Cylinder Diagrams: Visualising Multi-parameter Natural Transformations

Niels Voorneveld
Tallinn University of Technology

September 9, 2022
SYCO 9

European Union
European Social Fund
Investing
in your future

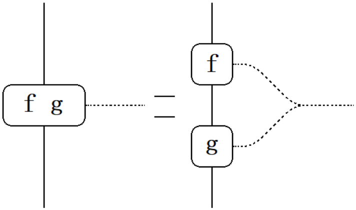
Things I like:

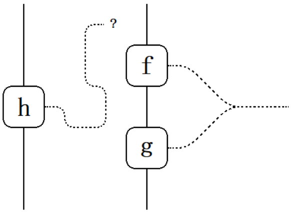
- Functional programs and composition.
- Effects like global store and nondeterminism.
- Parallelism and concurrency.

Dimensions of a page:

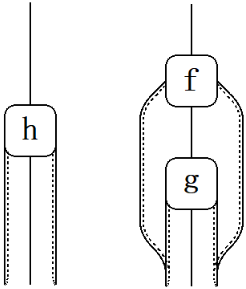
- Height.
- Width.
- ...

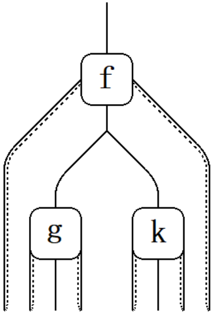
Effectful composition





How to add parallelism?





Motivation

Possible solutions:

- Surface diagrams, sheet diagrams.
- Ribbon diagrams, tape diagrams.
- String diagrams with special operations.

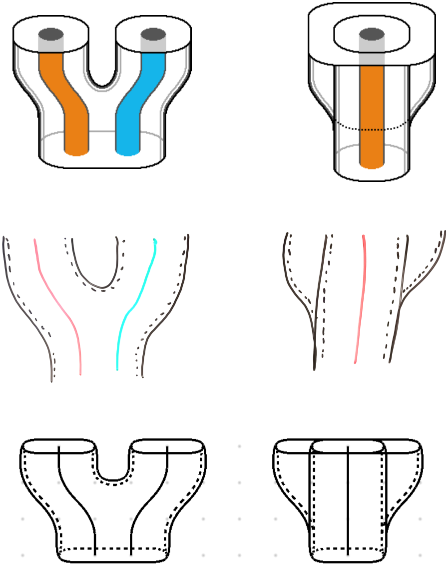
Solution of this talk: Cylinder diagrams
Inflate the strings and allow them to contain diagrams.

- Allows for arbitrary nesting.
- Can be neatly flattened to 2 dimensions.

Methods for drawing:

- This presentation will use drawing assembled using tiles with premade illustrations.
- Can also be illustrated by 2-dimensional string diagrams using *bracket-strings*.

Illustration: Parallel and sequential composition



Core Categorical structure

Given a monoidal category $\mathcal{C}$,
we describe the category consisting of:

- As objects multi-endofunctors on $\mathcal{C}$.
- As morphisms natural transformations between them.

The resulting category is also monoidal.

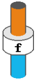
It contains $\mathcal{C}$ as a full subcategory of constant functors.

Main example is taking the category of sets $\mathcal{C} = \mathbf{Set}$,
with the Cartesian product as monoidal product.

Objects and morphisms of $\mathcal{C}$

Objects are represented (solid) cylinders,
and functions by blocks between them:

Object: $A \quad =$ 

Morphism: $\left(f : \dfrac{A}{B}\right) \quad =$ 

Special morphisms which exist in **Set** ($\times$ is the product):

