Diagrammatic Semantics for Symmetric Traced Monoidal Categories

George Kaye

04 March 2021

University of Birmingham

What’s the point?

Compositional processes

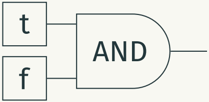
We can model compositional processes.

$$\boxed{F}$$

$$\boxed{F} \cdot \boxed{G}$$

$$\begin{array}{c} \boxed{F} \\ \otimes \\ \boxed{G} \end{array}$$

Compositional processes

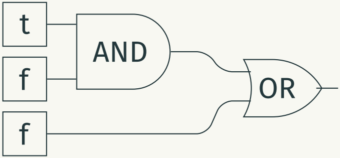


$t \otimes f \cdot \mathsf{AND}$

$\rightsquigarrow$

f

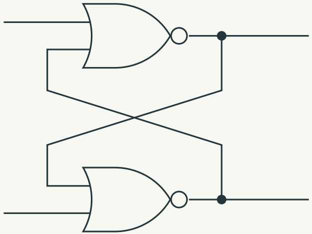
Compositional processes



$$((t \otimes f \cdot \mathsf{AND}) \otimes f) \cdot \mathsf{OR} \quad \rightsquigarrow \quad (f \otimes f) \cdot \mathsf{OR} \quad \rightsquigarrow \quad f$$

$$(t \otimes f \otimes f) \cdot (\mathsf{AND} \otimes \mathrm{id}_1) \cdot \mathsf{OR} \quad \rightsquigarrow \quad ?$$

Compositional processes



$$\mathsf{Tr}^1((\sigma_{1,1} \cdot \mathsf{NOR} \cdot \prec) \otimes \mathsf{id}_1 \cdot \mathsf{id}_1 \otimes (\mathsf{NOR} \cdot \prec) \cdot \sigma_{1,1} \otimes \mathsf{id}_1)$$

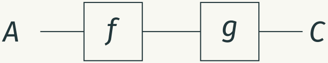
Graphical languages for monoidal categories

Generators

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

Composition

$$f \cdot g : A \to C$$



Categories – identity morphisms

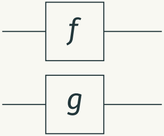
$$\mathrm{id}_A : A \to A \qquad \mathrm{id}_B : B \to B$$

$$\mathrm{id}_A \cdot f : A \to B \qquad f : A \to B \qquad f \cdot \mathrm{id}_B : A \to B$$

Monoidal categories

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

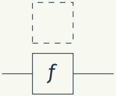
$f \otimes g : A \otimes C \to B \otimes D$



Monoidal categories – monoidal unit

$$\mathrm{id}_I \;:\; I \;\to\; I$$

$$\mathrm{id}_I \otimes f \;:\; I \otimes A \;\to\; I \otimes B$$

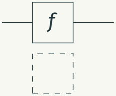


$$=$$

$$f \;:\; A \;\to\; B$$

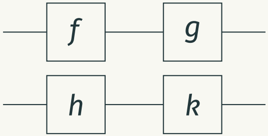
$$=$$

$$f \otimes \mathrm{id}_I \;:\; A \otimes I \;\to\; B \otimes I$$

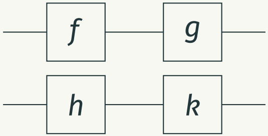


Monoidal categories – functoriality

$$(f \otimes h) \cdot (g \otimes k) \;:\; A \otimes D \;\rightarrow\; C \otimes F$$



$$(f \cdot g) \otimes (h \cdot k) \;:\; A \otimes D \;\rightarrow\; C \otimes F$$



We did it – bureaucracy is no more!

