TOWARDS
PATTERN - MATCHING
~~BINDING~~ & SUBSTITUTION
in
STRING DIAGRAMS

ROSS DUNCAN — UNIVERSITY OF STRATHCLYDE

Pattern matching, binding and substitution

```
f [] = []
f (x :: xs) = (x+1) :: (f xs)
```

Pattern matching, binding and substitution

```
f [] = []
f (x :: xs) = (x+1) :: (f xs)
```

```
f (1 :: (2 :: 3 :: []))
```

Pattern matching, binding and substitution

```
f [] = []
f (x :: xs) = (x+1) :: (f xs)
```

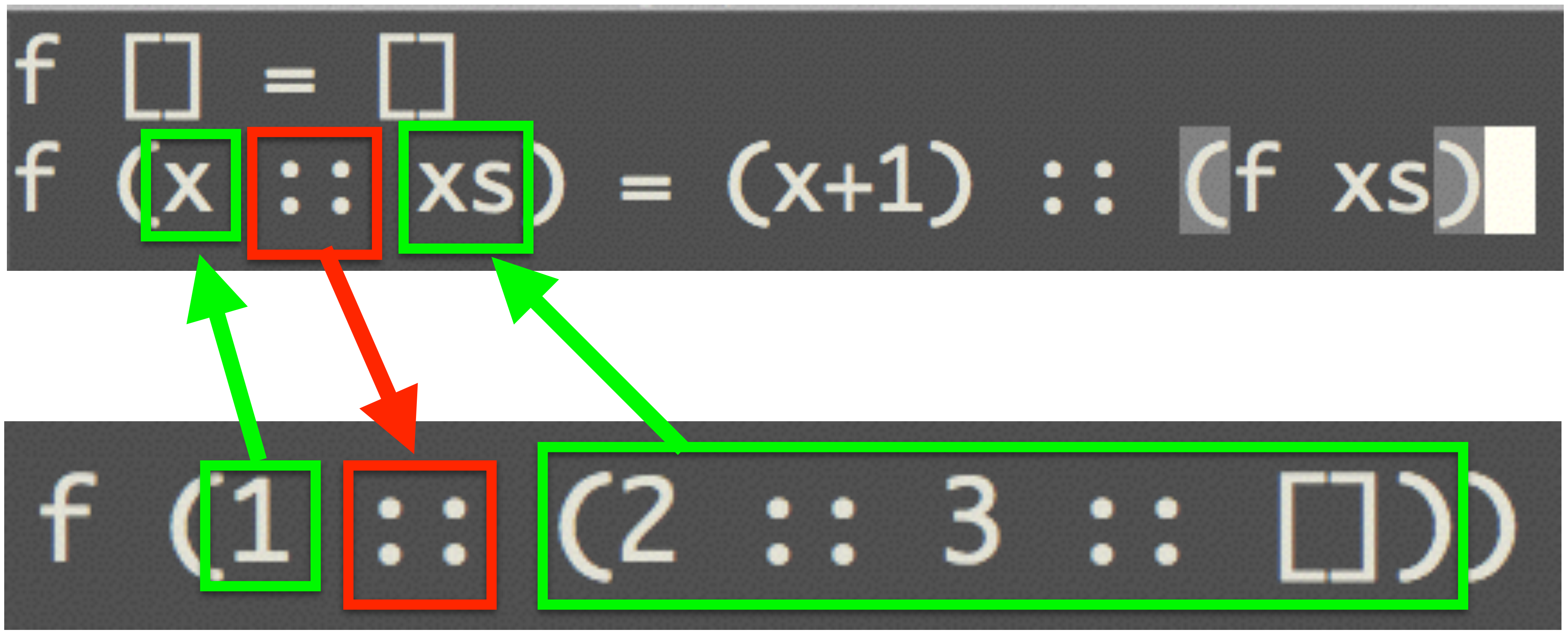
```
f (1 :: (2 :: 3 :: []))
```

Pattern matching, binding and substitution

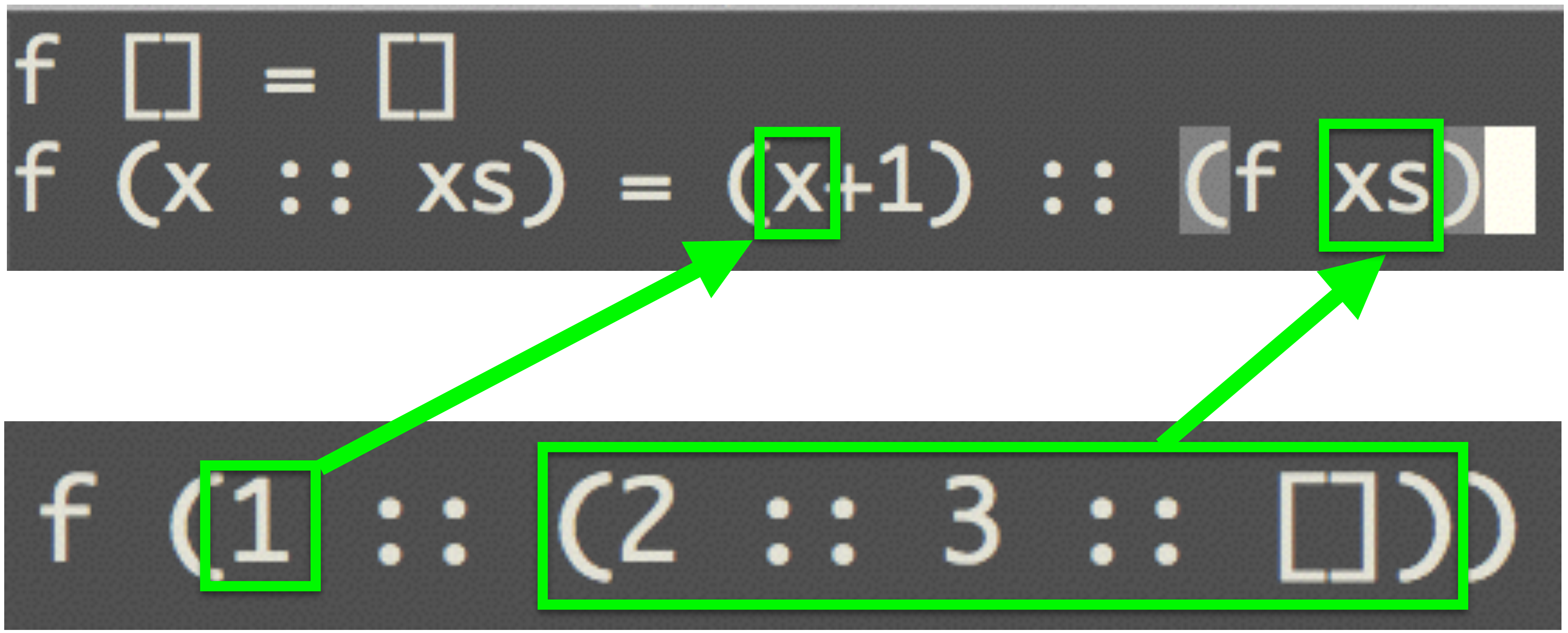
```
f [] = []
f (x :: xs) = (x+1) :: (f xs)
```

```
f (1 :: (2 :: 3 :: []))
```

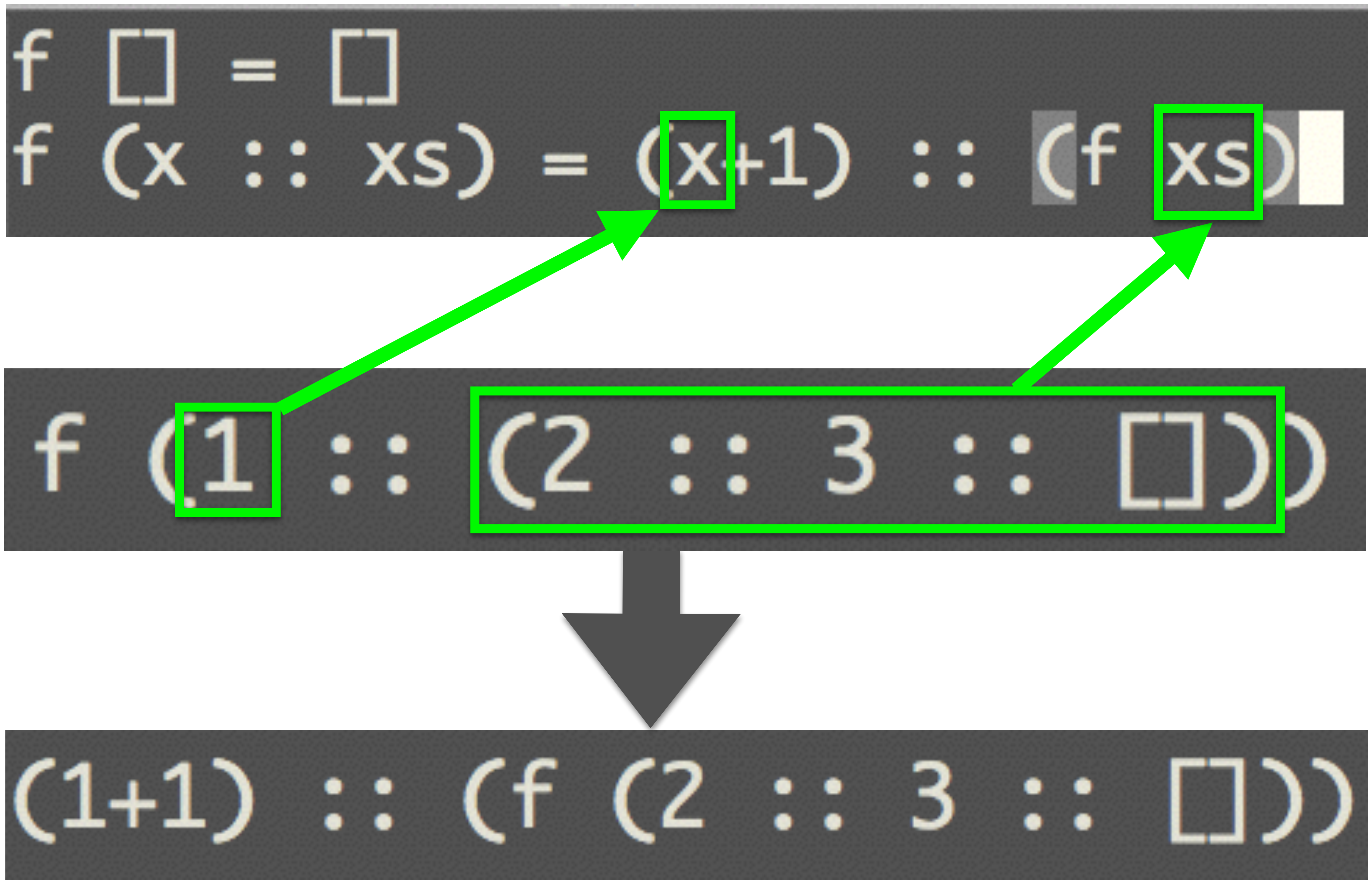
Pattern matching, binding and substitution



Pattern matching, binding and substitution



Pattern matching, binding and substitution



1.

