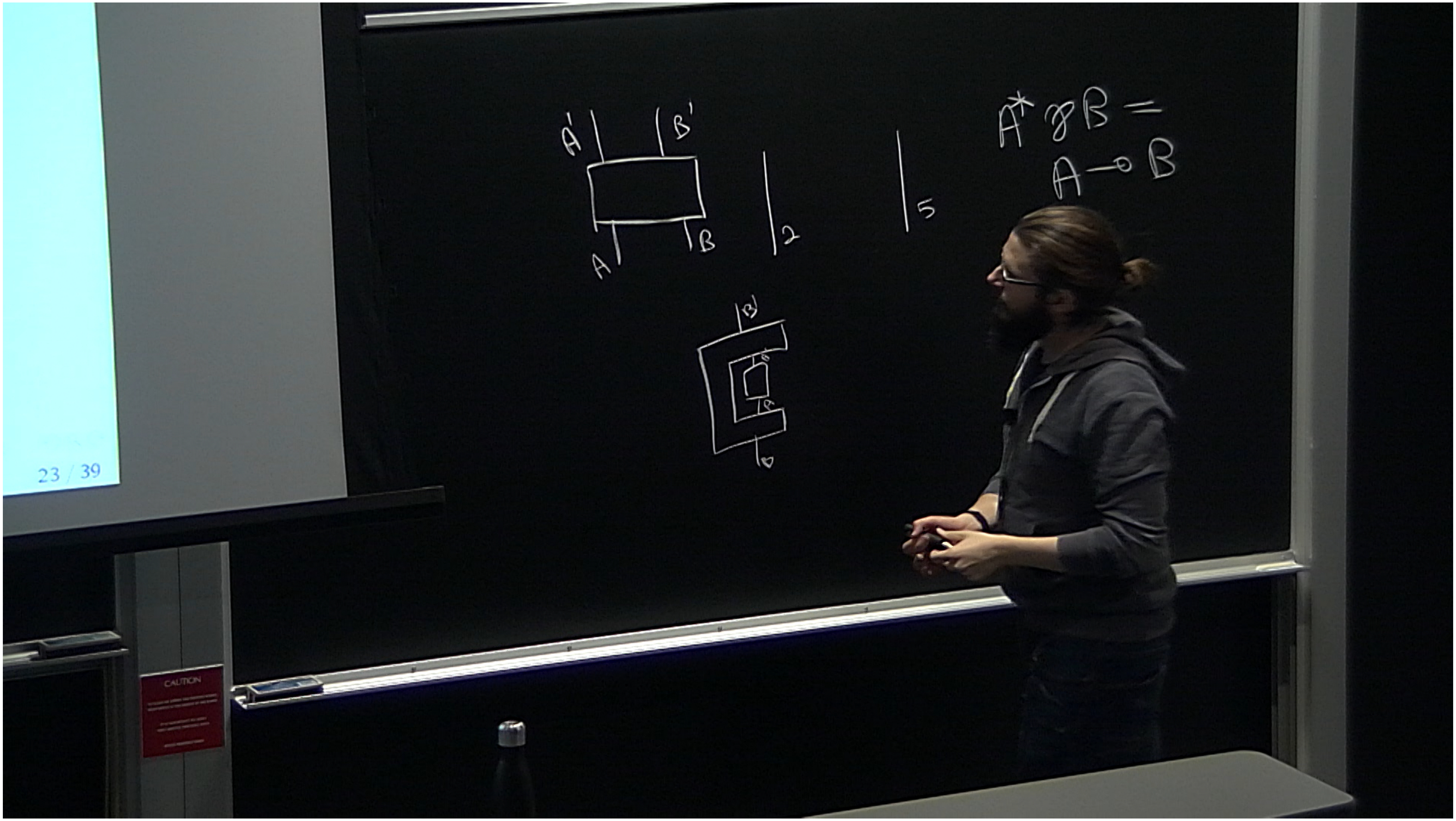
$$
\begin{aligned}
(\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}') &\cong \boldsymbol{A}^* \parr \boldsymbol{A}' \parr \boldsymbol{B}^* \parr \boldsymbol{B}' \\
&\cong \boldsymbol{A}^* \parr \boldsymbol{B}^* \parr \boldsymbol{A}' \parr \boldsymbol{B}' \\
&\cong (\boldsymbol{A} \otimes \boldsymbol{B})^* \parr \boldsymbol{A}' \parr \boldsymbol{B}' \\
&\cong (\boldsymbol{A} \otimes \boldsymbol{B})^* \parr (\boldsymbol{A}' \otimes \boldsymbol{B}') \\
&\cong \boldsymbol{A} \otimes \boldsymbol{B} \multimap \boldsymbol{A}' \otimes \boldsymbol{B}'
\end{aligned}
$$

When $\otimes \neq \parr$

For f.o. $\boldsymbol{A}, \boldsymbol{A}', \boldsymbol{B}, \boldsymbol{B}'$:

$$\begin{aligned}
(\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}') &\cong \boldsymbol{A}^* \parr \boldsymbol{A}' \parr \boldsymbol{B}^* \parr \boldsymbol{B}' \\
&\cong \boldsymbol{A}^* \parr \boldsymbol{B}^* \parr \boldsymbol{A}' \parr \boldsymbol{B}' \\
&\cong (\boldsymbol{A} \otimes \boldsymbol{B})^* \parr \boldsymbol{A}' \parr \boldsymbol{B}' \\
&\cong (\boldsymbol{A} \otimes \boldsymbol{B})^* \parr (\boldsymbol{A}' \otimes \boldsymbol{B}') \\
&\cong \boldsymbol{A} \otimes \boldsymbol{B} \multimap \boldsymbol{A}' \otimes \boldsymbol{B}'
\end{aligned}$$

$(\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}') =$ all causal processes