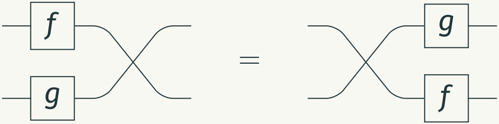
Symmetric monoidal categories

Symmetry

$$\sigma_{A,B} \ : \ A \otimes B \ \to \ B \otimes A$$

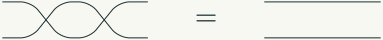
Symmetric monoidal categories – axioms

Naturality



Hexagon

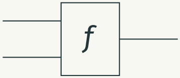
Self-inverse



PROPs

A PROP is a monoidal category where the objects are natural numbers and tensor product is addition.

$$f : 2 \to 1$$



Free categories

A signature is a set of generators.

$$\{\prec\ :\ 1 \to 2, \succ\ :\ 2 \to 1\}$$

We create terms by combining generators.

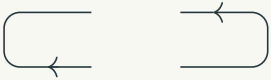
$$\prec \otimes \mathrm{id}_1 \cdot \mathrm{id}_1 \otimes \sigma_{1,1} \cdot \succ \otimes \mathrm{id}_1$$

Bending the wires

All our wires have gone from left to right.

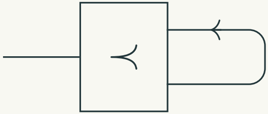
Can we bend them?

We can in a compact closed category.



Bending the wires

Compact closed categories have flexible causality.

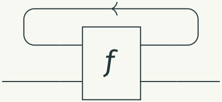


In some cases this is bad.

Symmetric traced monoidal categories

The trace is a single atomic action.

$$f : X \otimes M \to X \otimes N \quad \Rightarrow \quad \mathrm{Tr}^X(F) : M \to N$$



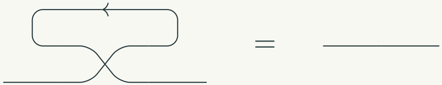
Symmetric traced monoidal categories – axioms

Tightening

$$\mathrm{Tr}^X(\mathrm{id}_X \otimes g \cdot f \cdot \mathrm{id}_X \otimes h) = g \cdot \mathrm{Tr}^X(f) \cdot h$$

Yanking

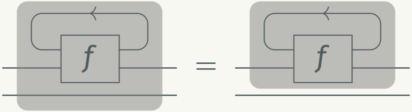
$$\mathrm{Tr}^X(\sigma_{X,X}) = \mathrm{id}_X$$



Symmetric traced monoidal categories – axioms

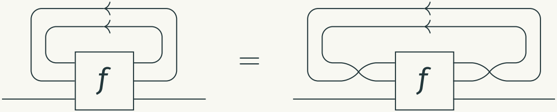
Superposing

$$\mathrm{Tr}^X(f \otimes \mathrm{id}_A) = \mathrm{Tr}^X(f) \otimes \mathrm{id}_A$$



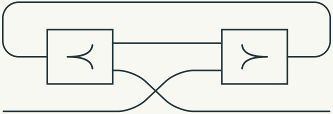
Exchange

$$\mathrm{Tr}^Y(\mathrm{Tr}^X(f)) = \mathrm{Tr}^X(\mathrm{Tr}^Y(\sigma_{Y,X} \otimes \mathrm{id}_A \cdot f \cdot \sigma_{X,Y} \otimes \mathrm{id}_A))$$



Free traced categories

$$\mathrm{Tr}^1(\prec \otimes \mathrm{id}_1 \cdot \mathrm{id}_1 \otimes \sigma_{1,1} \cdot \succ \otimes \mathrm{id}_1)$$



From here on, we will fix an arbitrary traced PROP $\mathbf{Term}_\Sigma$, generated freely over some signature Σ.

Adding extra structure

Graphical calculi ‘absorb’ the painful axioms of categories.

Does this solve all of our problems?

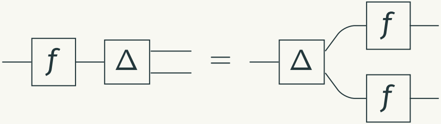
No.

We often work in categories with extra structure.