Monoidal Categories

1. Symmetric Monoidal Categories

A monoidal category M is a category with a bifunctor, $\otimes$ or $\square$,

$$\square : M \times M \to M$$

written for objects a, b of M variously as a "product"

$$(a, b) \to a \square b, a \otimes b, \text{or } a\,b$$

which is associative up to a natural isomorphism

$$\alpha : a(b\,c) \cong (a\,b)c \tag{1}$$

and is equipped with an element e, which is unit up to natural isomorphisms

$$\lambda : e\,a \cong a\,, \qquad \rho : a\,e \cong e\,. \tag{2}$$

These maps must satisfy certain commutativity requirements; for α, a pentagonal diagram

$$\begin{tikzcd}
a(b(c\,d)) \arrow[r, "\alpha"] \arrow[d, "1\alpha"'] & (a\,b)(c\,d) \arrow[r, "\alpha"] & ((a\,b)c)d \arrow[d, "\alpha 1"] \\
a((b\,c)d) \arrow[rr, "\alpha"] & & (a(b\,c))d\,,
\end{tikzcd} \tag{3}$$

as in § VII.1.(5), and for λ and ρ the two commutativities

$$\begin{tikzcd}
a(e\,c) \arrow[r, "\alpha"] \arrow[d, "1\lambda"'] & (a\,e)c \arrow[d, "\rho 1"] \\
a\,c \arrow[r, phantom, "="] & a\,c\,,
\end{tikzcd} \qquad \lambda = \rho : e\,e \to e\,. \tag{4}$$

A *braiding* for a monoidal category M consists of a family of isomorphisms

$$\gamma_{a,b} : a \square b \cong b \square a \tag{5}$$

natural in a and $b \in M$, which satisfy for e the commutativity

$$\begin{tikzcd}
a \square e \arrow[r, "\gamma"] \arrow[d, "\rho"'] & e \square a \arrow[d, "\lambda"] \\
a \arrow[r, phantom, "="] & a
\end{tikzcd} \tag{6}$$

and which, with the associativity α, make both the following hexagonal diagrams commute (with the symbol $\square$ omitted):

$$\begin{tikzcd}
(a\,b)c \arrow[r, "\gamma"] \arrow[d, "\alpha^{-1}"] & c(a\,b) \arrow[d, "\alpha"] & a(b\,c) \arrow[r, "\gamma"] \arrow[d, "\alpha"] & (b\,c)a \arrow[d, "\alpha^{-1}"] \\
a(b\,c) \arrow[d, "1\cdot\gamma"'] & (c\,a)b \arrow[d, "\gamma\cdot 1"] & (a\,b)c \arrow[d, "\gamma\cdot 1"] & b(c\,a) \arrow[d, "1\cdot\gamma"] \\
a(c\,b) \arrow[r, "\alpha"] & (a\,c)b\,, & (b\,a)c \arrow[r, "\alpha^{-1}"] & b(a\,c)\,.
\end{tikzcd} \tag{7}$$

Note that the first diagram replaces each $\gamma_{ab,c}$ which has a product $a\,b$ as first index by two γ's with single indices, while the second hexagonal diagram does the same for $\gamma_{a,bc}$ with a product as second index. Note also that the first hexagon of (7) for γ implies the second diagram for γ^{-1}, and conversely. Thus, when γ is a braiding for M, then γ^{-1} is also a braiding for M.

A *symmetric monoidal category*, as already defined in § VII. 7, is a category with a braiding γ such that every diagram

$$\begin{tikzcd}
a\,b \arrow[r, "\gamma_{a,b}"] \arrow[dr] & b\,a \arrow[d, "\gamma_{b,a}"] \\
& a\,b
\end{tikzcd} \tag{8}$$

commutes. For this case, either one of the hexagons (7) implies the other.

1bis.

Monoidal Categories (Graphically)

Why Diagrams?

Why Diagrams?

$$\frac{1}{2}\left(\begin{pmatrix}1\\1\end{pmatrix}\otimes\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&e^{i\gamma}\end{pmatrix}\otimes\begin{pmatrix}0&1\\1&0\end{pmatrix}\right)$$

$$\circ\Bigg(\Bigg(\bigg(\begin{pmatrix}1&0&0&0\\0&0&0&1\end{pmatrix}\otimes\begin{pmatrix}1&1\\1&-1\end{pmatrix}\bigg)\bigg)\circ\Bigg(\begin{pmatrix}0&1\\1&0\end{pmatrix}\otimes\begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\Bigg)\Bigg)\circ$$

$$\Bigg(\Bigg(\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\otimes\begin{pmatrix}1&0\\0&1\end{pmatrix}\Bigg)\Bigg)\Bigg)\otimes\begin{pmatrix}1&0\\0&1\end{pmatrix}\Bigg)$$

$$\circ\Bigg(\Bigg(\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{pmatrix}\circ\bigg(\bigg(\begin{pmatrix}\cos\frac{\pi}{6}\\i\sin\frac{\pi}{6}\end{pmatrix}\otimes\begin{pmatrix}1&0\\0&e^{i\beta}\end{pmatrix}\bigg)\bigg)\Bigg)\otimes\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&e^{i\alpha}\end{pmatrix}\Bigg)$$

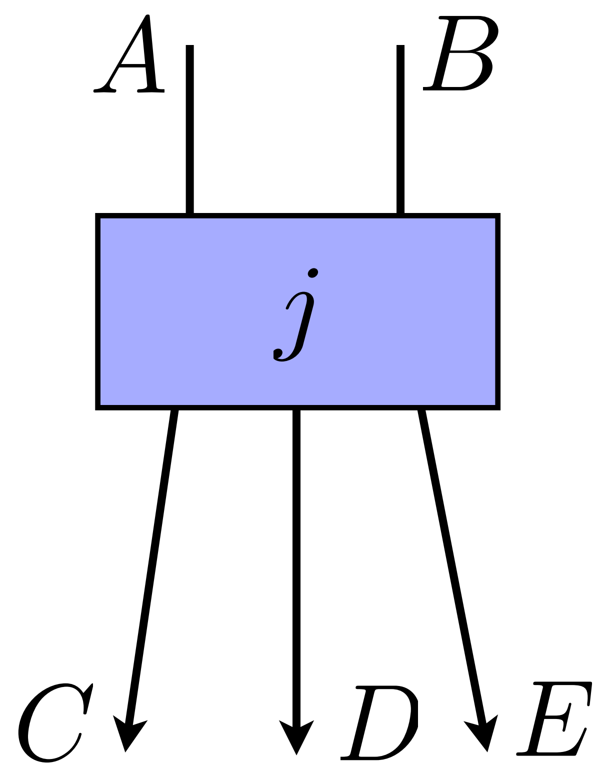
$$= ?$$

Why Diagrams?

- Great when we have parallel and sequential composition
- Essential for talking about interacting algebraic and coalgebraic things
- Different kinds of diagram give different kinds of monoidal category

Diagrams

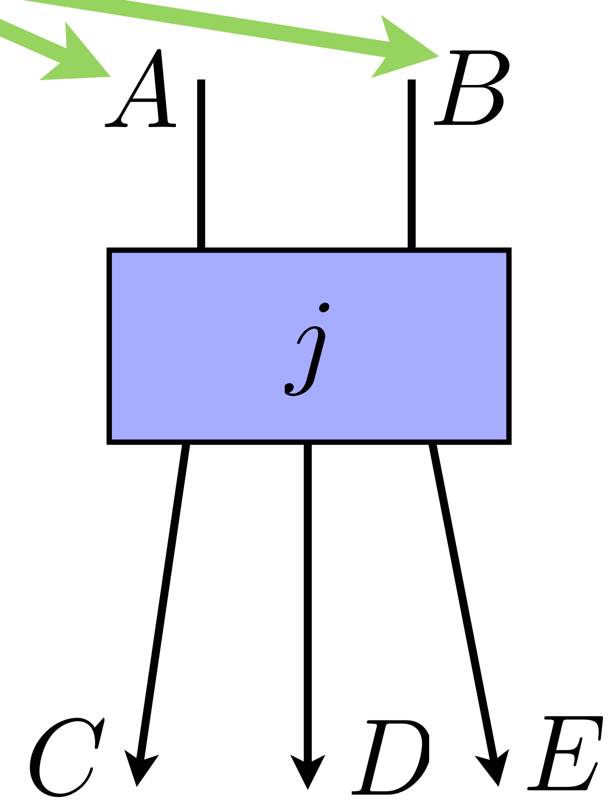
$$j : A \otimes B \to C \otimes D \otimes E$$



