UNIVERSITY OF LEICESTER

Department of Informatics

Presenting finite posets (with monotone maps?)

... using string diagrams

Sam Balco

Thursday 22nd June, 2017

Outline

What are string diagrams?

String diagrams algebraically

Presenting finite sets and functions

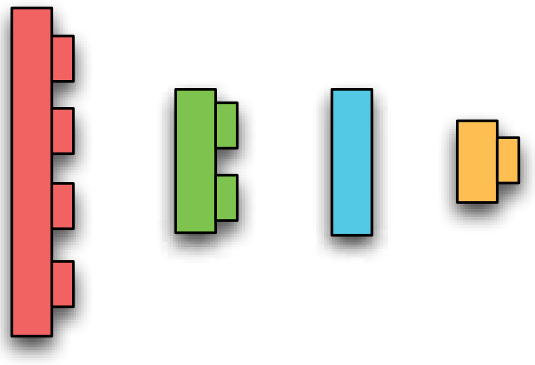
Refresher on posets

Presenting posets

Presenting posets and monotone functions?

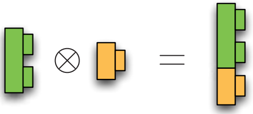
What are string diagrams?

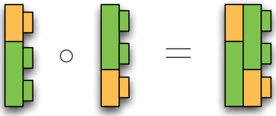
They are a bit like legos!

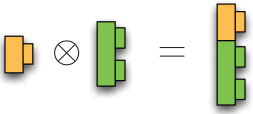


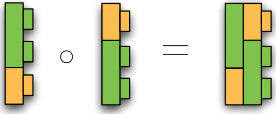
What are string diagrams?

We can combine legos/string diagrams in two different ways:



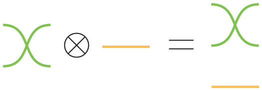


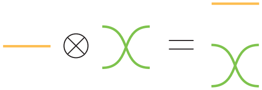


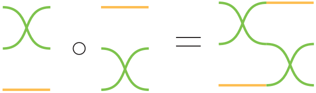


What are string diagrams?

We can combine legos/string diagrams in two different ways:







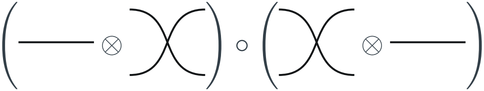
String diagrams algebraically

We can assign each string a type, given by the number of input ports (on the left) and the number of output ports on the right:

$id : 1 \to 1 \qquad \gamma : 2 \to 2 \qquad \mu : 2 \to 1 \qquad \eta : 0 \to 1$

$$\frac{S : k \to l \qquad T : m \to n}{S \otimes T : k + m \to l + n} \qquad\qquad \frac{S : k \to l \qquad T : l \to m}{S \circ T : k \to m}$$

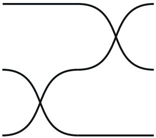
String diagrams algebraically



$$=$$

$$(id \otimes \gamma) \circ (\gamma \otimes id)$$

$$=$$



Presenting finite sets and functions

So what can we do with id, γ, μ and η?

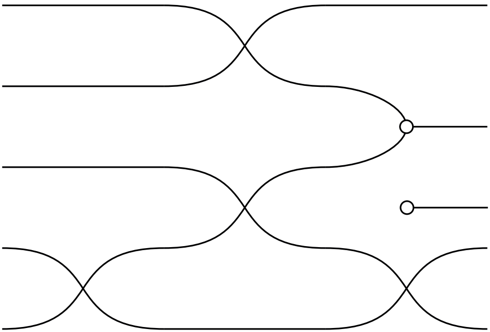
Say we have function $f : [5] \to [5]$ (where $[n] = \{0 \ldots n-1\}$):

$$
\begin{aligned}
f(0) &= 1 \\
f(1) &= 0 \\
f(2) &= 4 \\
f(3) &= 3 \\
f(4) &= 1
\end{aligned}
$$

Presenting finite sets and functions

So what can we do with *id*, γ, μ and η?

We can express this function as a string diagram:



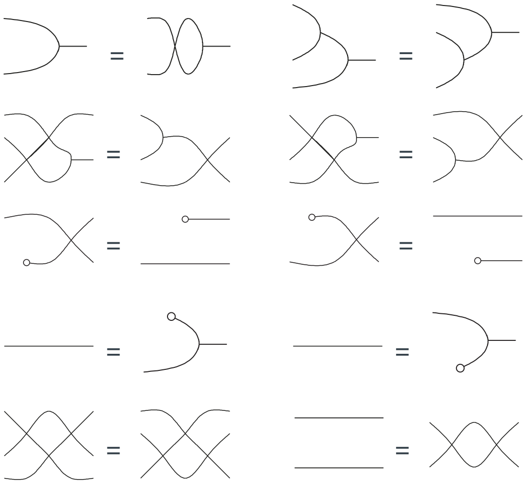
Presenting finite sets and functions

So what can we do with id, γ, μ and η?