23 / 39
A'
B'
A
B
2
5
$A^* \otimes B =$
A—o B
CAUTION

Theorem

$(\boldsymbol{A} \multimap \boldsymbol{A}') \otimes (\boldsymbol{B} \multimap \boldsymbol{B}') =$ *causal*, **non-signalling** *processes*

Refining causal structure

Since $\boldsymbol{I} \cong \boldsymbol{I}^* = (I, \{1\})$, a standard theorem of $*$-autonomous gives a canonical embedding:

$$(\boldsymbol{A} \multimap \boldsymbol{A}') \otimes (\boldsymbol{B} \multimap \boldsymbol{B}') \hookrightarrow (\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}')$$

Refining causal structure

Since $\boldsymbol{I} \cong \boldsymbol{I}^* = (I, \{1\})$, a standard theorem of $*$-autonomous gives a canonical embedding:

$$(\boldsymbol{A} \multimap \boldsymbol{A}') \otimes (\boldsymbol{B} \multimap \boldsymbol{B}') \;\hookrightarrow\; (\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}')$$

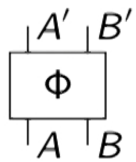
What about in between?

$$(\boldsymbol{A} \multimap \boldsymbol{A}') \otimes (\boldsymbol{B} \multimap \boldsymbol{B}') \;\hookrightarrow\; \cdots \;\hookrightarrow\; (\boldsymbol{A} \multimap \boldsymbol{A}') \parr (\boldsymbol{B} \multimap \boldsymbol{B}')$$

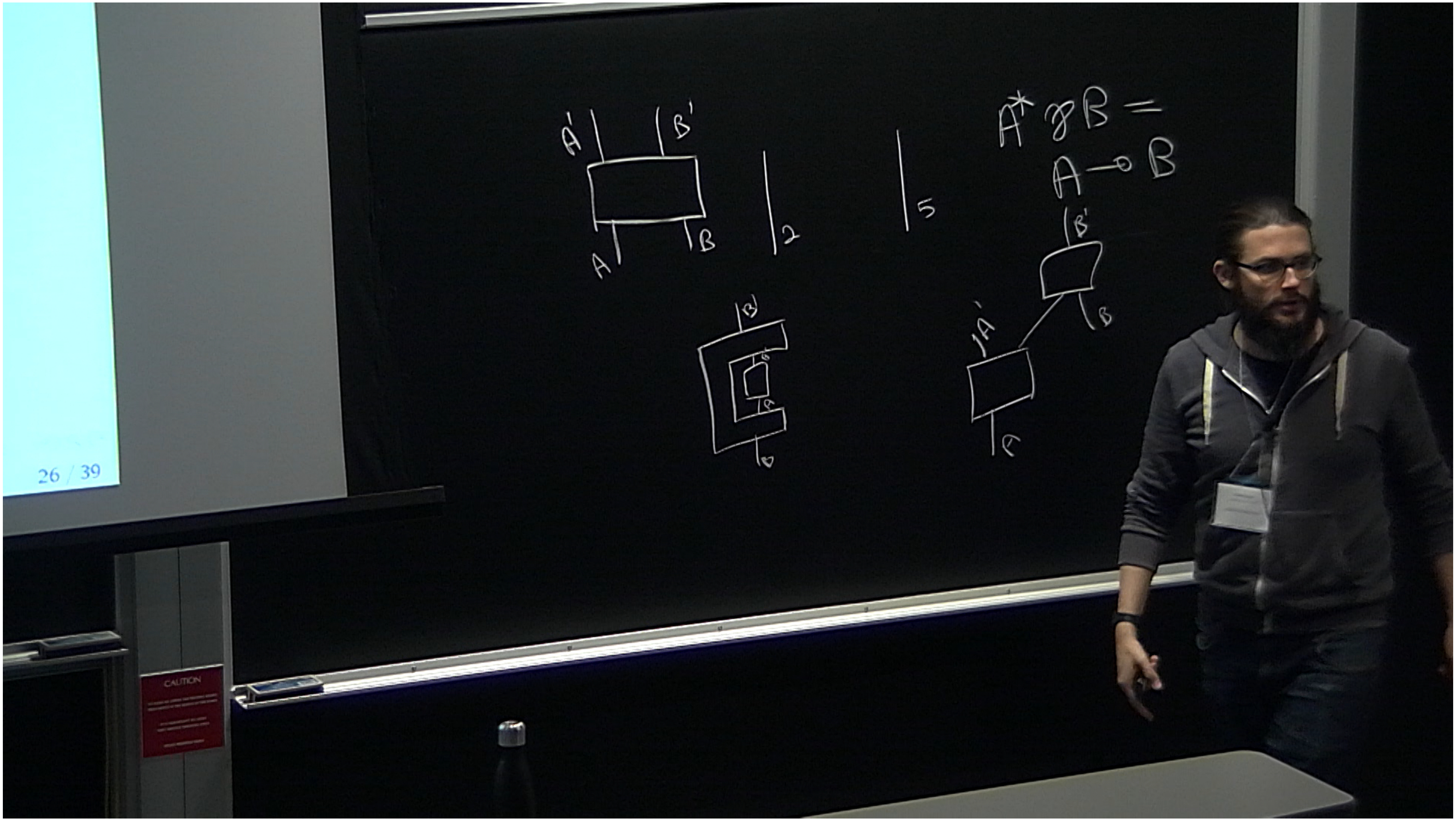
One-way signalling

Theorem

One-way signalling processes are processes of the form:



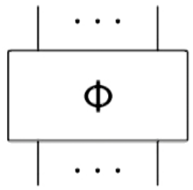
$$: \; \boldsymbol{A} \multimap (\boldsymbol{A}' \multimap \boldsymbol{B}) \multimap \boldsymbol{B}'$$



26 / 39
$A^* \otimes B =$
$A \multimap B$

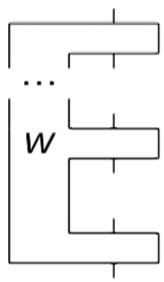
Further examples

- n-party non-signalling:

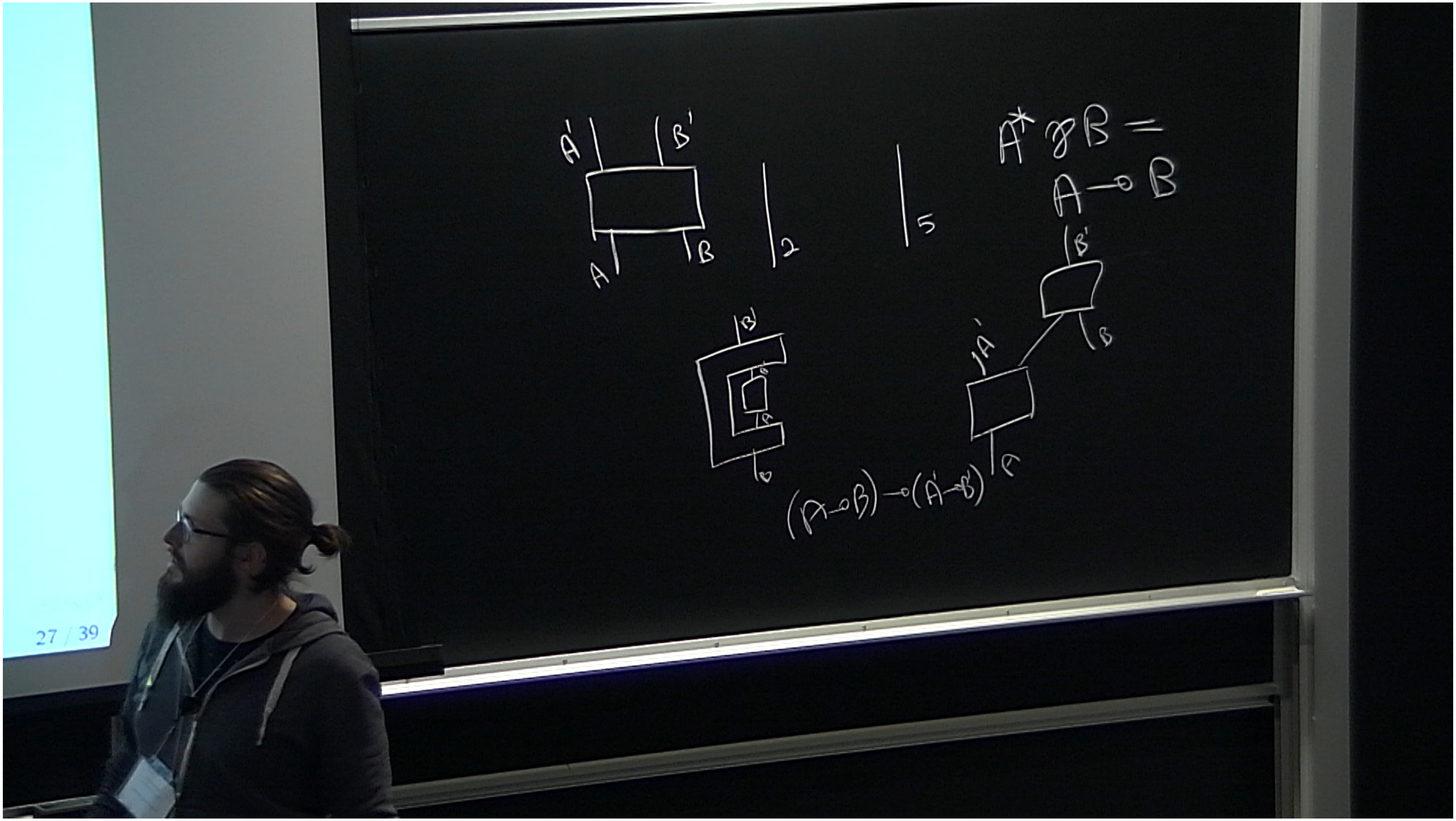


$$: \; (\boldsymbol{A}_1 \multimap \boldsymbol{A}'_1) \otimes \cdots \otimes (\boldsymbol{A}_n \multimap \boldsymbol{A}'_n)$$

- Quantum n-combs:



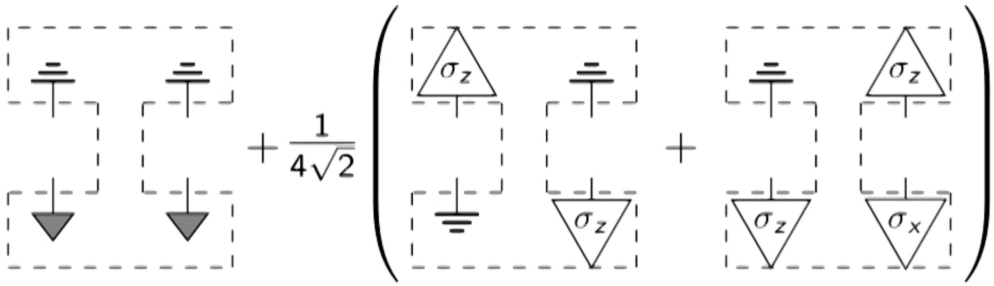
$$: \; \boldsymbol{A}_1 \multimap (\boldsymbol{A}'_1 \multimap (\cdots) \multimap \boldsymbol{A}_n) \multimap \boldsymbol{A}'_n$$



A'
B'
A
B
2
5
A* ⅋ B =
A ⊸ B
(A ⊸ B) ⊸ (A' ⊸ B')
27 / 39

Further examples

- Indefinite causal structures e.g. OCB W-process:



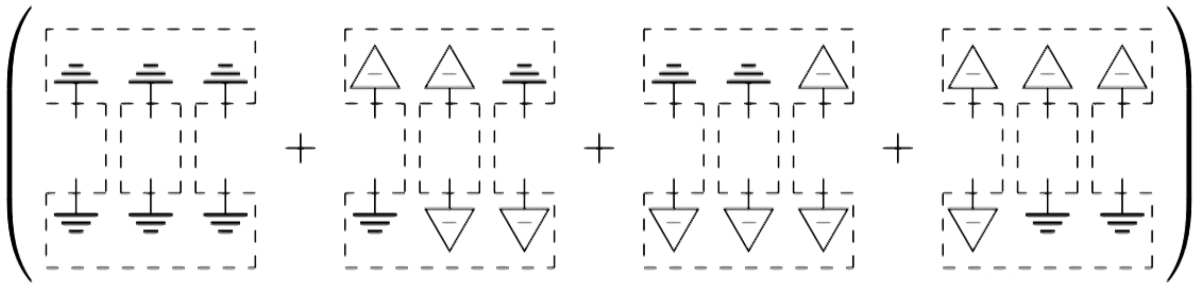
...and quantum switch are of type:

$$\left[(\boldsymbol{A}_0 \multimap \boldsymbol{A}_1) \otimes (\boldsymbol{B}_0 \multimap \boldsymbol{B}_1)\right]^* =: \mathbf{NS}(\boldsymbol{A}, \boldsymbol{B})^*$$

...in $\mathrm{Caus}[\mathbf{CPM}]$

Further examples

- Baumeler and Wolf give a 3-party *classical* process with indefinite causal structure:

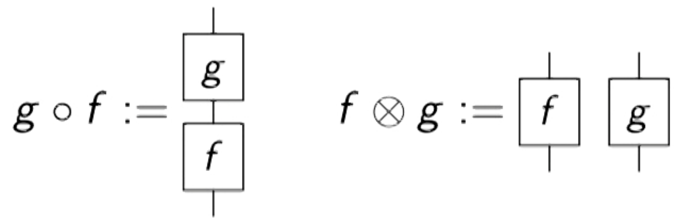
$\frac{1}{8}$ 

which is of type:

$$\left[(\boldsymbol{A} \multimap \boldsymbol{A}') \otimes (\boldsymbol{B} \multimap \boldsymbol{B}') \otimes (\boldsymbol{C} \multimap \boldsymbol{C}') \right]^*$$

Composing causal structures

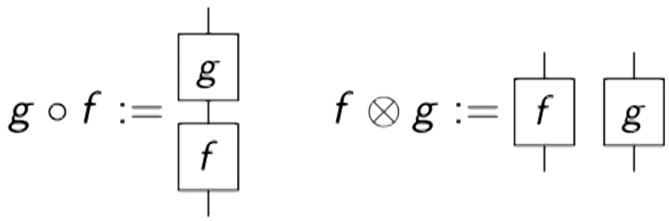
- First-order processes have well-defined operations $\circ$ and $\otimes$



...which suffice to build everything.

Composing causal structures

- First-order processes have well-defined operations $\circ$ and $\otimes$

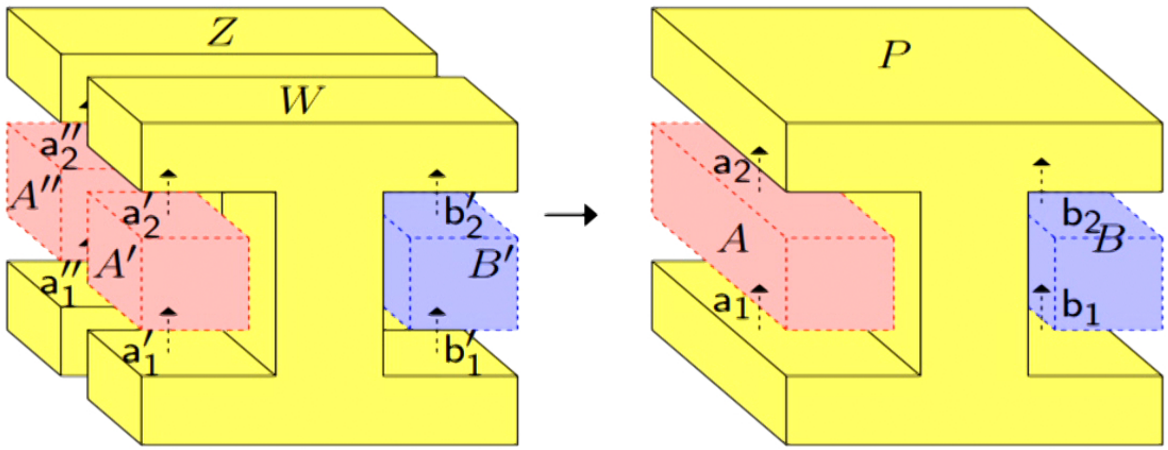


...which suffice to build everything.

- For higher-order processes, things are not so simple:
    - **Thing** $\otimes$ **Thing** is typically not a **Thing** again
    - Many ways to plug together, only some are well-defined.

Example: Tensor of two process matrices

...is not what we expect (or would like):



[1]

---

[1] Jia, Sakharwade. arXiv:1706.05532

[2] Guérin, Krumm, Budroni, Brukner. arXiv:1806.10374

Example: Tensor of two process matrices

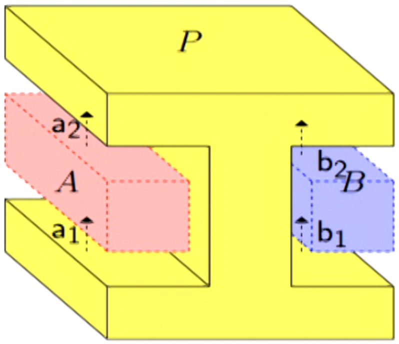
$$W_1 \otimes W_2 : \left[(\boldsymbol{A}_0 \multimap \boldsymbol{A}_1) \otimes (\boldsymbol{B}_0 \multimap \boldsymbol{B}_1) \right]^* \otimes \left[(\boldsymbol{A}'_0 \multimap \boldsymbol{A}'_1) \otimes (\boldsymbol{B}'_0 \multimap \boldsymbol{B}'_1) \right]^*$$