Diagrams

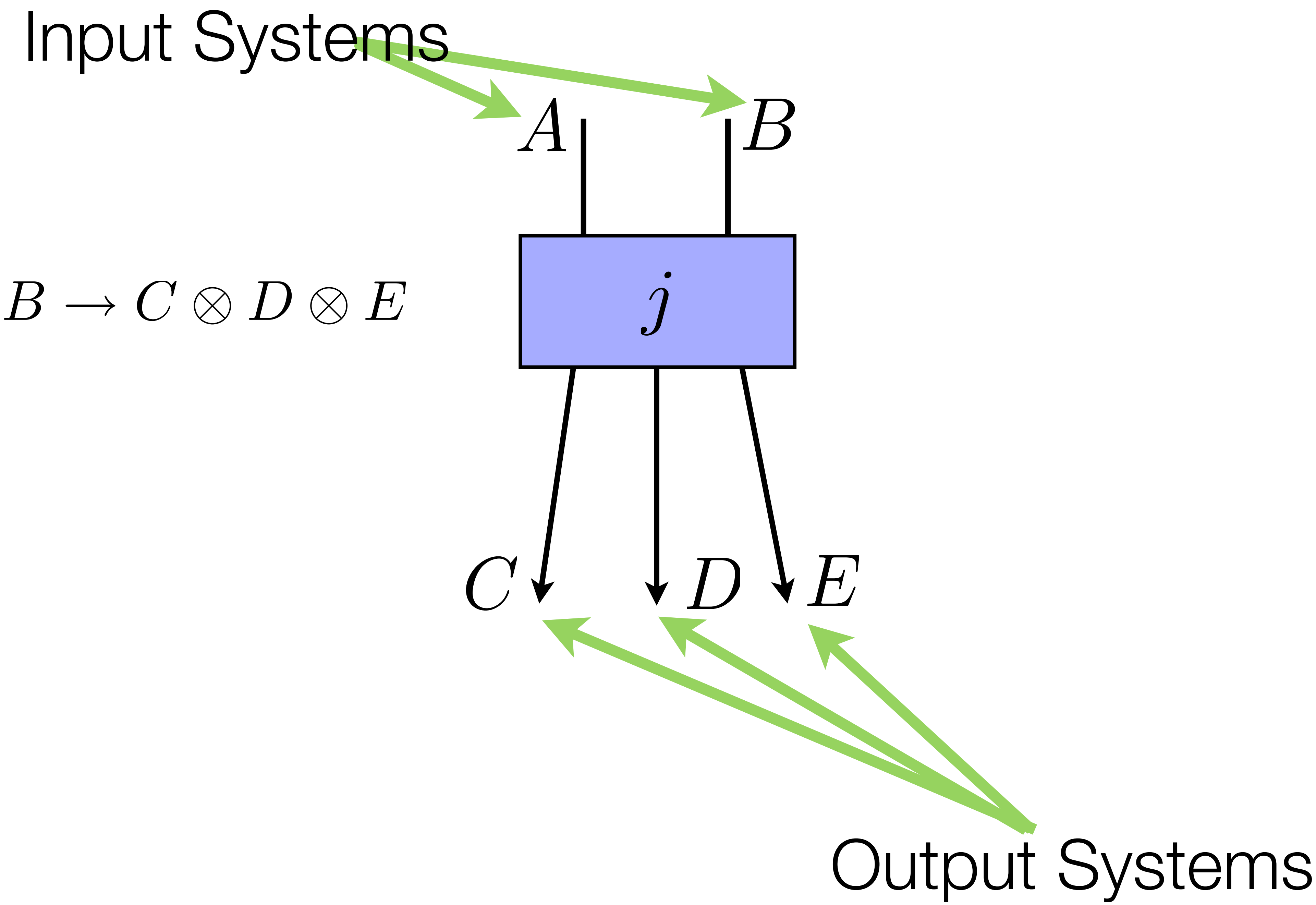
Input Systems

$$j : A \otimes B \to C \otimes D \otimes E$$



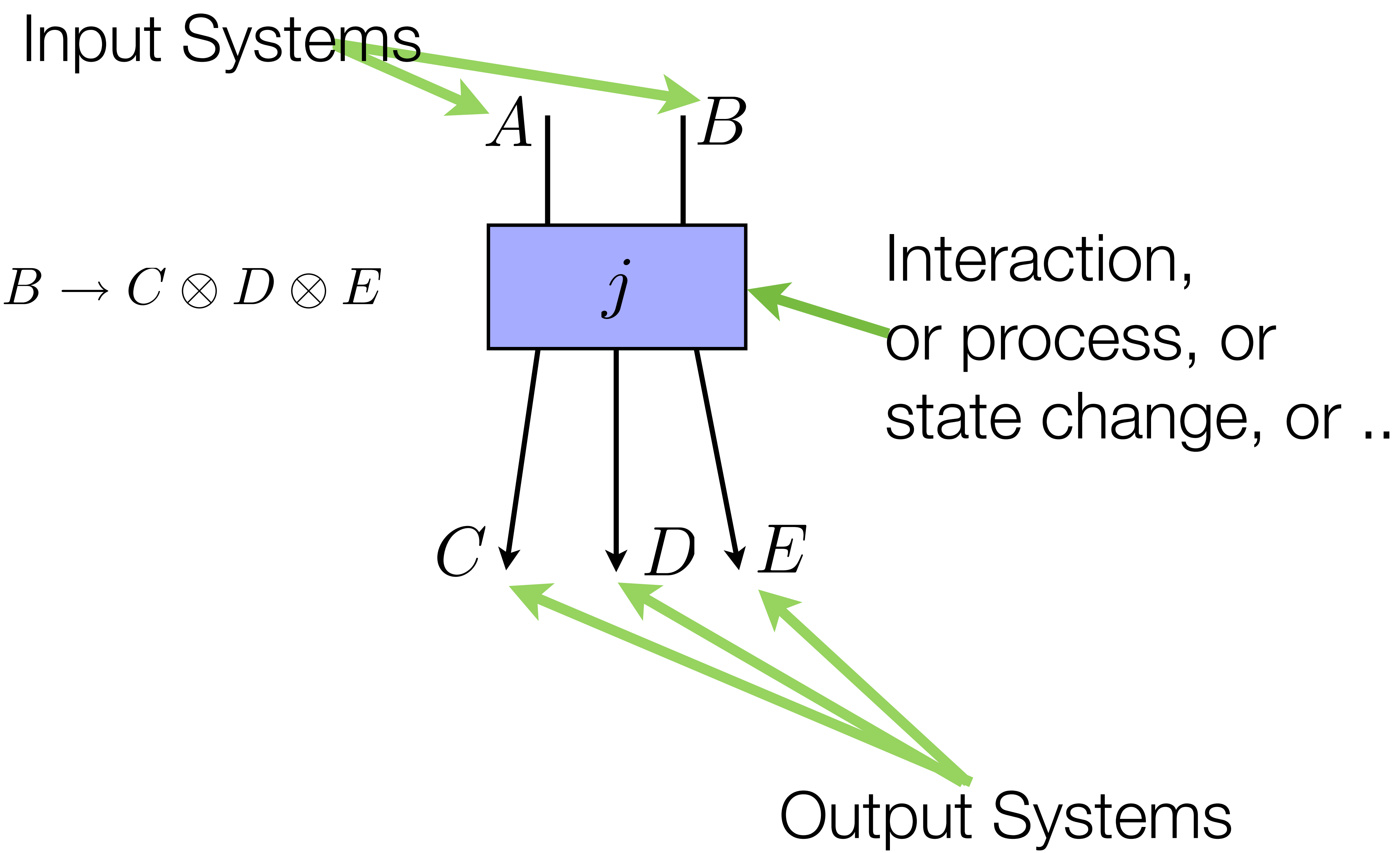
Diagrams

$$j : A \otimes B \to C \otimes D \otimes E$$



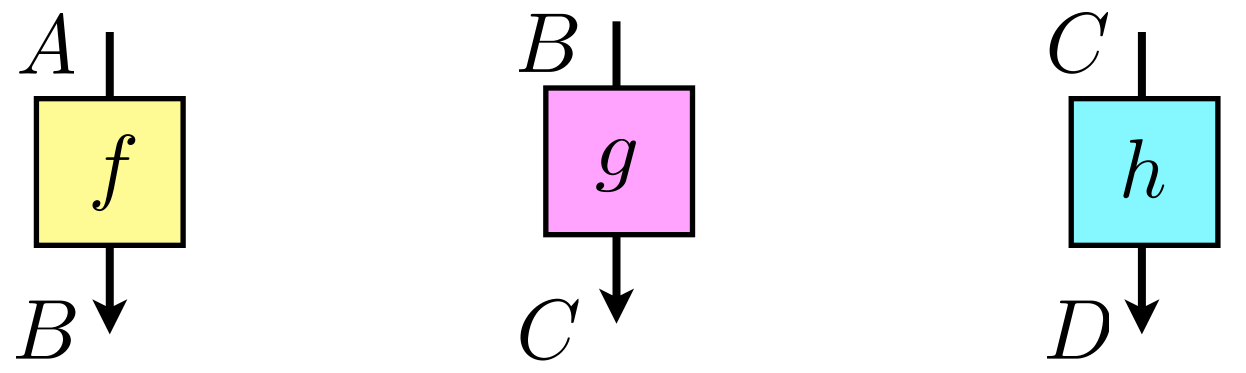
Diagrams

$$j : A \otimes B \to C \otimes D \otimes E$$



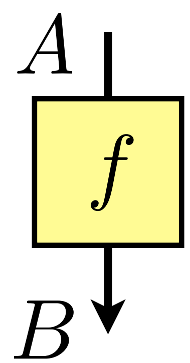
Monoidal Categories

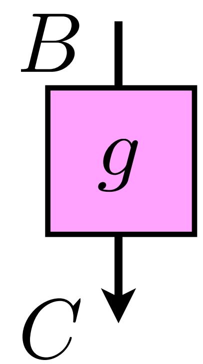
$f : A \to B$ $\quad$ $g : B \to C$ $\quad$ $h : C \to D$

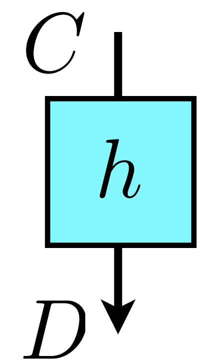


Monoidal Categories

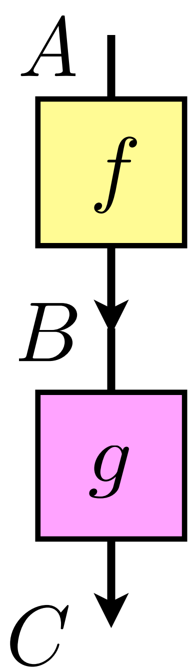
$f : A \to B$ $\quad$ $g : B \to C$ $\quad$ $h : C \to D$





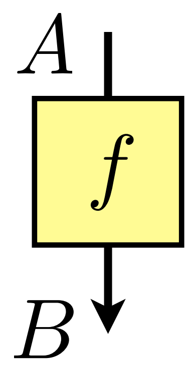


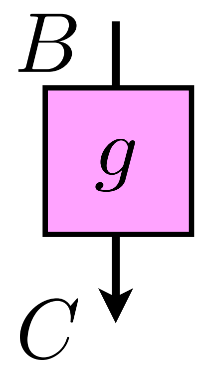
$g \circ f : A \to C$

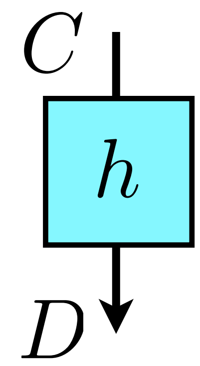


Monoidal Categories

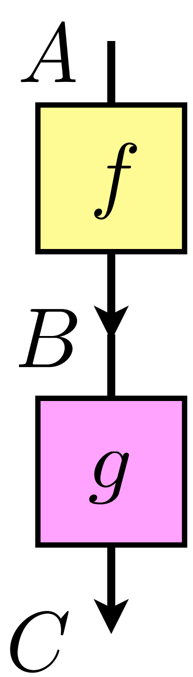
$f : A \longrightarrow B$ $\qquad$ $g : B \longrightarrow C$ $\qquad$ $h : C \longrightarrow D$



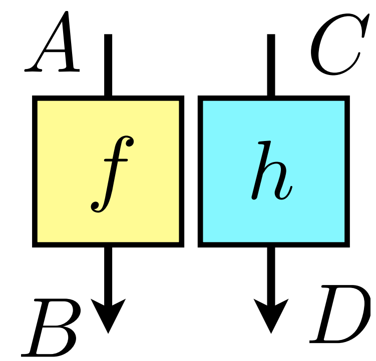




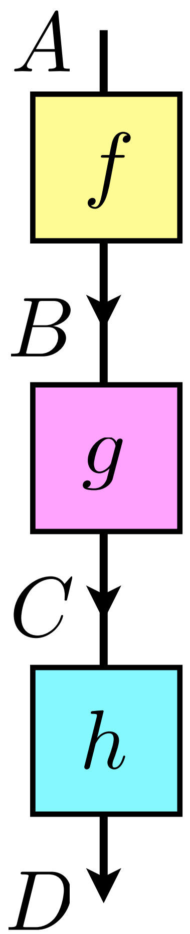
$g \circ f : A \longrightarrow C$

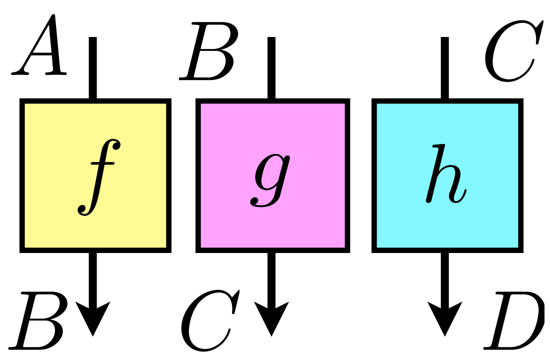


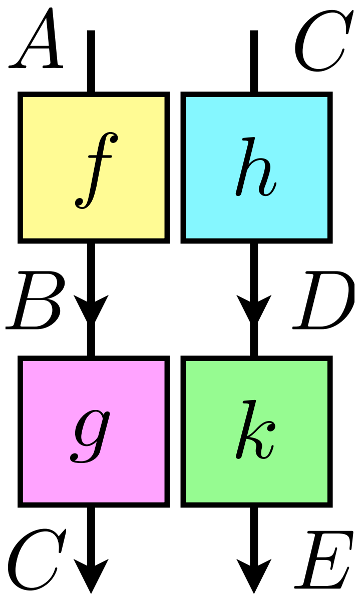
$f \otimes h : A \otimes C \longrightarrow B \otimes D$



Monoidal Categories







Monoidal Categories

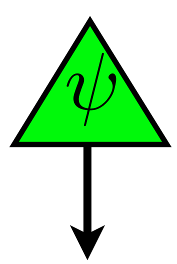
Monoidal categories have a special *unit* object called I which is a left and right identity for the tensor:

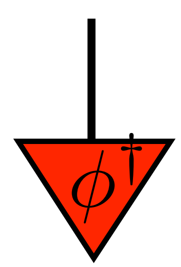
$$I \otimes A = A = A \otimes I$$

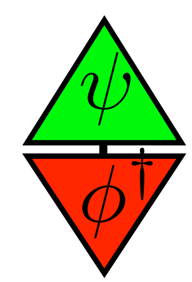
$$\mathrm{id}_I \otimes f = f = f \otimes \mathrm{id}_I$$

No lines are drawn for I in the graphical notation:

$$\psi : I \longrightarrow A \qquad \phi^\dagger : A \longrightarrow I \qquad \phi^\dagger \circ \psi : I \longrightarrow I$$