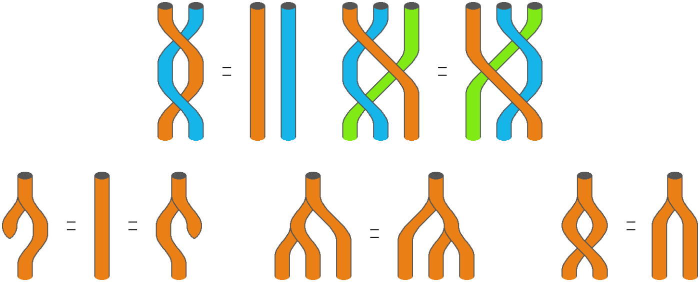
$\left(S : \dfrac{X \times Y}{Y \times X}\right) \quad =$ 

$C : \dfrac{X}{X \times X} \quad =$ 

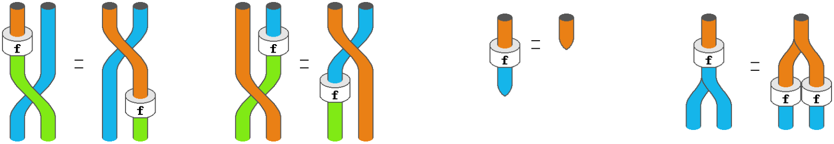
$D : \dfrac{X}{1} \quad =$ 

Objects and morphisms of $\mathcal{C}$

Special equations:

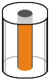


Naturality:

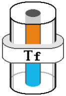


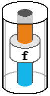
Endofunctors of $\mathcal{C}$

Functors are represented by hollow cylinders, within which lies the object or morphism to which it is applied.

$A =$  $\mapsto$ $TA =$ 

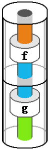
$\left(f : \frac{A}{B}\right) =$ 

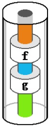
$\mapsto$ $\left(Tf : \frac{TA}{TB}\right) =$ 

$=$ 

Functor law for morphism composition:

$$\left(f : \frac{X}{Y}, g : \frac{Y}{Z}\right) \mapsto \left(Tg \circ Tf = T(g \circ f) : \frac{TX}{TZ}\right)$$

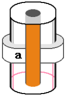


$=$


Natural Transformations of $\mathcal{C}$

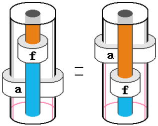
Natural transformations, a parametrized morphisms:

$$\left(\forall X.\ a_X : \frac{TX}{RX}\right) =$$



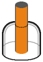
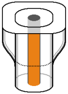
The object X goes through the natural transformation block, illustrating naturality:

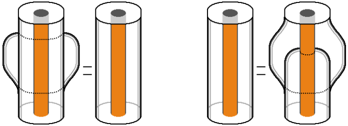
$$\left(\forall f : \frac{X}{Y}.\ a_Y \circ Tf = Rf \circ a_X\right)$$

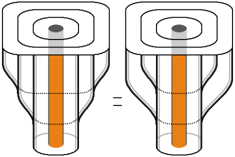


Let there be monads

Definition

A *monad* is a functor T with two natural transformations $\left(\eta_X : \frac{X}{TX}\right) =$  and $\left(\mu_X : \frac{TTX}{TX}\right) =$ , satisfying the following two equations:

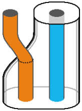
(Left and right-unit) 

(Associativity) 

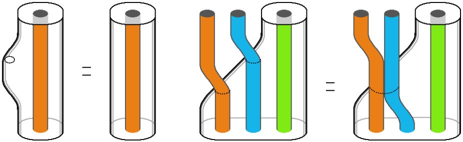
Strength

Strength is illustrated by pulling in the parameter:

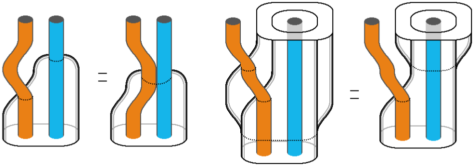
$$\left(\sigma_{X,Y} : \frac{X \times TY}{T(X \times Y)}\right) =$$



Monoidal strength equations:

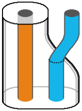


Strong monad equations:



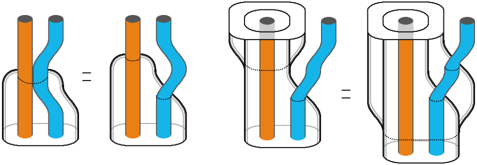
Costrength

Using swap, we define costrength as

$$\left(\tau_{X,Y} : \frac{TX \times Y}{T(X \times Y)}\right) =$$  $:=$ .