Example: Tensor of two process matrices

$$W_1 \otimes W_2 : \left[(\boldsymbol{A}_0 \multimap \boldsymbol{A}_1) \otimes (\boldsymbol{B}_0 \multimap \boldsymbol{B}_1) \right]^* \otimes \left[(\boldsymbol{A}'_0 \multimap \boldsymbol{A}'_1) \otimes (\boldsymbol{B}'_0 \multimap \boldsymbol{B}'_1) \right]^*$$

$$\not\cong$$

$$\left[(\boldsymbol{A}_0 \otimes \boldsymbol{A}'_0 \multimap \boldsymbol{A}_1 \otimes \boldsymbol{A}'_1) \otimes (\boldsymbol{B}_0 \otimes \boldsymbol{B}'_0 \multimap \boldsymbol{B}_1 \otimes \boldsymbol{B}'_1) \right]^*$$

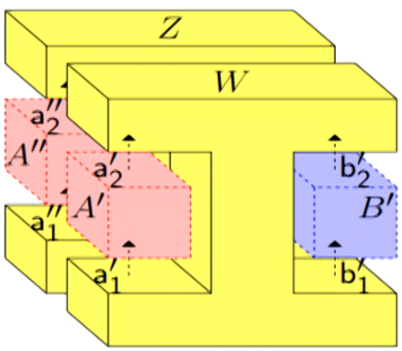


Example: Tensor of two process matrices

$$W_1 \otimes W_2 : \left[(\boldsymbol{A}_0 \multimap \boldsymbol{A}_1) \otimes (\boldsymbol{B}_0 \multimap \boldsymbol{B}_1)\right]^* \otimes \left[(\boldsymbol{A}'_0 \multimap \boldsymbol{A}'_1) \otimes (\boldsymbol{B}'_0 \multimap \boldsymbol{B}'_1)\right]^*$$

$$\not\cong$$

$$\left[(\boldsymbol{A}_0 \multimap \boldsymbol{A}_1) \otimes (\boldsymbol{B}_0 \multimap \boldsymbol{B}_1) \otimes (\boldsymbol{A}'_0 \multimap \boldsymbol{A}'_1) \otimes (\boldsymbol{B}'_0 \multimap \boldsymbol{B}'_1)\right]^*$$



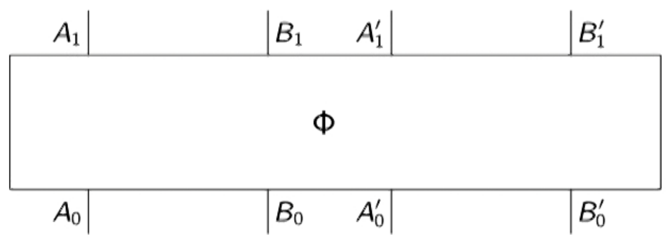
Example: Tensor of two process matrices

$$W_1 \otimes W_2 : \left[(\boldsymbol{A}_0 \multimap \boldsymbol{A}_1) \otimes (\boldsymbol{B}_0 \multimap \boldsymbol{B}_1) \right]^* \otimes \left[(\boldsymbol{A}'_0 \multimap \boldsymbol{A}'_1) \otimes (\boldsymbol{B}'_0 \multimap \boldsymbol{B}'_1) \right]^*$$

$$\cong$$

$$\left[\underbrace{((\boldsymbol{A}_0 \multimap \boldsymbol{A}_1) \otimes (\boldsymbol{B}_0 \multimap \boldsymbol{B}_1))}_{\text{n.s. sector}} \parr \underbrace{((\boldsymbol{A}'_0 \multimap \boldsymbol{A}'_1) \otimes (\boldsymbol{B}'_0 \multimap \boldsymbol{B}'_1))}_{\text{n.s. sector}} \right]^*$$

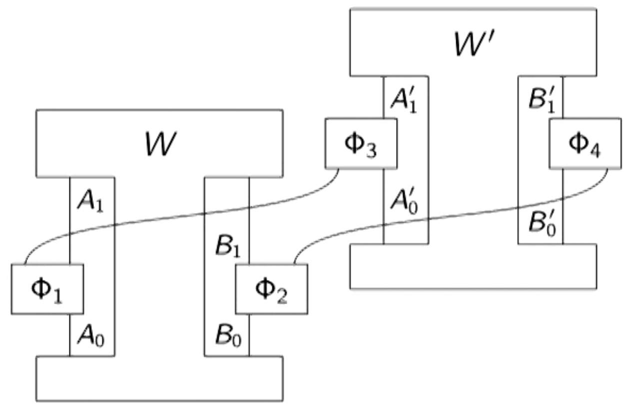
$$\Phi : \underbrace{((A_0 \multimap A_1) \otimes (B_0 \multimap B_1))}_{\text{n.s. sector}} \parr \underbrace{((A'_0 \multimap A'_1) \otimes (B'_0 \multimap B'_1))}_{\text{n.s. sector}}$$



$$(A_0 \multimap (A_1 \multimap A_0') \multimap A_1') \otimes (B_0 \multimap (B_1 \multimap B_0') \multimap B_1')$$

$$\Big\downarrow$$

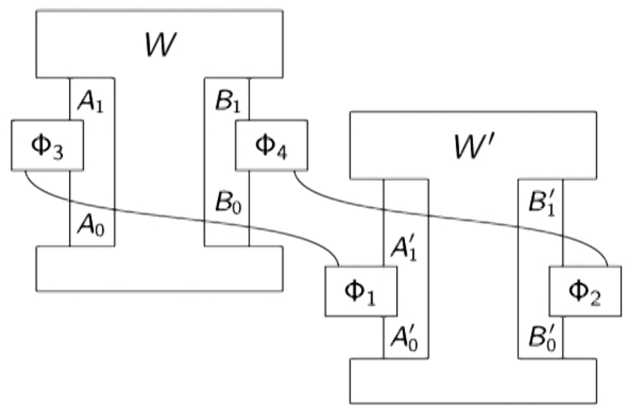
$$\underbrace{((A_0 \multimap A_1) \otimes (B_0 \multimap B_1))}_{\text{n.s. sector}} \parr \underbrace{((A_0' \multimap A_1') \otimes (B_0' \multimap B_1'))}_{\text{n.s. sector}}$$



$$(A_0' \multimap (A_1' \multimap A_0) \multimap A_1) \otimes (B_0' \multimap (B_1' \multimap B_0) \multimap B_1)$$

$$\downarrow$$

$$\underbrace{((A_0 \multimap A_1) \otimes (B_0 \multimap B_1))}_{\text{n.s. sector}} \parr \underbrace{((A_0' \multimap A_1') \otimes (B_0' \multimap B_1'))}_{\text{n.s. sector}}$$



Q: How can we tell if a plugging is allowed?