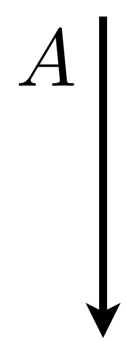






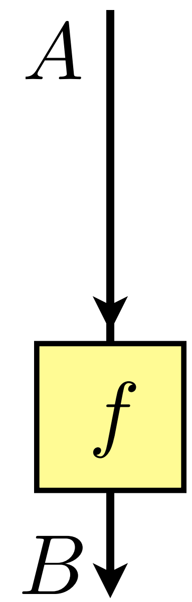
Categories

$$\mathrm{id}_A : A \to A$$



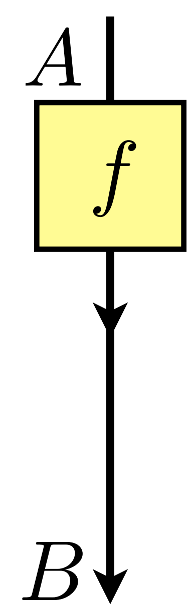
Categories

$$f \circ \mathrm{id}_A : A \to B$$



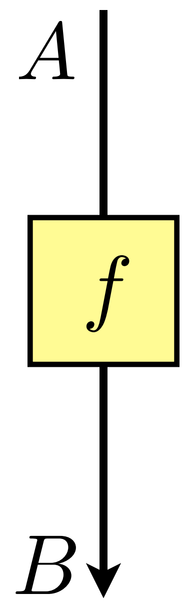
Categories

$$\mathrm{id}_B \circ f : A \to B$$



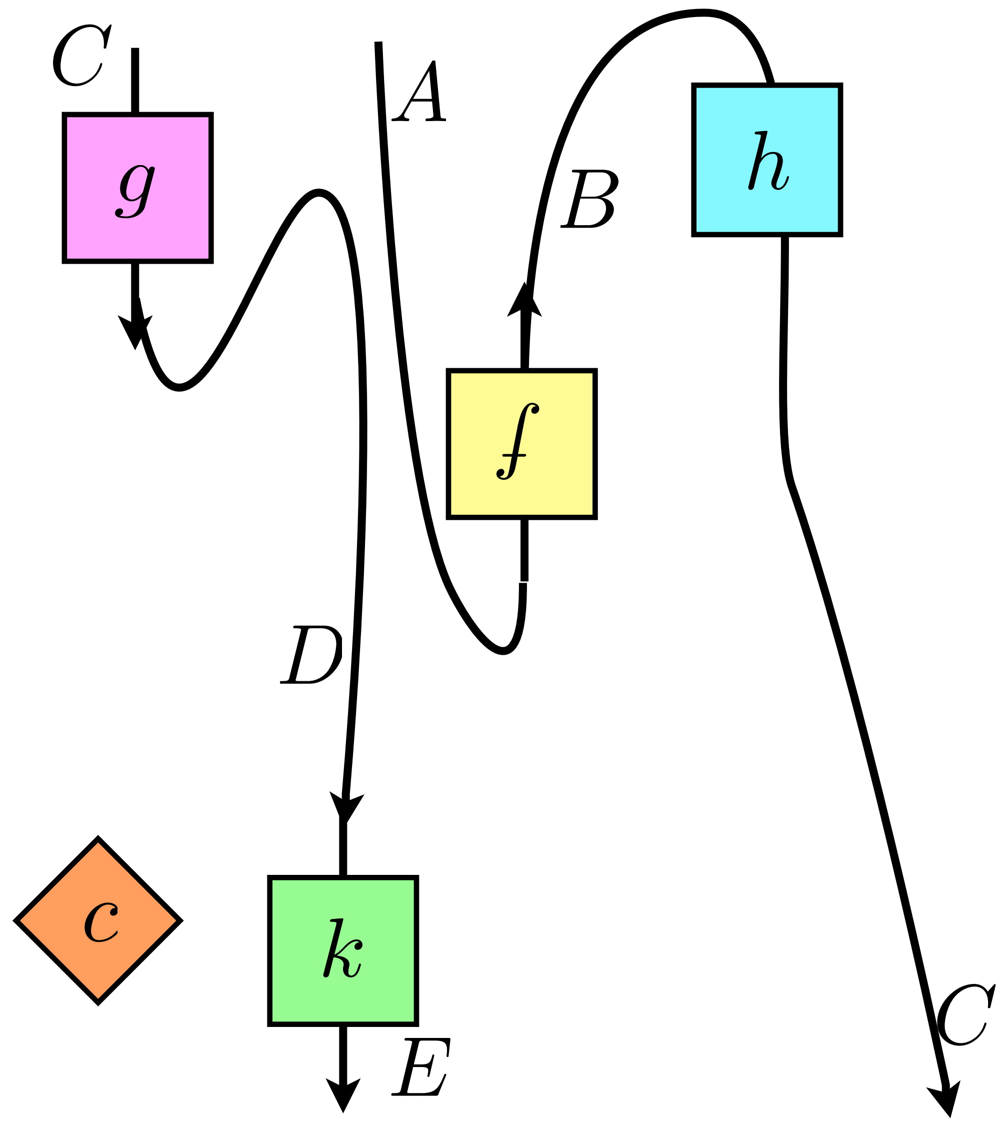
Categories

$$f : A \to B$$



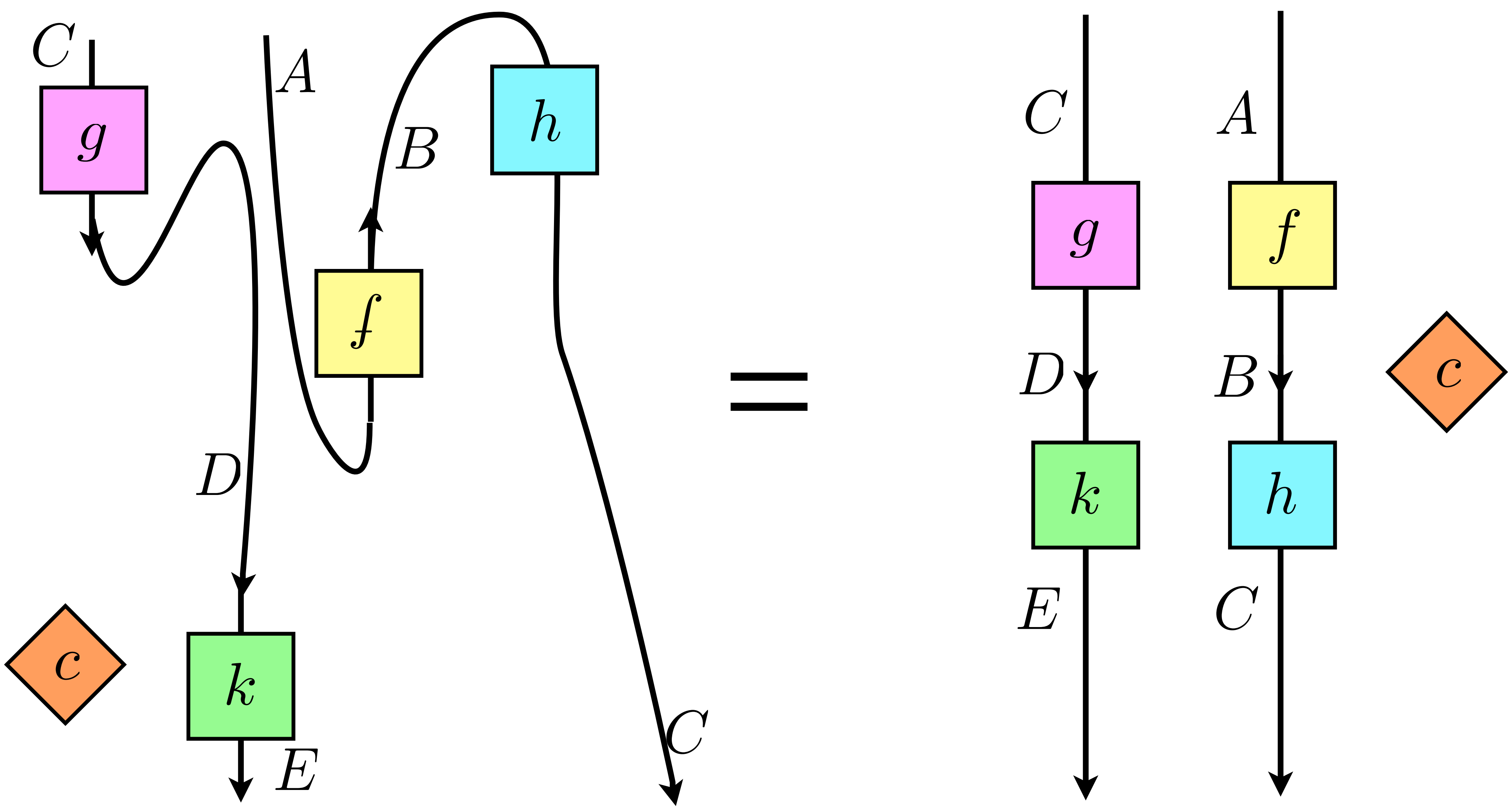
Graphical Calculus Theorem

Thm: one diagram can be deformed to another iff their denotations are equal by the structural equations of the category.



Graphical Calculus Theorem

<u>Thm</u>: one diagram can be deformed to another iff their denotations are equal by the structural equations of the category.

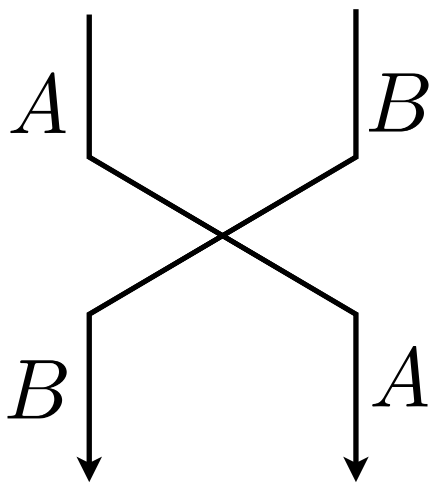


Crossing the streams

Are wires allowed to cross?

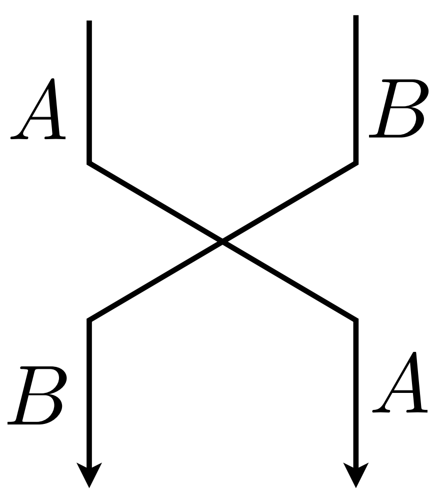
Crossing the streams

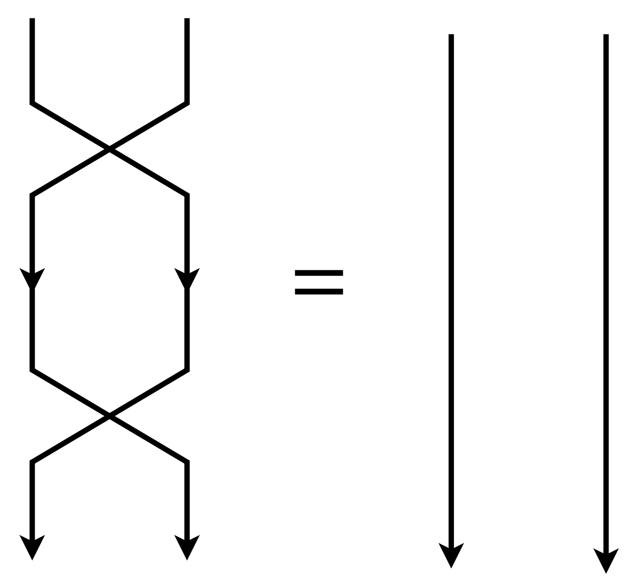
Are wires allowed to cross?



Crossing the streams

Are wires allowed to cross?