Or alternatively:

$$(id \otimes id \otimes id \otimes \gamma) \circ$$
$$(\gamma \otimes \gamma \otimes id) \circ$$
$$(id \otimes \mu \otimes \eta \otimes \gamma)$$

String diagram equations



Presenting finite sets and functions

More formally, the category of functions on finite sets **F**, is *presented by* the string diagram rewriting system.

Our category **F** has natural numbers $n, m \in \mathbb{N}$ as objects and arrows $f_{\mathbf{F}} : n \to m$, which are functions $f : [n] \to [m]$.

We can turn **F** into a symmetric monoidal category, by defining a bifunctor $\otimes : \mathbf{F} \times \mathbf{F} \to \mathbf{F}$ and pick an identity object **I**.

We can chose 0 as our identity object (remember objects in **F** are natural numbers) and define $\otimes$ to be $n \otimes m = n + m$ on objects and "disjoint union" on functions, namely:

Presenting finite sets and functions

Given arrows $f_{\mathbf{F}} : k \to l$ and $g_{\mathbf{F}} : m \to n$, which are the functions $f : [k] \to [l]$ and $g : [m] \to [n]$, we define the arrow $f_{\mathbf{F}} \otimes g_{\mathbf{F}} : k + m \to l + n$ to be the function:

$$f \otimes g : [k+m] \to [l+n]$$
$$f \otimes g\ (x) = \begin{cases} f(x) & \text{if } x < k \\ g(x-k)+l & \textit{otherwise} \end{cases}$$

Essentially stacking one function disjointly on top of the other.

Presenting finite sets and functions

What do we get, if we remove γ from our theory?

What do we get, if we remove μ?

What do we get, if we remove η?

What do we get, if we remove μ and η?

Presenting finite sets and functions

What do we get, if we remove γ from our theory?

Linear orders and monotone maps

What do we get, if we remove μ?

What do we get, if we remove η?

What do we get, if we remove μ and η?

Presenting finite sets and functions

What do we get, if we remove γ from our theory?
Linear orders and monotone maps

What do we get, if we remove μ?
Finite sets and injective functions

What do we get, if we remove η?

What do we get, if we remove μ and η?

Presenting finite sets and functions

What do we get, if we remove γ from our theory?
Linear orders and monotone maps

What do we get, if we remove μ?
Finite sets and injective functions

What do we get, if we remove η?
Finite sets and surjective functions

What do we get, if we remove μ and η?

Presenting finite sets and functions

What do we get, if we remove γ from our theory?
Linear orders and monotone maps

What do we get, if we remove μ?
Finite sets and injective functions

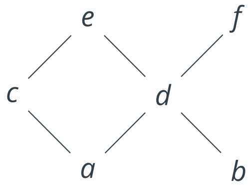
What do we get, if we remove η?
Finite sets and surjective functions

What do we get, if we remove μ and η?
Finite sets and bijective functions / Permutations

Refresher on posets

A *poset* is a set S, together with a partial order relation $\leq_S : S \times S$, which is *reflexive*, *transitive* and *antisymmetric*.

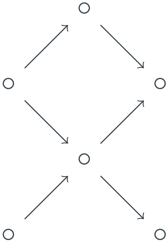
For example, for $S = \{a, b, c, d, e\}$ and $a \leq_S c, a \leq_S d, b \leq_S d$, $c \leq_S e, d \leq_S e$, we can draw the following diagram, representing the poset:



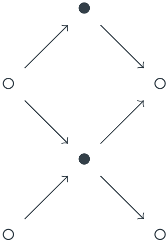
Presenting posets

$$
\begin{tikzcd}
& c \arrow[dr] & \\
a \arrow[ur] \arrow[dr] & & e \\
& d \arrow[ur] \arrow[dr] & \\
b \arrow[ur] & & f
\end{tikzcd}
$$

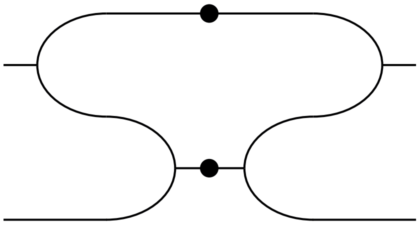
Presenting posets



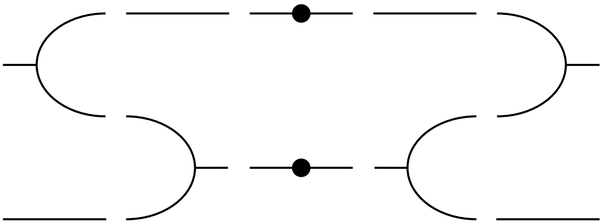
Presenting posets



Presenting posets



Presenting posets



Presenting posets

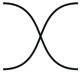
$$(\delta \otimes id) \circ (id \otimes \mu) \circ (\sigma \otimes \sigma) \circ (id \otimes \delta) \circ (\mu \otimes id)$$