Adding extra structure

Naturality of Cartesian copy

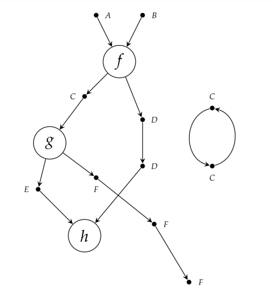
$$f \cdot \Delta_n = \Delta_n \cdot f \otimes f$$



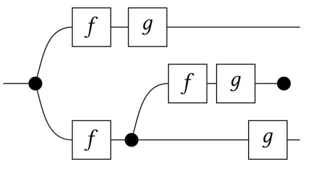
We need to define our graphs a little more rigorously...

Combinatorial diagrams

String graphs

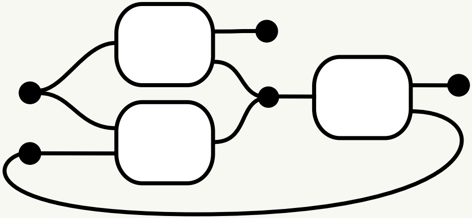


Hypergraphs



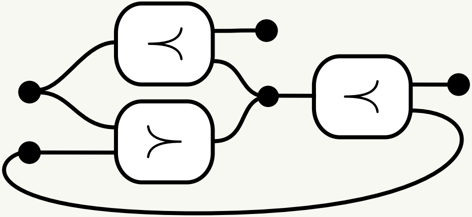
Hypergraphs

Hypergraphs

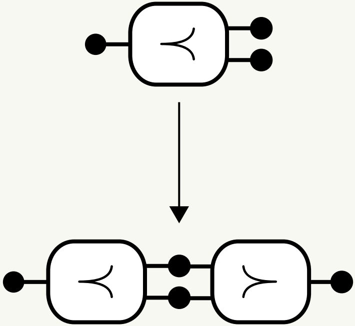


Labelled hypergraphs

$$\Sigma = \{\prec \ : \ 1 \ \to \ 2, \succ \ : \ 2 \ \to \ 1\}.$$



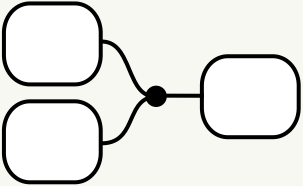
Hypergraph morphisms



Bijective maps $\Rightarrow$ isomorphism.

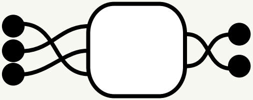
Hypergraphs are not enough

Splitting wires



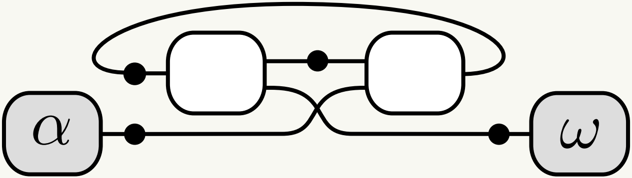
not allowed!

Interfaces



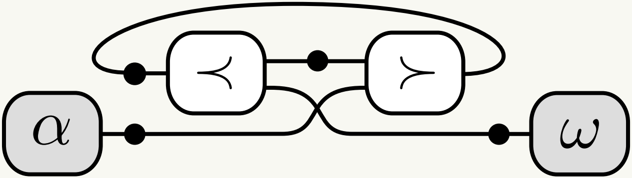
how?

Interfaced linear hypergraphs

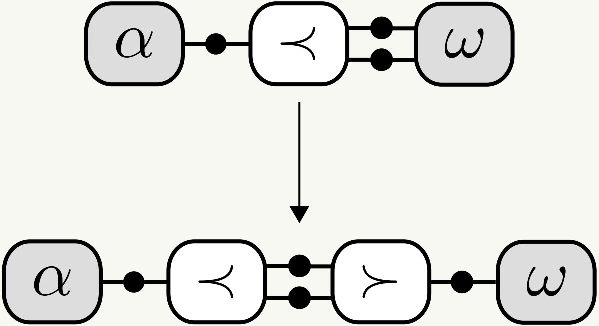


Labelled interfaced linear hypergraphs

$$\Sigma = \{\prec \ : \ 1 \ \to \ 2, \succ \ : \ 2 \ \to \ 1\}$$



Interfaced linear hypergraph morphisms



Bijective maps + preserves interface $\Rightarrow$ isomorphism $\equiv$.