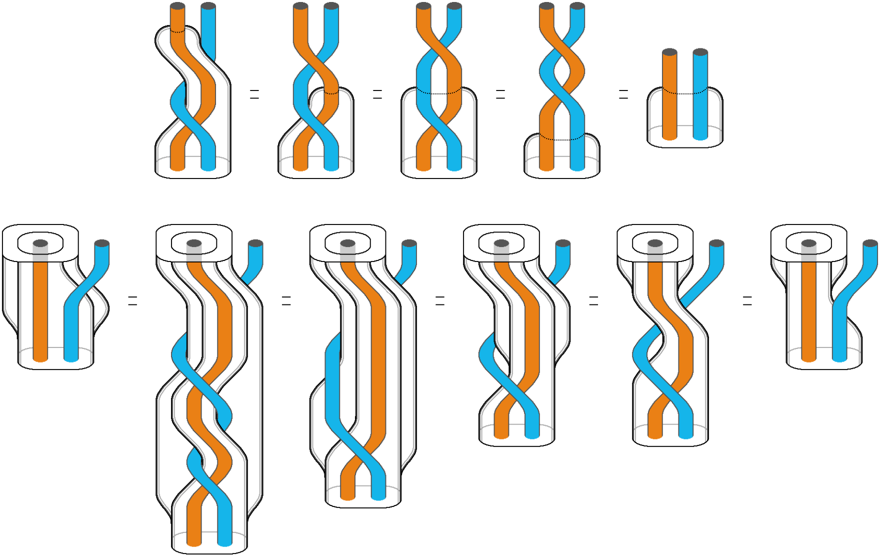
Lemma

A strong monad is a costrong monad.



Costrength

Proof.



□

Commutativity

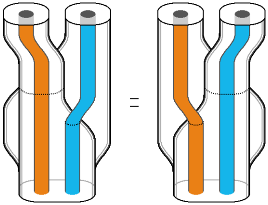
How do we *merge* monads?

One method is to use strength and costrength.

Commutativity says the order doesn't matter.

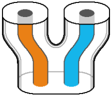
Definition

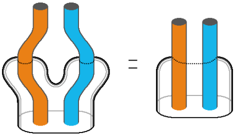
A monad is commutative if the following equation holds:

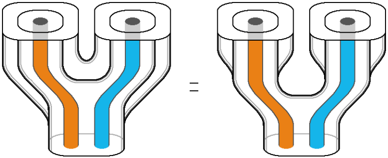


Monoidal Monad

Definition

T is a *monoidal monad* if there is a natural transformation $\left(m : \frac{TX \times TY}{T(X \times Y)}\right) =$  satisfying the following two equations:

(η) 

(μ) 

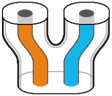
Lemma

Any strong commutative monad is a monoidal monad.

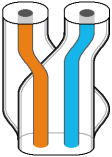
Monoidal Monad

Proof.
We define the following natural transformation:

$$\left(m : \frac{TX \times TY}{T(X \times Y)}\right) =$$

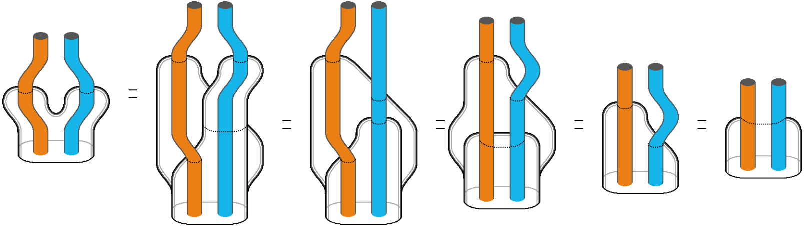


$:=$

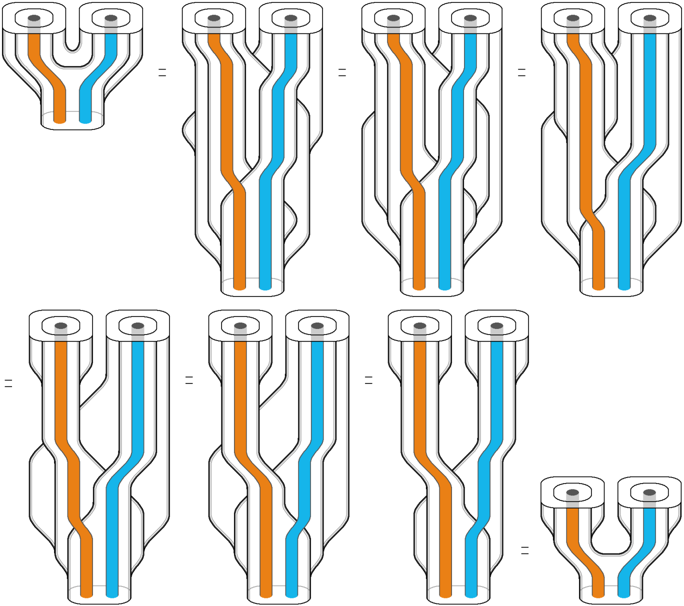


.

Using commutativity, we can prove both monoidal monad properties:

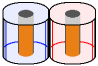


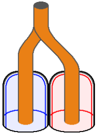
Monoidal Monad



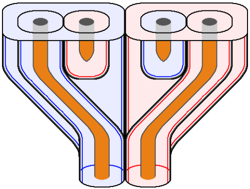
Product of Monads

T and R both monads.
Assume we have copy and delete.

$$(T \times R)(X) = (TX \times RX) =$$ 

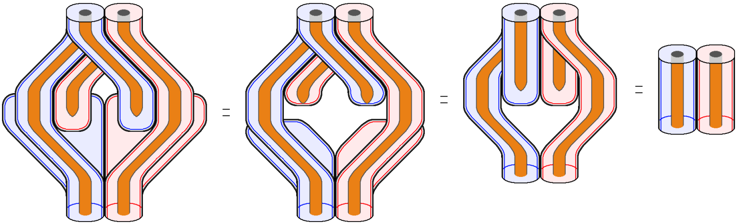
- The monad unit $\left(\eta_X^{T\times R} : \frac{X}{TX \times RX}\right) =$  .

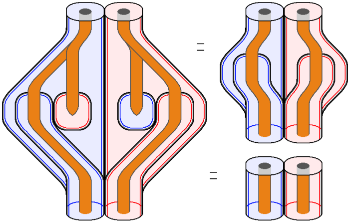
- The monad multiplication

$$\left(\mu_X^{T\times R} : \frac{T(TX\times RX)\times R(TX\times RX)}{TX\times RX}\right) =$$ 

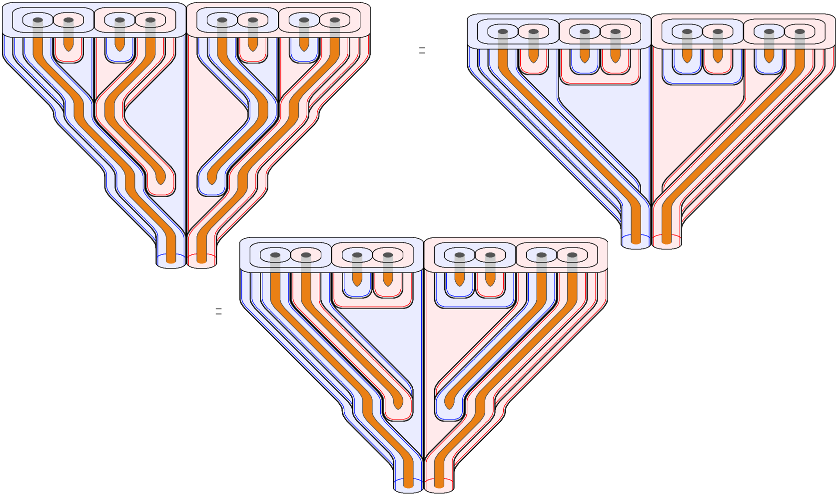
Product of Monads

This is a monad. Proof:

▶ 

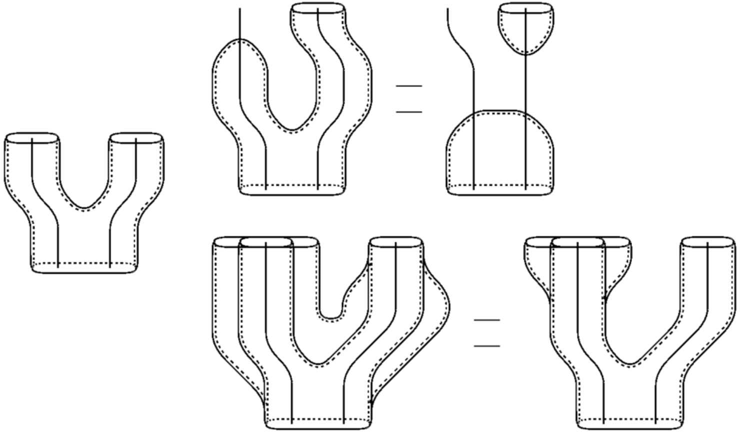
▶ 

Product of Monads

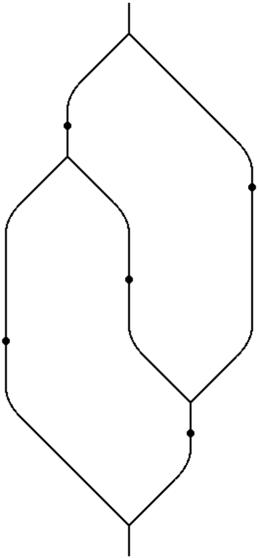


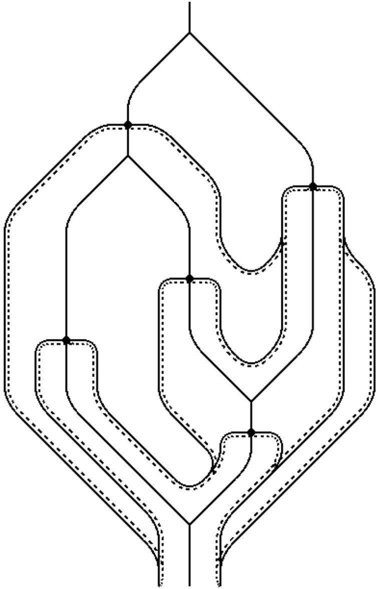
Personal motivation: Comomonad interactions

- Models with both leading/active monadic structure, and following/passive comonad structure.
- Interaction between programs as lead and follow, using interaction laws.



Personal motivation: Comomonad interactions





Multi-functors generically

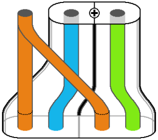
Functors like $X \times TY$ have multiple arguments.

Generically, this can be illustrated by glueing cylinders.

Natural transformations can be illustrated by connecting the appropriate arguments.

For example, coproduct in **Set** can be represented by glued cylinders. Below is an illustration of distributivity.

$$\left(\delta_{X,Y,Z} : \frac{X \times (Y + Z)}{(X \times Y) + (X \times Z)}\right) =$$



Conclusions: Open questions

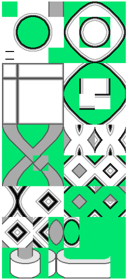
How to do cylinder diagrams monoidally?

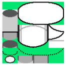
Geometric interpretation of cylinder diagrams.

Considering other generic features.

Conclusions: Pictures

Most images were composed by combining tiles in layers, using the following tilesets:





Working on more automatized methods too.

String drawing method on grid: Link