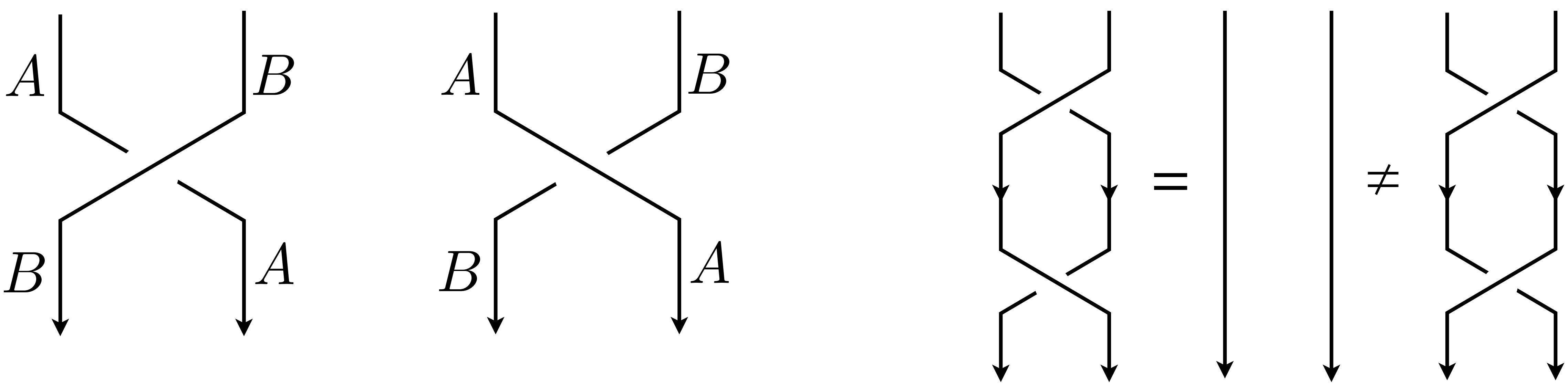




YES : symmetric monoidal — diagrams are DAGs

Crossing the streams

Are wires allowed to cross?



YES, BUT : braided monoidal

— diagrams are framed tangles

Crossing the streams

Are wires allowed to cross?

Crossing the streams

Are wires allowed to cross?

No : (planar) monoidal — diagrams are planar DAGs

Monoidal Theories

Syntactic presentation of a diagrammatic theory:

Generators
symbols with arity and coarity

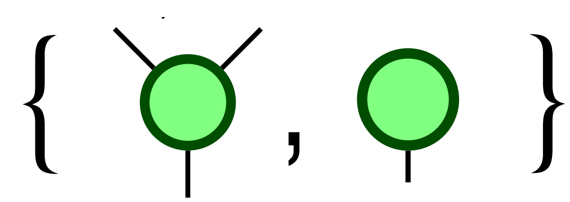
$$(\Sigma, E)$$

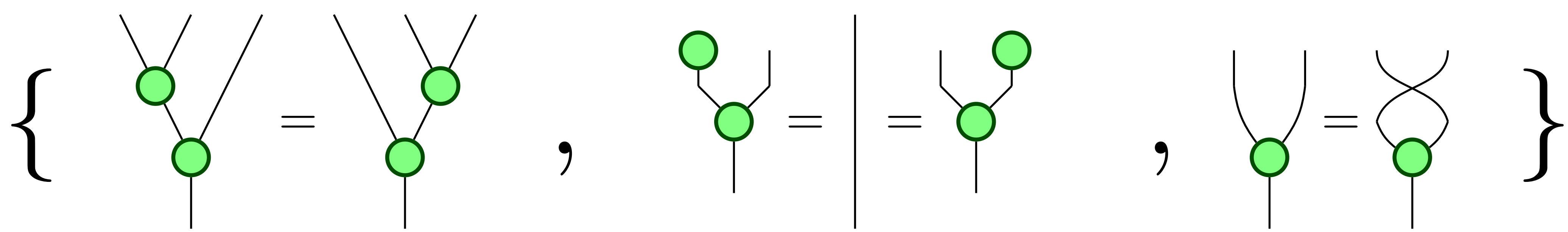
Relations
equations between terms of same type

NB : a *PRO(P)* is a (symmetric) monoidal category where the wires don't have types.

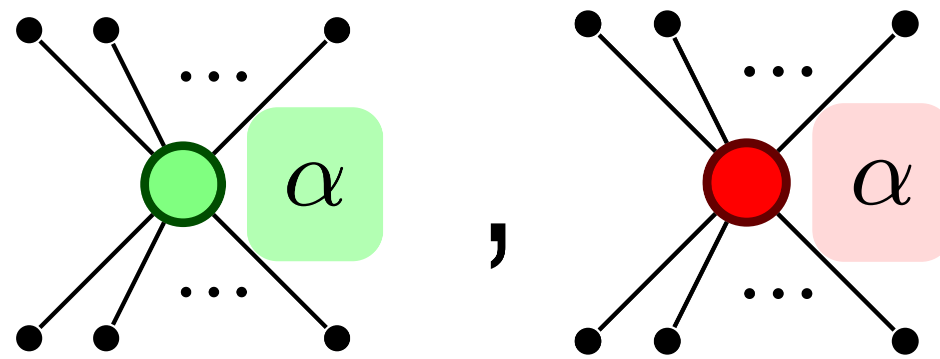
Example: commutative monoids

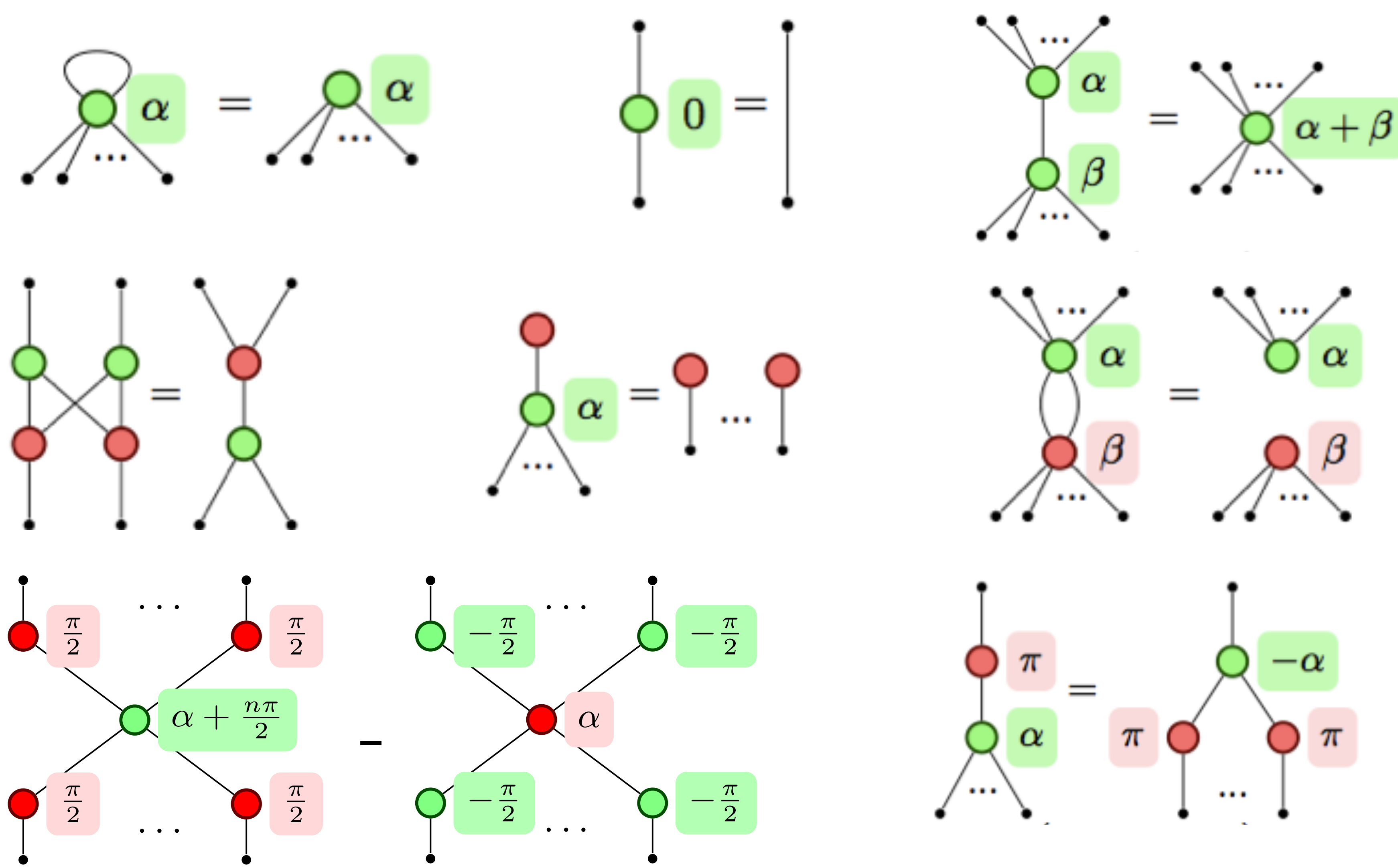
The PROP of commutative monoids $\mathbb{M}$

$\Sigma =$ 

$E =$ 

Example : the ZX-calculus

$\Sigma = \{$  $\}$

$E = \{$  $\}$

TOWARDS
PATTERN-MATCHING

~~BINDING~~ & SUBSTITUTION

in

STRING DIAGRAMS

Computing Science Group

Geometry of abstraction in quantum computation

Dusko Pavlovic
Oxford University and Kestrel Institute

CS-RR-09-13

Geometry of abstraction in quantum computation

Pavlovic (2009,2012)

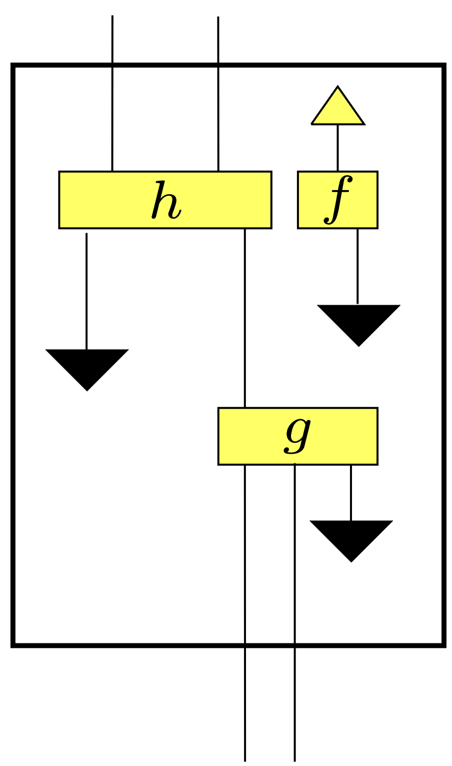
Quantum algorithms are sequences of abstract operations, performed on non-existent computers. They are in obvious need of categorical semantics.