Soundness and completeness

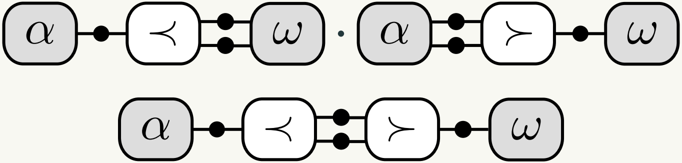
Soundness

Equal terms
in the category

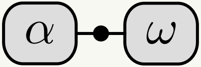
$\Rightarrow$ Isomorphic interpretations
as hypergraphs

Composition

$\prec \cdot \succ \ : \ 1 \ \to \ 1$

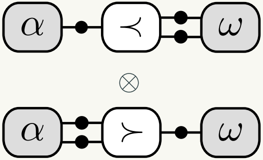


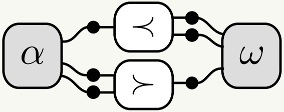
$\mathrm{id}_1 : 1 \to 1$



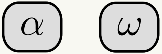
Monoidal tensor

$$\prec \otimes \succ \;:\; 1+2 \;\to\; 2+1$$



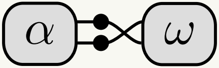


$$\mathrm{id}_0 \;:\; 0 \;\to\; 0$$



Symmetry

$$\sigma_{1,1} \;:\; 2 \;\to\; 2$$



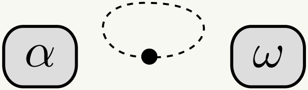
We can build up larger symmetries by composing symmetries and identities.

The problem with trace

Trace of the identity

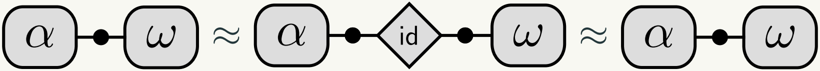
$$\mathrm{id}_A : A \to A \quad \xrightarrow{\mathrm{Tr}^A} \quad \mathrm{Tr}^A(\mathrm{id}_A) : I \to I$$

But...



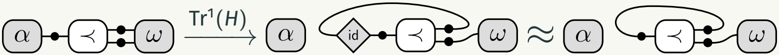
not allowed!

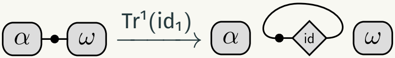
Homeomorphism



Trace

Trace is defined recursively over the number of wires.



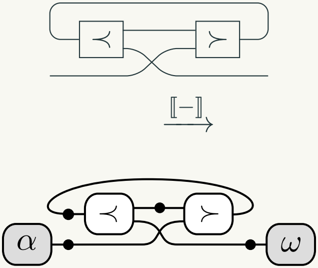


Interpreting terms as graphs

We assemble our hypergraphs into a traced PROP $\mathbf{HypTerm}_\Sigma$.

$$[\![-]\!] \;:\; \mathbf{Term}_\Sigma \;\to\; \mathbf{HypTerm}_\Sigma$$

Interpreting terms as graphs



Soundness

Theorem (Soundness)

For any morphism $f, g \in \textbf{Term}_\Sigma$, if $f = g$ under the equational theory of the category, then their interpretations as linear hypergraphs are isomorphic.

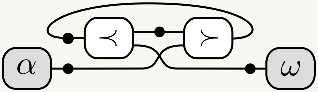
Definability

An interfaced linear hypergraph

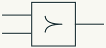
$\Rightarrow$ A set of corresponding terms in the category

Stacking

First we set an order $\leq$ on our edges and stack them.