Presenting posets

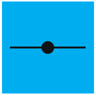
We have some additional combinators in this theory:

$id : 1 \to 1$

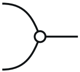
$\gamma : 2 \to 2$



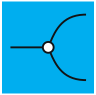
$\sigma : 1 \to 1$



$\mu : 2 \to 1$

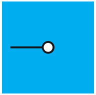


$\delta : 1 \to 2$

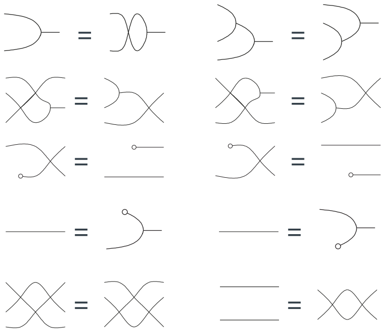


$\eta : 0 \to 1$

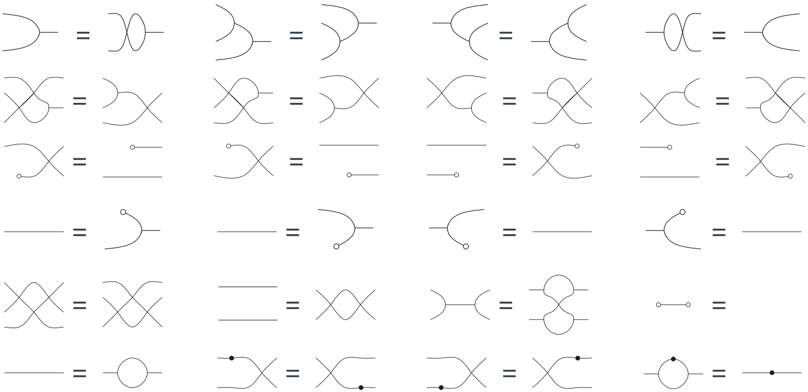
$\varepsilon : 1 \to 0$



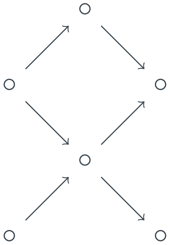
String diagram equations

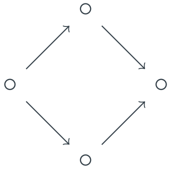


String diagram equations

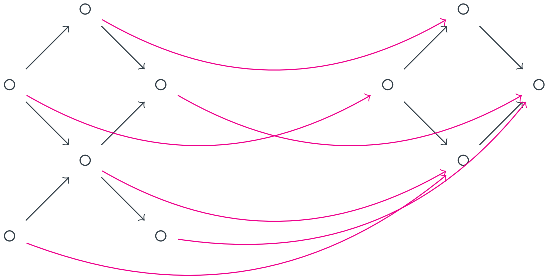


Presenting posets and monotone functions?

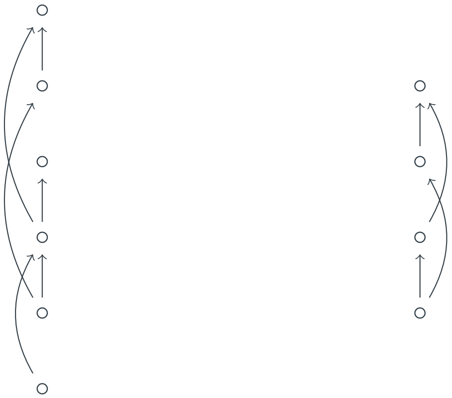




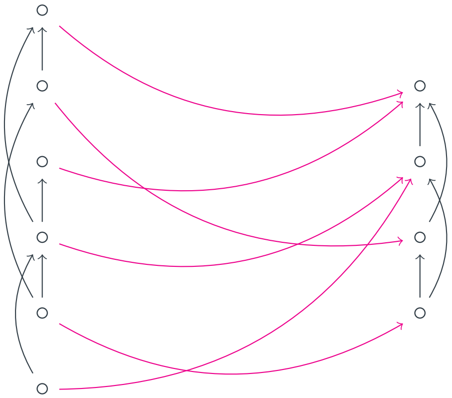
Presenting posets and monotone functions?



Presenting posets and monotone functions?



Presenting posets and monotone functions?



Who's stuff I ~~stole~~ borrowed

Drawing string diagrams is quite tedious!

I therefore borrowed some excellent ones from Pawel Sobocinski's MGS 2017 lecture notes (also, checkout his blog on Graphical Linear Algebra!)
(p. 3-4)

I also used some string diagrams from S. Mimram's paper Presenting Finite Posets, on which this presentation was based
(p. 11,27)

UNIVERSITY OF
LEICESTER