First decompose into signalling and non-signalling sectors:

Q: How can we tell if a plugging is allowed?

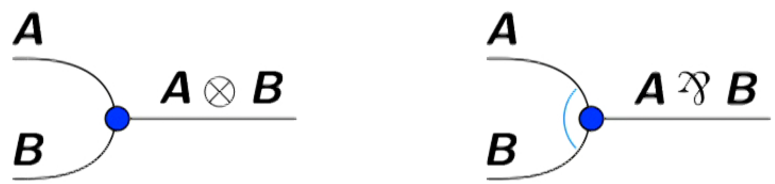
First decompose into signalling and non-signalling sectors:

Then re-compose with the opposite markings:

Q: How can we tell if a plugging is allowed?

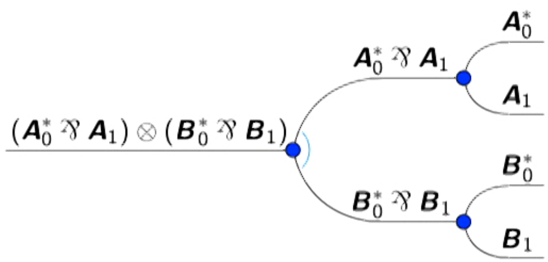
First decompose into signalling and non-signalling sectors:

Then re-compose with the opposite markings:

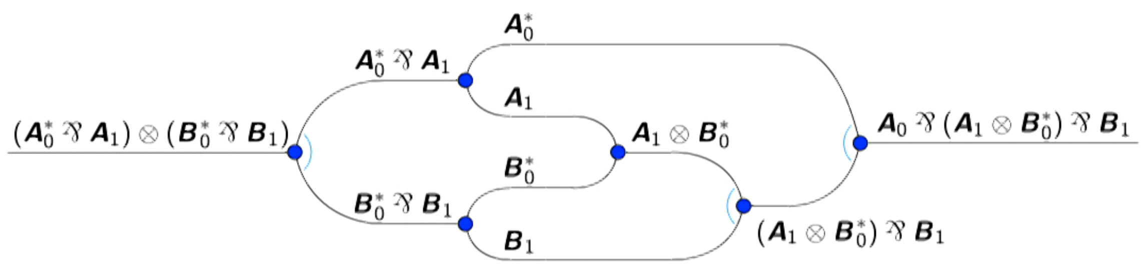


And check that there are no 'singalling loops'.

Example



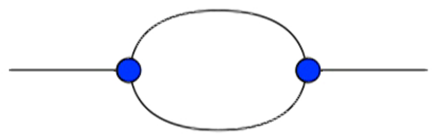
Example



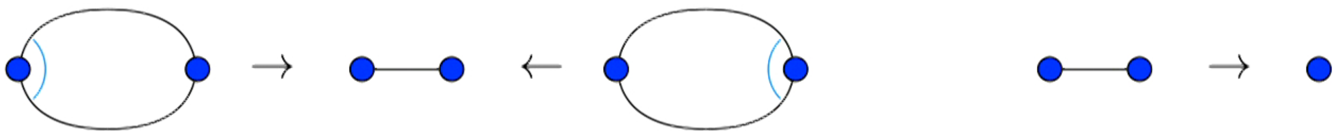
$(A_0^* \parr A_1) \otimes (B_0^* \parr B_1)$
$A_0^* \parr A_1$
A_0^*
A_1
$A_1 \otimes B_0^*$
B_0^*
$B_0^* \parr B_1$
B_1
$(A_1 \otimes B_0^*) \parr B_1$
$A_0 \parr (A_1 \otimes B_0^*) \parr B_1$

Q: How can we tell if a plugging is allowed?

Signalling loops break normalisation:

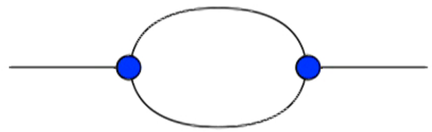


Check for them by applying these rules on the graph:

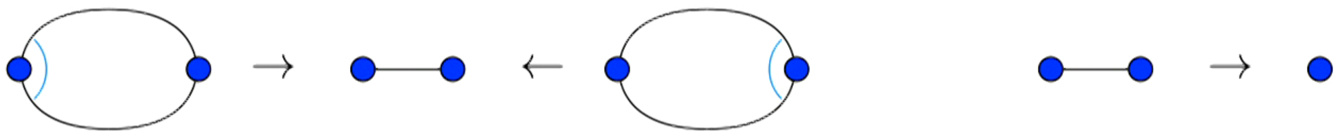


Q: How can we tell if a plugging is allowed?

Signalling loops break normalisation:



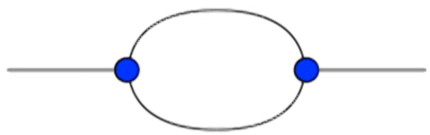
Check for them by applying these rules on the graph:



Contract to a point
$\implies$ no singalling loops
$\implies$ valid composition

...but why does it work?