Geometry of abstraction in quantum computation

Pavlovic (2009,2012)

monoidal category $\mathcal{C}$

polynomial monoidal category $\mathcal{C}[x : X]$

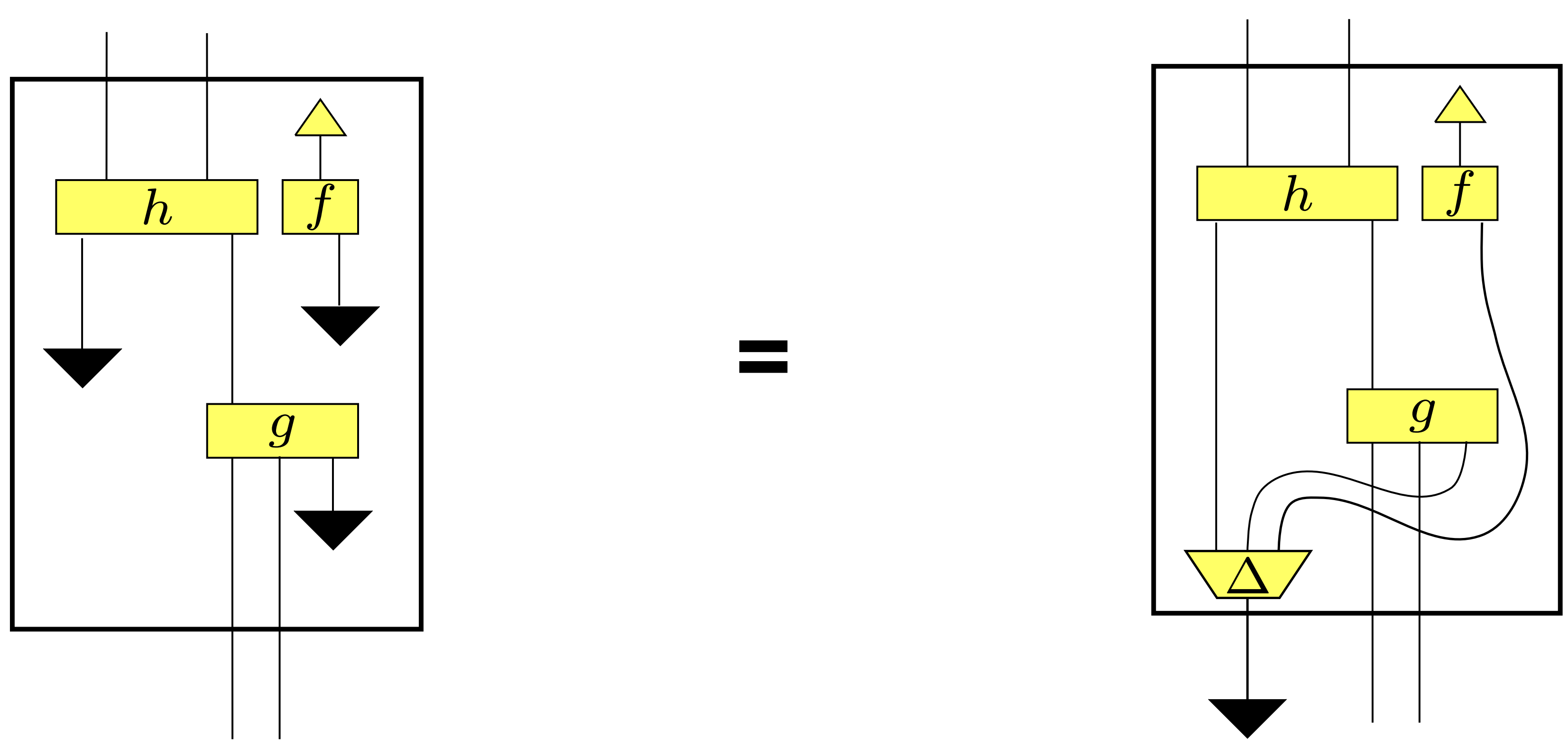


Geometry of abstraction in quantum computation

Pavlovic (2009,2012)

monoidal category $\mathcal{C}$

polynomial monoidal category $\mathcal{C}[x : X]$



Geometry of abstraction in quantum computation

Pavlovic (2009,2012)

monoidal category $\mathcal{C}$

polynomial monoidal category $\mathcal{C}[x : X]$

Theorem 3.4 *The category* $\mathrm{Abs}_{\mathrm{C}}$ *of monoidal abstractions is equivalent with the category* $\mathrm{C}_{\times}$ *of commutative comonoids in* C.

Geometry of abstraction in quantum computation

Pavlovic (2009,2012)

monoidal category $\mathcal{C}$

polynomial monoidal category $\mathcal{C}[x : X]$

Theorem 3.4 *The category* $\mathrm{Abs}_{\mathrm{C}}$ *of monoidal abstractions is equivalent with the category* $\mathrm{C}_{\times}$ *of commutative comonoids in* C.

Corollary 4.5 *The category of dagger-monoidal abstractions* $\ddagger$-$\mathrm{Abs}_{\mathrm{C}}$ *is equivalent with the category* C_{Δ} *of commutative dagger-Frobenius algebras and comonoid homomorphisms in* C

TOWAR

PATTERN - MATCHING

~~BINDING~~ & SUBSTITU

in

STRING DIAGRAM

1. OPERADs

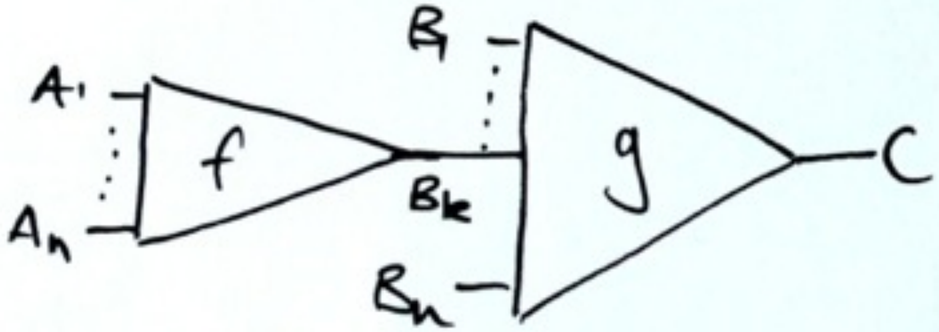
$$A \xrightarrow{f} B \xrightarrow{g} C$$

(arrows in a category)

1. OPERADs

$$A \xrightarrow{f} B \xrightarrow{g} C$$

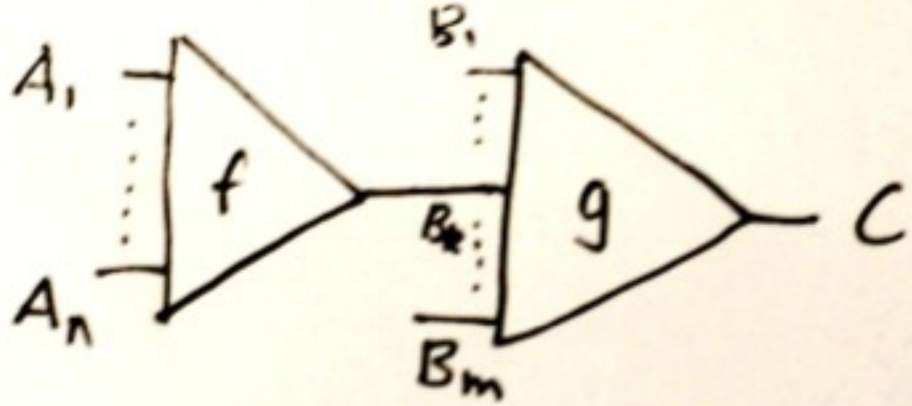
(arrows in a category)



(arrows in an operad)

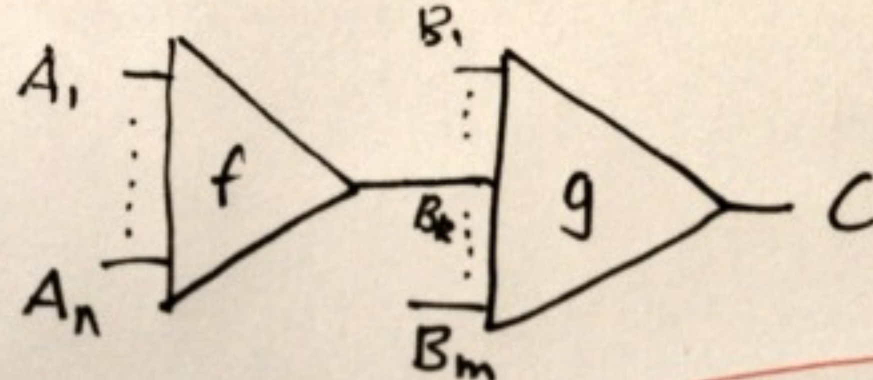
aka. multicategory.

1. OPERADS



$$\frac{x_1 : A_1, \ldots, x_n : A_n \vdash f : B_k \qquad y_1 : B_1, \ldots, y_k : B_k, \ldots, y_m : B_n \vdash g : C}{y_1 : B_1, \ldots, x_1 : A_1, \ldots, x_n : A_n, \ldots, y_m : B_m \vdash g[f/x] : C}\ \text{CUT}$$

1. OPERADS



$$\frac{x_1 : A_1, \dots, x_n : A_n \vdash f : B_k \qquad y_1 : B_1, \dots, \cancel{y_k : B_k}, \dots, y_m : B_n \vdash g : C}{y_1 : B_1, \dots, \boxed{x_1 : A_1, \dots, x_n : A_n}, \dots, y_m : B_m \vdash g[f/y_k] : C}\ \text{CUT}$$

2. Making an Operad from a PRO

- Let (Σ, E) be a presentation of a PRO.
- Adjoin "enough" new generators $x : m \to n$ for every $m, n \in \mathbb{N}$.

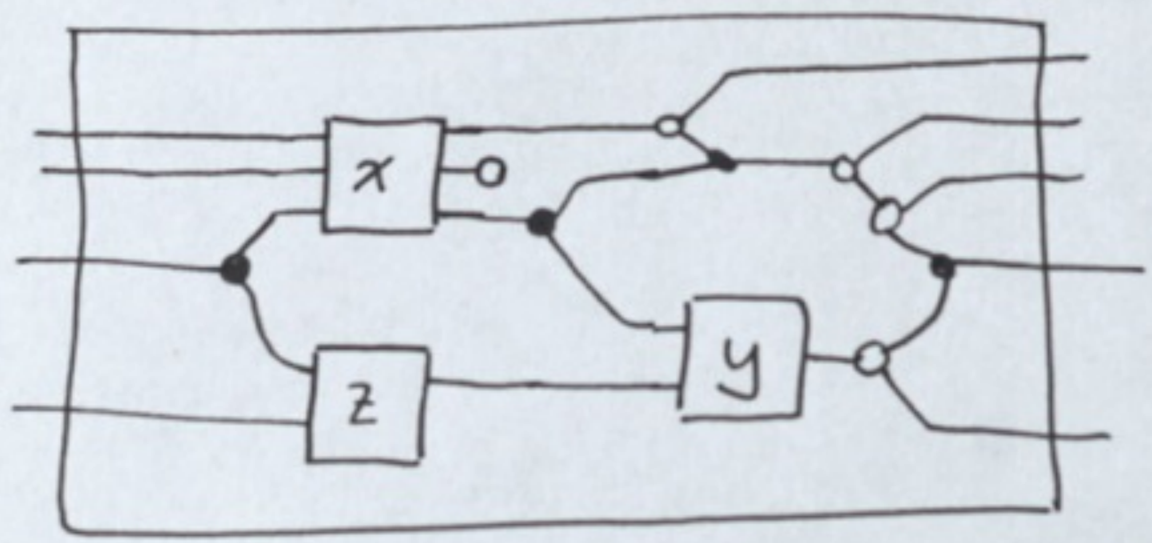
  variables.
- Then $(\Sigma + \mathrm{Var}, E)$ is again a PRO with (term) variables.

2. Making an Operad from a PRO

- Let (Σ, E) be a presentation of a PRO.
- Adjoin "enough" new generators $x : m \to n$ for every $m, n \in \mathbb{N}$.

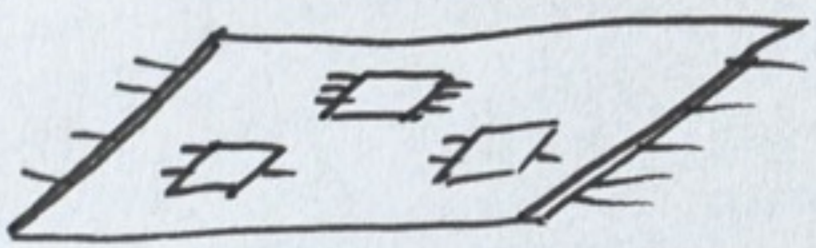
variables.