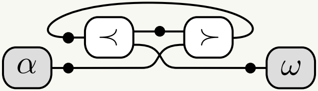


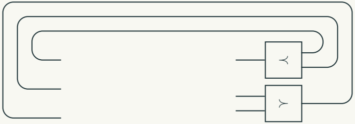
$$\prec \otimes \succ$$



Tracing

Then we trace around all the outputs of the stack:

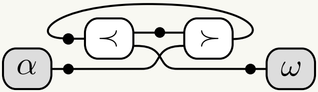


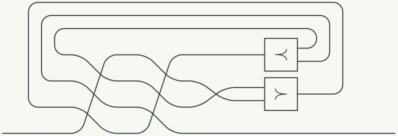


$$\mathrm{Tr}^3(?\cdot \prec \otimes \succ)$$

Shuffling

We then connect everything up:





$$\mathsf{Tr}^3(\sigma_{3,1} \cdot \sigma_{3,1} \cdot \mathsf{id}_1 \otimes \sigma_{1,1} \otimes \mathsf{id}_1 \cdot \prec \otimes \succ \otimes \mathsf{id}_1)$$

(**Exercise:** follow around the wires, make sure this is correct)

Definability

$$\langle\!\langle - \rangle\!\rangle \;:\; \mathbf{HypTerm}_{\Sigma} \;\to\; \mathbf{Term}_{\Sigma}$$

Proposition (Definability)

For every well-formed hypergraph F then $[\![\langle\!\langle F \rangle\!\rangle]\!] \equiv F$.

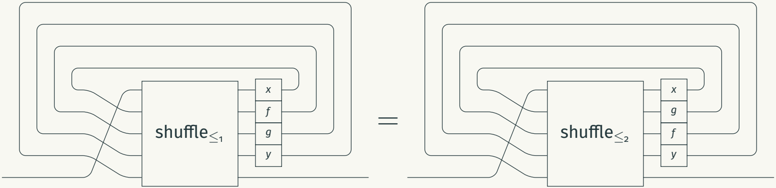
Coherence

But we cannot conclude completeness yet!

	An interfaced linear hypergraph
$\Rightarrow$	**Unique** morphism in the category, **up to the equational theory**

Coherence

Fortunately, we just need to show it for swapping two edges.



Proposition (Coherence)

For all orderings of edges $\leq_x$ on a hypergraph F,

$$\langle\!\langle F \rangle\!\rangle_{\leq_1} = \langle\!\langle F \rangle\!\rangle_{\leq_2} = \cdots = \langle\!\langle F \rangle\!\rangle_{\leq_n}$$

Completeness

Theorem (Completeness I)

For any interfaced linear hypergraph H, $[\![\langle\!\langle H \rangle\!\rangle]\!] \equiv H$.

Theorem (Completeness II)

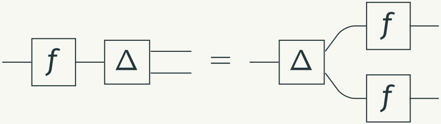
For any morphism $f \in \textbf{\textit{Term}}_{\Sigma}$, $\langle\!\langle [\![f]\!] \rangle\!\rangle = f$.

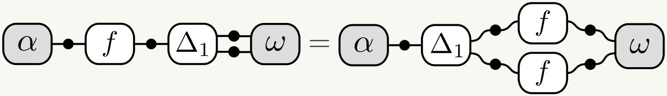
Graph rewriting

We express extra structure as additional axioms.

These axioms can be expressed as rewrite rules.