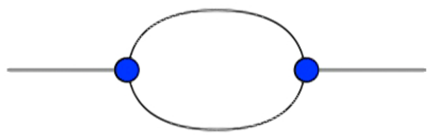
- 'Dot (de)compositions' are secretly *proof nets for multiplicative linear logic* (MLL), the type-logic of *-autonomous categories
- Contraction condition is equivalent to Danos-Regnier correctness condition for proof-nets



- MLL is sound for causal reasoning, regardless of dimension (or background theory...)

…but why does it work?

- ‘Dot (de)compositions’ are secretly *proof nets for multiplicative linear logic* (MLL), the type-logic of *-autonomous categories
- Contraction condition is equivalent to Danos-Regnier correctness condition for proof-nets



- MLL is sound for causal reasoning, regardless of dimension (or background theory...)
- **Open:** How much more for completeness?

Activities Google Chrome dec 9 14:50

git/hocc/notebooks/ | HOCC | Indefinite Causal Structur | akissinger/hocc: Experim

localhost:8888/notebooks/git/hocc/notebooks/HOCC.ipynb#

jupyter HOCC Last Checkpoint: 4 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

```
import math
import itertools
sys.path.append('..')
import hocc as cc
from hocc.variables import *
```

In [ ]:
```
a0 + a1
```

In [ ]:
```
def fo(X0,X1): return (X0 + X1) >> (X0 * X1)
def ns(X0, X1, Y0, Y1): return (X0 >> X1) * (Y0 >> Y1)
def fs(X0, X1, Y0, Y1): return X0 >> ((X1 >> Y0) >> Y1)
def fs1(X0, X1, Y0, Y1, Z0): return (X0 >> X1 * Z0) * (Z0 * Y0 >> Y1)
```

In [ ]:
```
g = cc.prove(
    (a0 >> a1) * (b0 >> b1),
    a0 >> ((a1 >> b0) >> b1)
)
if g: cc.draw(g)
```

In [ ]:
```
g.contract1()
cc.draw(g)
```

In [ ]:
```
g = cc.prove((a0 * b0) >> (a1 * b1), (a0 >> a1) + (b0 >> b1))
if g: cc.draw(g)
```

Activities Google Chrome dec 9 14:50

git/hocc/notebooks/ | HOCC | Indefinite Causal Structur | akissinger/hocc: Experim

localhost:8888/notebooks/git/hocc/notebooks/HOCC.ipynb#

jupyter HOCC Last Checkpoint: 4 hours ago (autosaved)

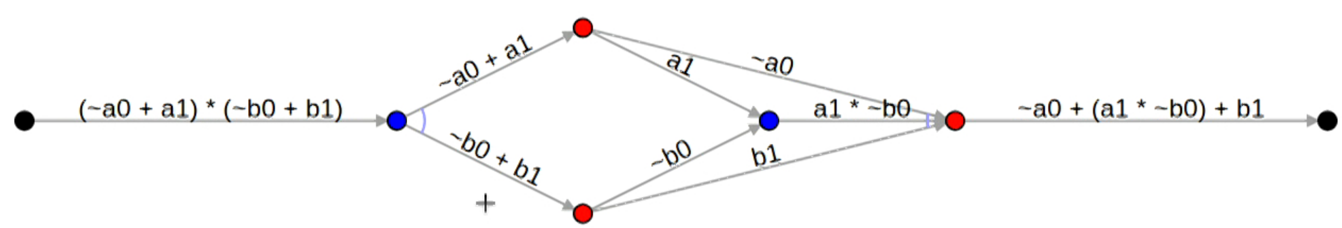
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

```
    a0 >> ((a1 >> b0) >> b1)
)
if g: cc.draw(g)
```




```
In [ ]: g.contract1()
        cc.draw(g)
```

Activities Google Chrome dec 9 14:51

git/hocc/notebooks/ | HOCC | Indefinite Causal Structur | akissinger/hocc: Experim

localhost:8888/notebooks/git/hocc/notebooks/HOCC.ipynb#

jupyter HOCC Last Checkpoint: 4 hours ago (unsaved changes)

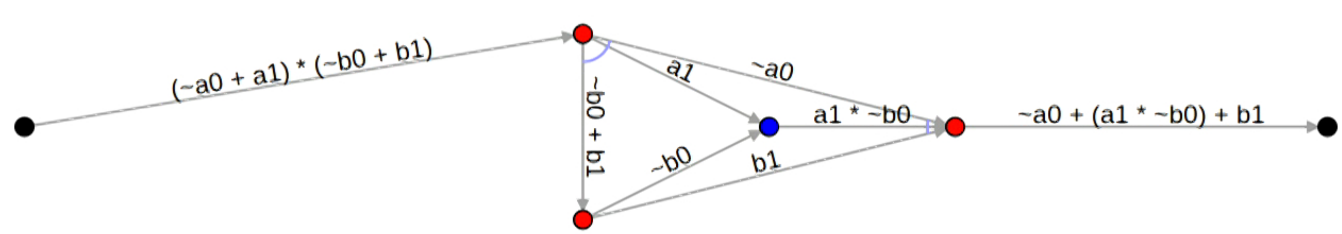
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

Run Code

In [12]:
```
g.contract1()
cc.draw(g)
```



jupyter HOCC Last Checkpoint: 4 hours ago (unsaved changes)

In [18]:
```
g.contract1()
cc.draw(g)
```

In [ ]:
```
g = cc.prove((a0 * b0) >> (a1 * b1), (a0 >> a1) + (b0 >> b1))
if g: cc.draw(g)
```

Activities Google Chrome dec 9 14:52

git/hocc/notebooks/ × HOCC × Indefinite Causal Structur × akissinger/hocc: Experim ×

localhost:8888/notebooks/git/hocc/notebooks/HOCC.ipynb#

jupyter HOCC Last Checkpoint: 4 hours ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [ ]:

```
g = cc.prove((a0 * b0) >> (a1 * b1), (a0 >> a1) + (b0 >> b1))
if g: cc.draw(g)
```