- Then $(\Sigma + Var, E)$ is again a PRO with (term) variables.

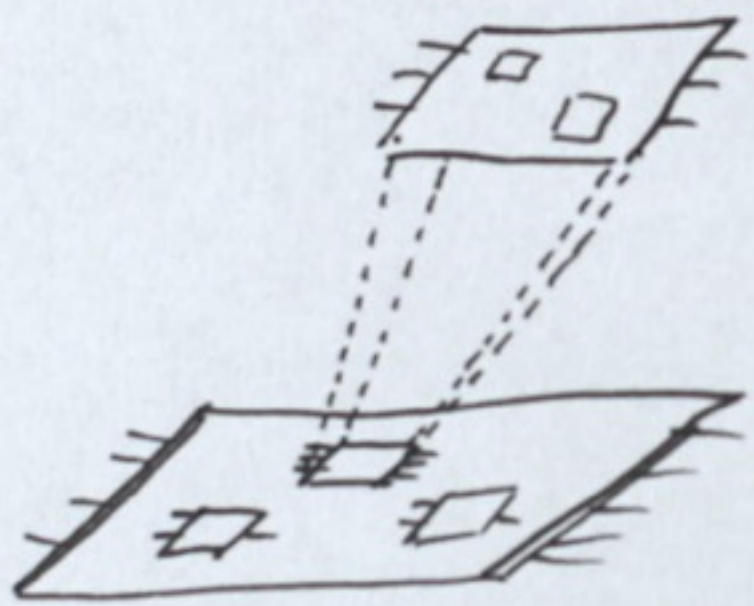


Assume variables only occur once (for now)

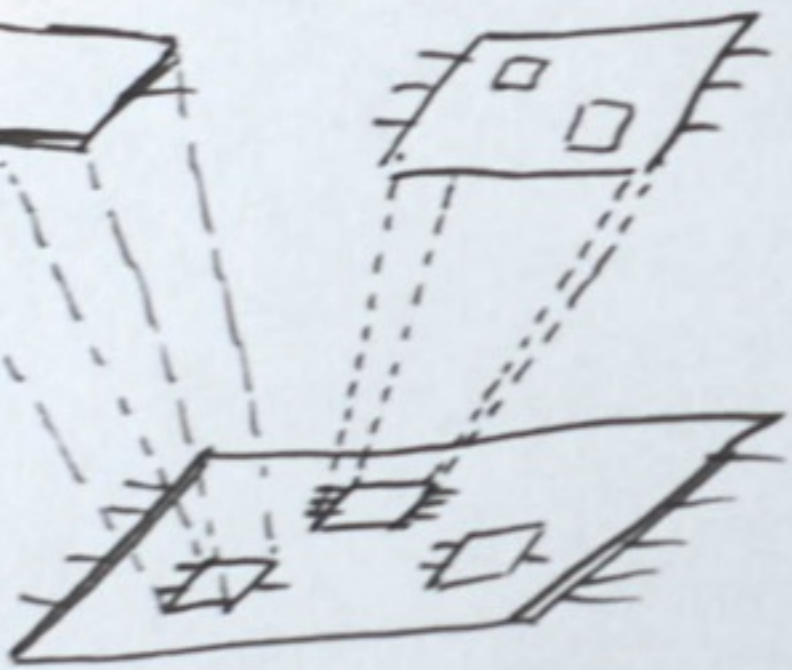
$$x:(3,3),\ y:(2,1),\ z:(2,1) \vdash f:(4,5)$$



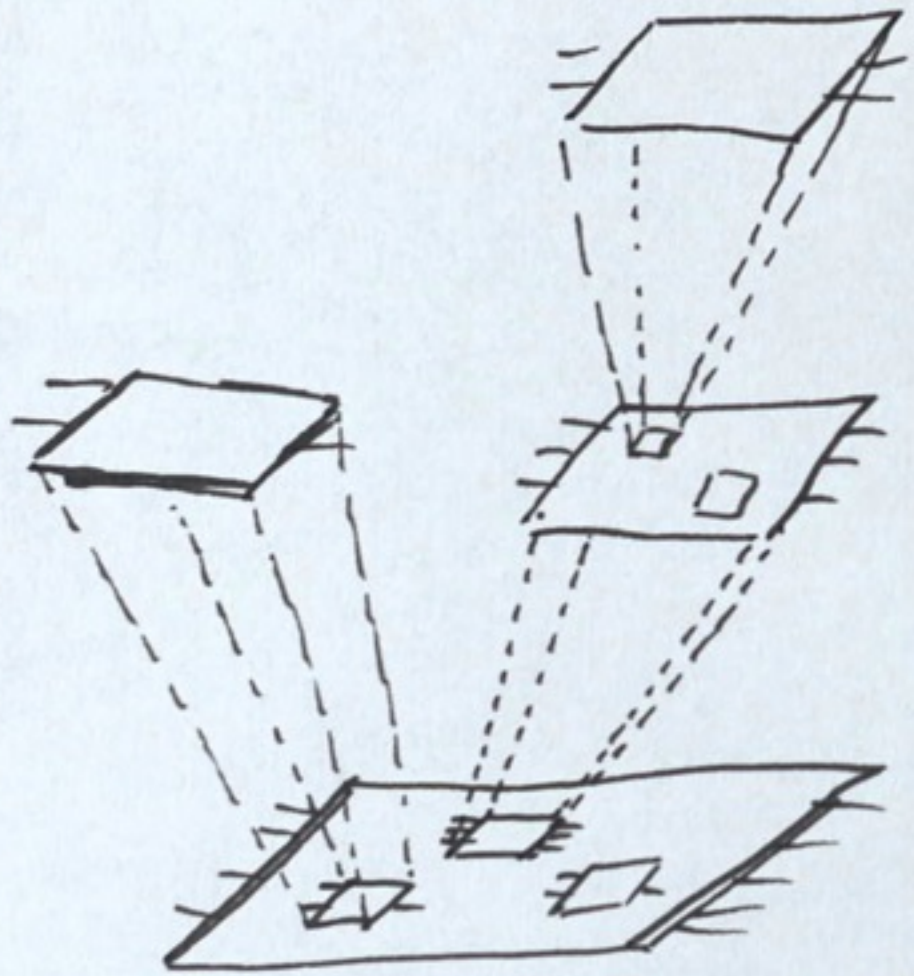
f



$f[g/x]$



$f[g/x][h/z]$



$f[g/x][h/z][k/w]$

etc.

Double Push-Out Rewriting

$$L = R$$

DOUBLE PUSH-OUT REWRITING

$$L \Rightarrow R$$

DOUBLE PUSH-OUT REWRITING

$$L \Rightarrow R$$

$$L \xleftarrow{i_1} \partial \xrightarrow{i_2} R$$

DOUBLE PUSH-OUT REWRITING

$$L \Rightarrow R$$

$$\begin{tikzcd}
L \arrow[d, hook, "m"'] & \partial \arrow[l, hook', "i_1"'] \arrow[r, hook, "i_2"] & R \\
G & &
\end{tikzcd}$$

Double Push-Out Rewriting

$$L \Rightarrow R$$

$$\begin{tikzcd}
L \arrow[d, hook, "m"'] & \partial \arrow[l, hook', "i_1"'] \arrow[r, hook, "i_2"] \arrow[d, hook] & R \\
G \arrow[ur, phantom, "\ulcorner", very near start] & G \setminus L \arrow[l, hook']
\end{tikzcd}$$

compute
push-out complement.

Double Push-Out Rewriting

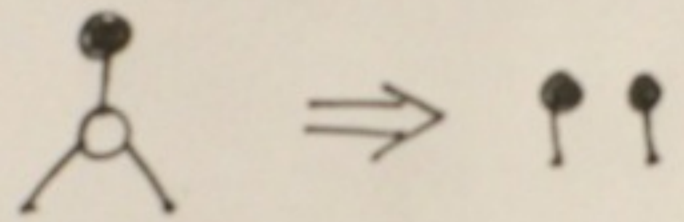
$$L \Rightarrow R$$

$$\begin{tikzcd}
L \arrow[d, hook, "m"'] & \partial \arrow[l, hook', "i_1"'] \arrow[r, hook, "i_2"] \arrow[d, hook, "j"'] \arrow[dl, phantom, "\urcorner", very near end] \arrow[dr, phantom, "\ulcorner", very near end] & R \arrow[d, hook] \\
G & G \setminus L \arrow[l, hook'] \arrow[r, hook] & G[R/L]
\end{tikzcd}$$

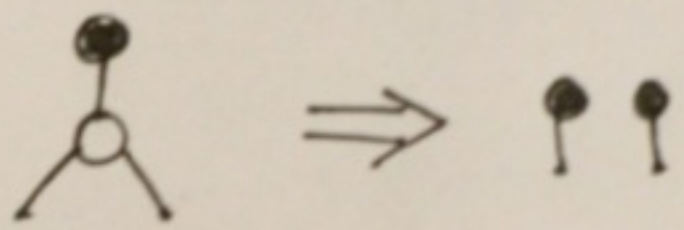
compute
push-out complement.

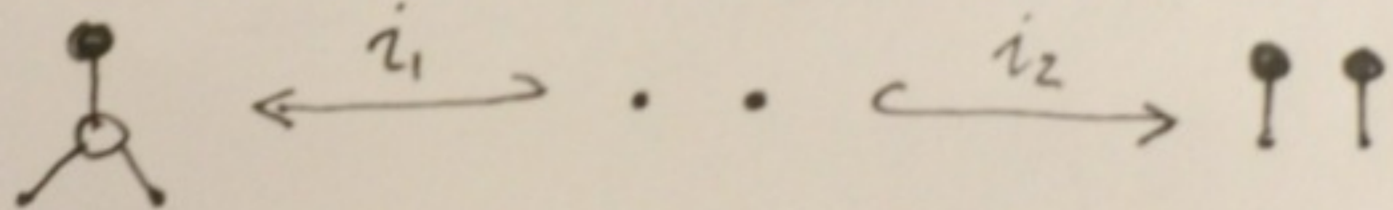
compute ~~po~~ pushout.