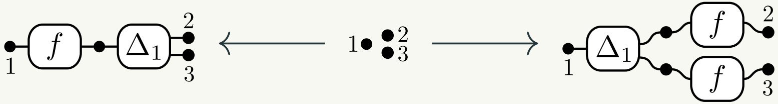
$$f \cdot \Delta_n = \Delta_n \cdot f \otimes f$$





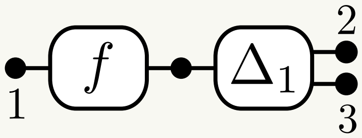
Rewrite rules

First let's see how it works with normal hypergraphs.

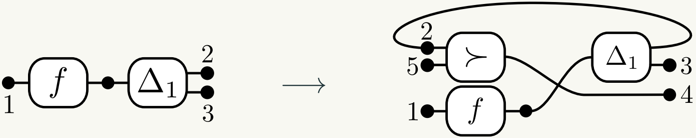


For a set of axioms $\mathcal{E} \in \textbf{Term}_\Sigma$, we write $[\![\mathcal{E}]\!]$ for their conversion into spans like this.

DPO rewriting

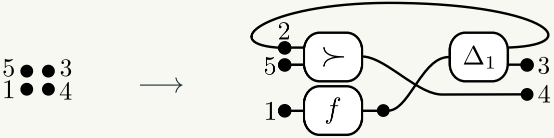


First we identify a matching morphism.

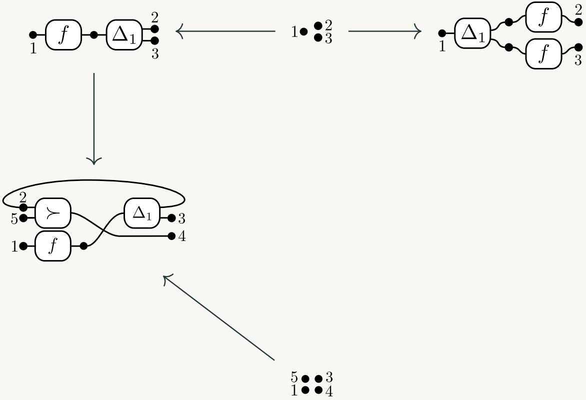


DPO rewriting

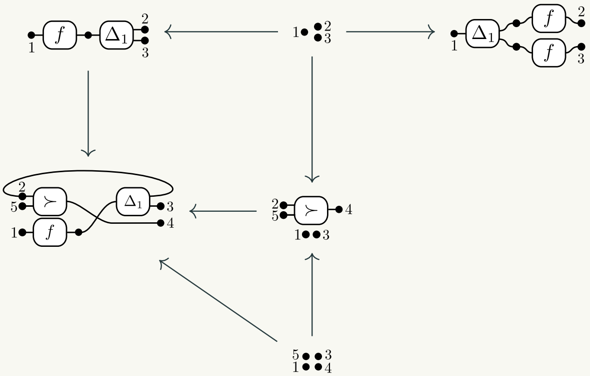
We also need an explicit morphism to denote the interfaces.



DPO rewriting

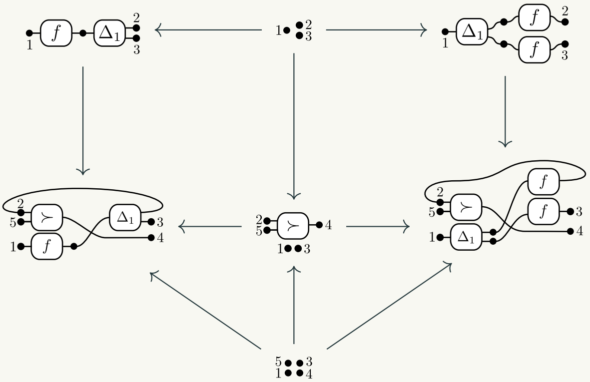


We then compute the pushout complement.



DPO rewriting

Then we perform a pushout on $C \leftarrow K \rightarrow R$.



We write $G \rightsquigarrow_{[\![\mathcal{E}]\!]} H$ if rewriting can be performed in this way.