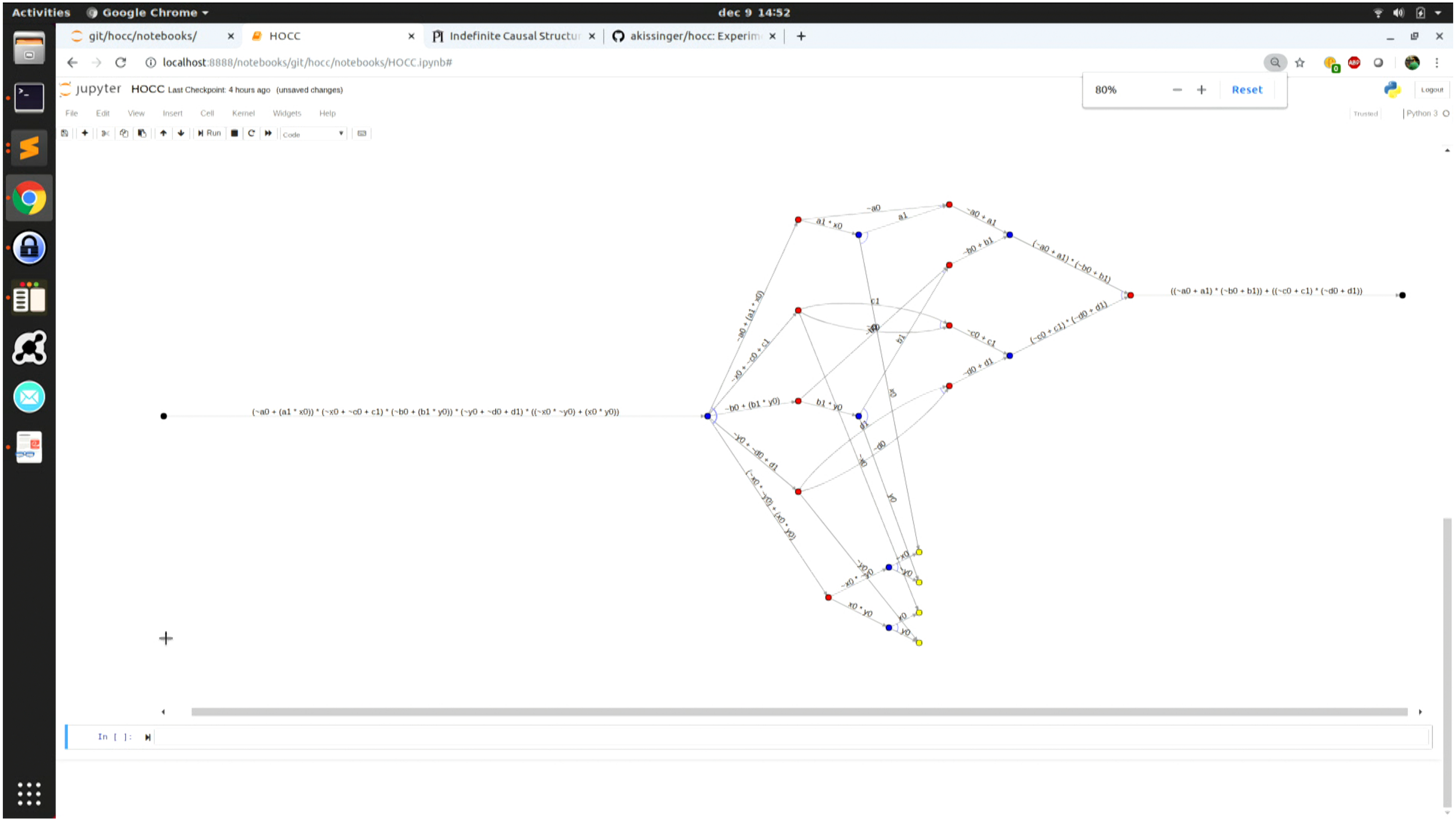
In [ ]:

```
g = cc.prove((a0 >> a1) + (b0 >> b1), (a0 * b0) >> (a1 * b1))
if g: cc.draw(g)
else: print("fail")
```

In [*]:

```
g = cc.prove(
      fs1(a0,a1,c0,c1,x0) *
      fs1(b0,b1,d0,d1,y0) *
      fo(x0,y0),
      ns(a0,a1,b0,b1) + ns(c0,c1,d0,d1),
      checker=cc.proofnet.switching_checker)
if g != None:
    cc.d3.draw(g)
```

In [ ]:


Activities
Google Chrome
dec 9 14:52
git/hocc/notebooks/
HOCC
Indefinite Causal Structur
akissinger/hocc: Experim
localhost:8888/notebooks/git/hocc/notebooks/HOCC.ipynb#
jupyter HOCC Last Checkpoint: 4 hours ago (unsaved changes)
80%
Reset
File Edit View Insert Cell Kernel Widgets Help
Run
Code
Logout
Python 3
(~a0 + (a1 * x0)) * (~x0 + ~c0 + c1) * (~b0 + (b1 * y0)) * (~y0 + ~d0 + d1) * ((~x0 * ~y0) + (x0 * y0))
((~a0 + a1) * (~b0 + b1)) + ((~c0 + c1) * (~d0 + d1))
In [ ]:

Thanks

github.com/akissinger/hocc

- *A categorical semantics for causal structure.* Aleks Kissinger and Sander Uijlen. Logical Methods in Computer Science, 15(3). arXiv:1701.04732
- *Causal structures and the classification of higher order quantum computation.* Paulo Perinotti. arXiv:1612.05099
- *Tensor products of process matrices with Indefinite Causal Structure.* Ding Jia, Nitica Sakharwade. Phys. Rev. A. 97, 032110. 2018
- *Composition rules for quantum processes: a no-go theorem.* Guérin, Krumm, Budroni, Brukner. arXiv:1806.10374
- *Proof Nets and the Identity of Proofs.* Lutz Straßburger. INRIA Rapport de recherche no. 6013. 2006.