Not all structures are compatible with DPO rewriting.

The framework of adhesive categories is often used to ensure that pushout complements are always unique, if they exist.

Proposition

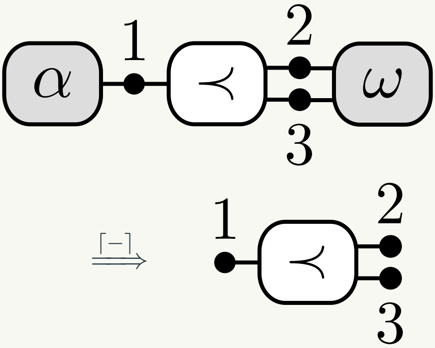
The category of (vanilla) hypergraphs is adhesive.

Adhesive categories

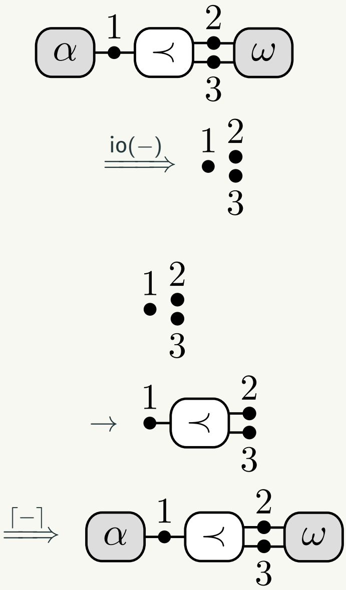
Unfortunately, $\mathbf{LHyp}_{\Sigma}$ is not adhesive...

We'll just do the rewriting in **Hyp** instead!

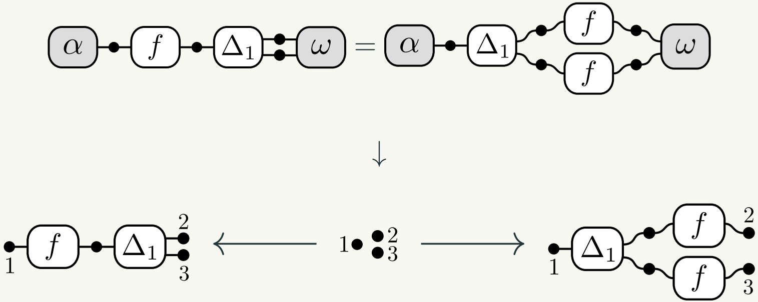
Trimming



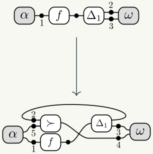
Reinterfacing



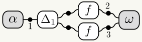
Rewrite rule



Rewriting

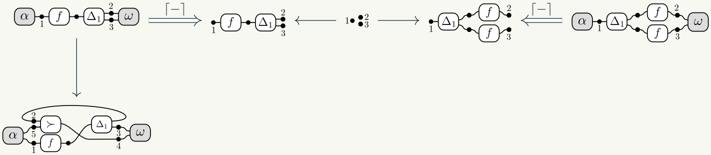


α
1
f
Δ_1
2
3
ω
2
5
1
$\succ$
f
Δ_1
3
4
α
ω

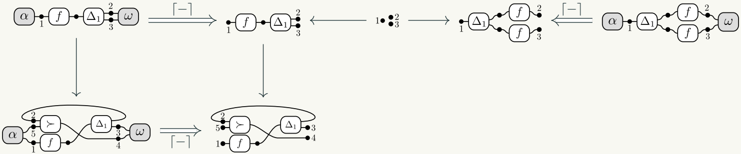


α
1
Δ_1
f
f
2
3
ω

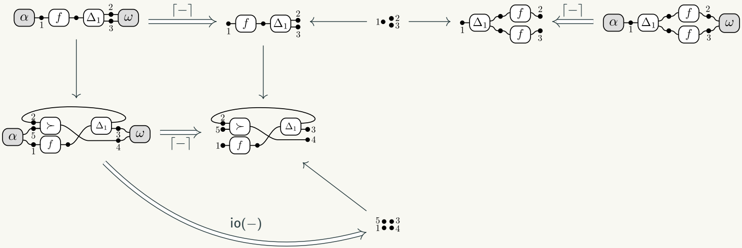
Rewriting



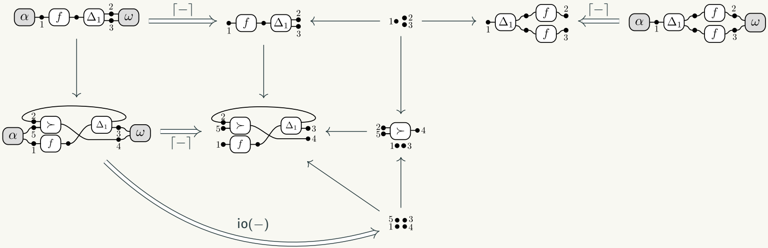
Rewriting



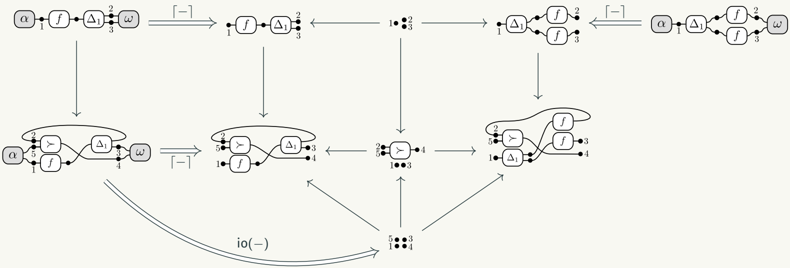
Rewriting



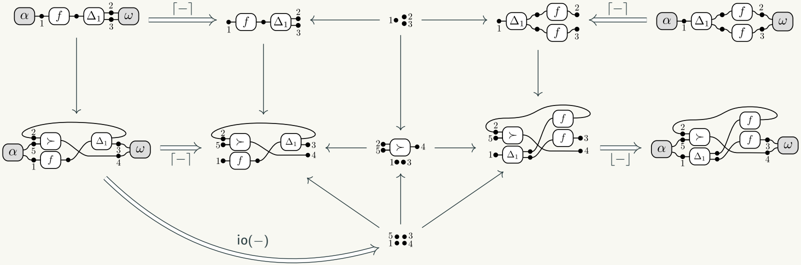
Rewriting



Rewriting



Rewriting

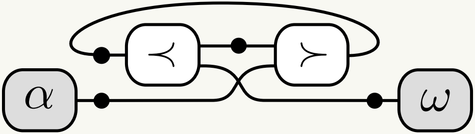


Generalisation

Can we generalise to arbitrary STMCs?

Just add vertex labels!

Conclusion



We have a sound and complete graphical language for STMCs.

We can reason in STMCs purely graphically.

We can add extra axioms using graph rewrites.

Just formulate the axioms as rewrite rules.

F. Bonchi, F. Gadducci, A. Kissinger, P. Sobociński, and F. Zanasi.
Rewriting modulo symmetric monoidal structure.
In *2016 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2016.

H. Ehrig, M. Pfender, and H. J. Schneider.
Graph-grammars: An algebraic approach.
In *14th Annual Symposium on Switching and Automata Theory (swat 1973)*, pages 167–180. IEEE, 1973.

M. Hasegawa.
On traced monoidal closed categories.
Mathematical Structures in Computer Science, 19(2):217–244, 2009.

A. Kissinger.
Pictures of processes: Automated graph rewriting for monoidal categories and applications to quantum computing, 2012.

P. Selinger.
A survey of graphical languages for monoidal categories.
In *New structures for physics*, pages 289–355. Springer, 2010.