DPO REWRITING

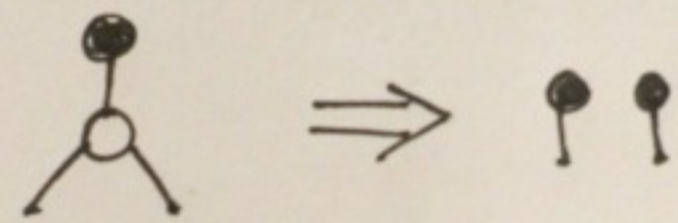


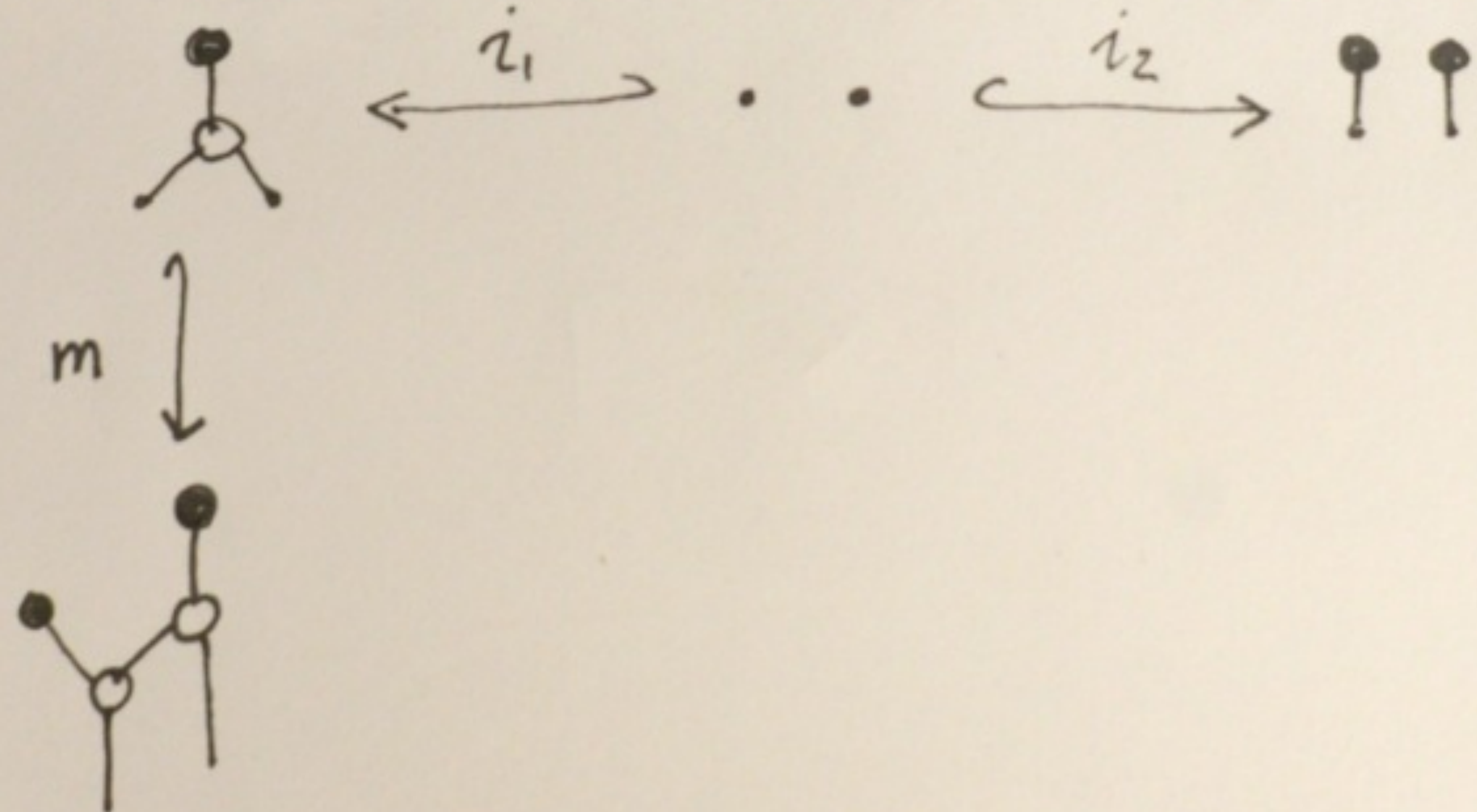
DPO REWRITING



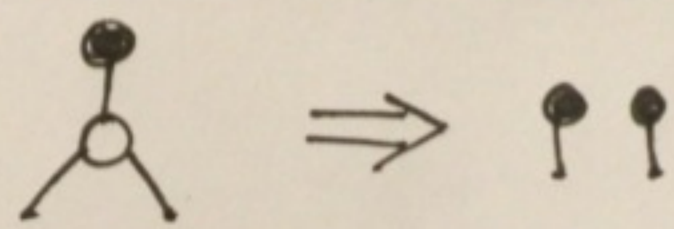


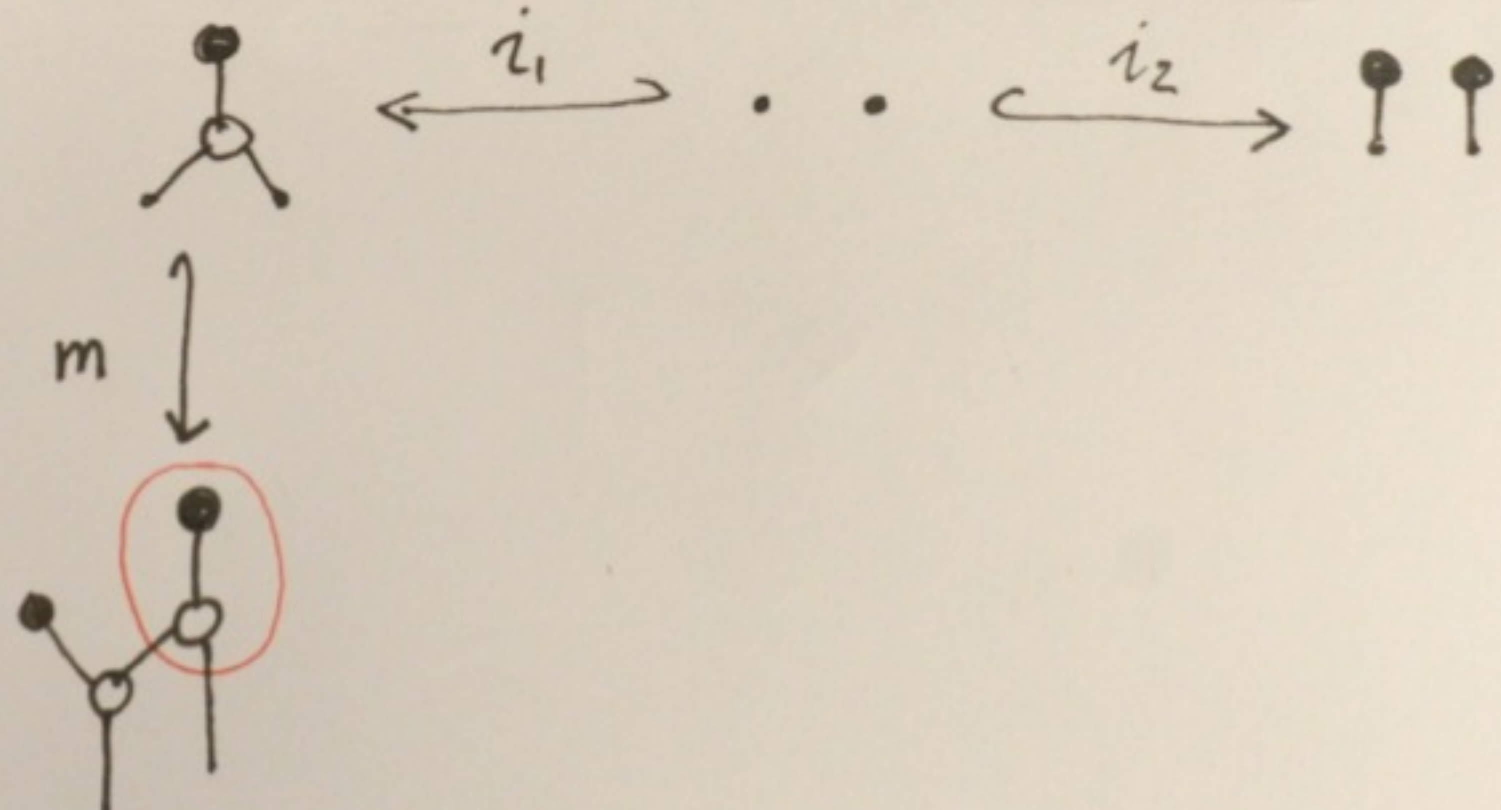
DPO REWRITING



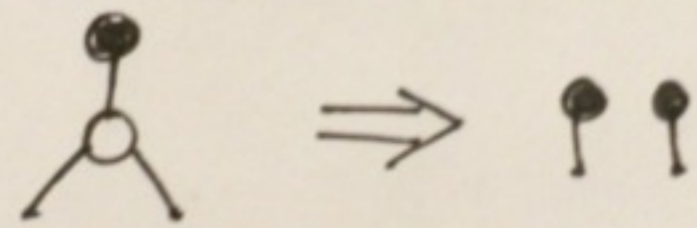


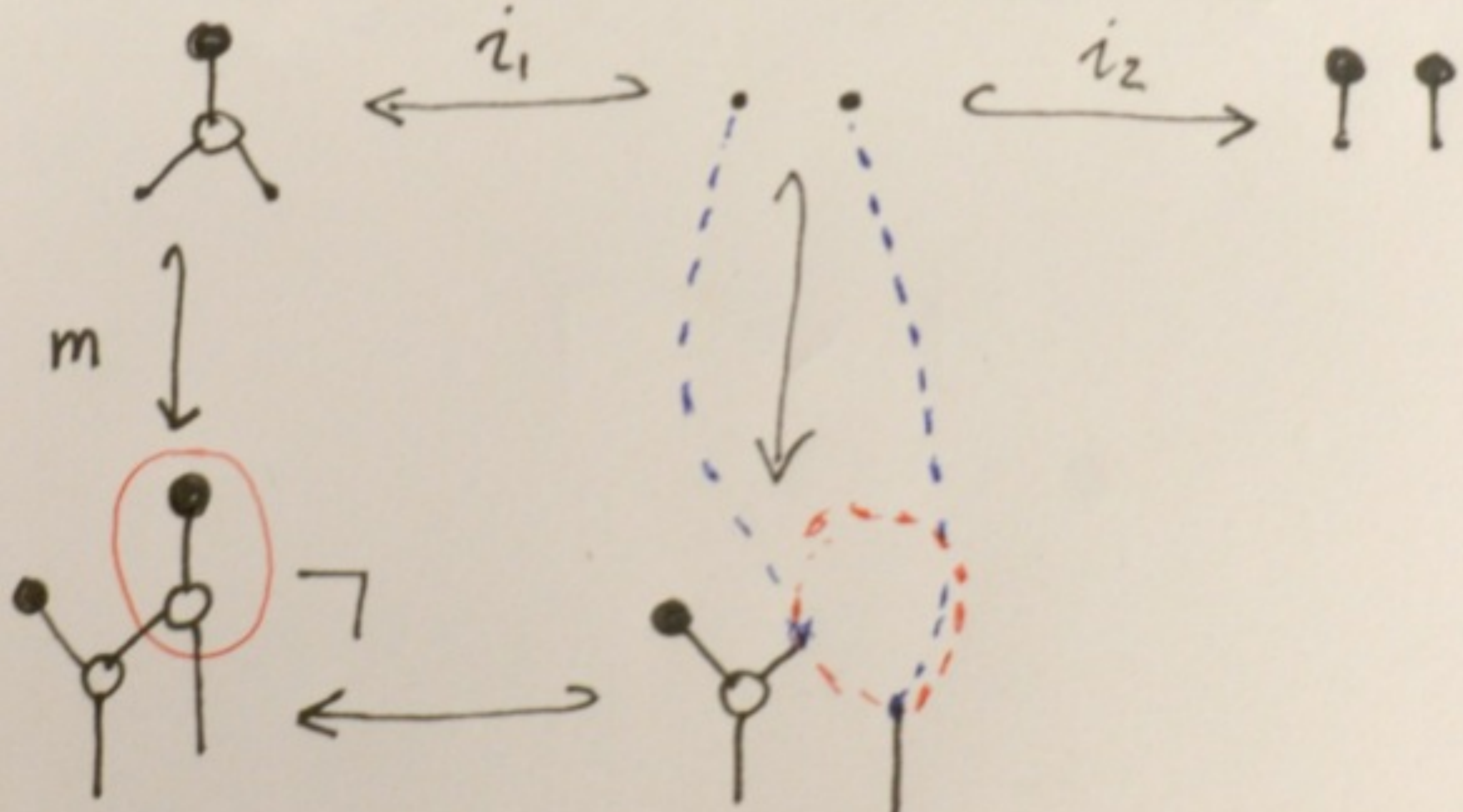
DPO REWRITING



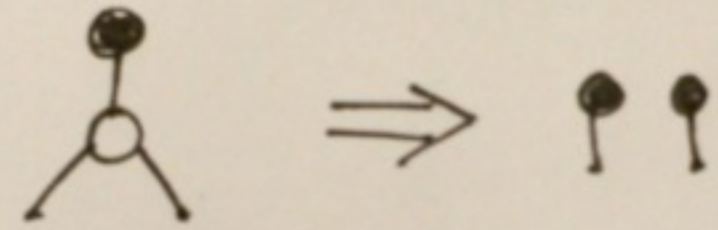


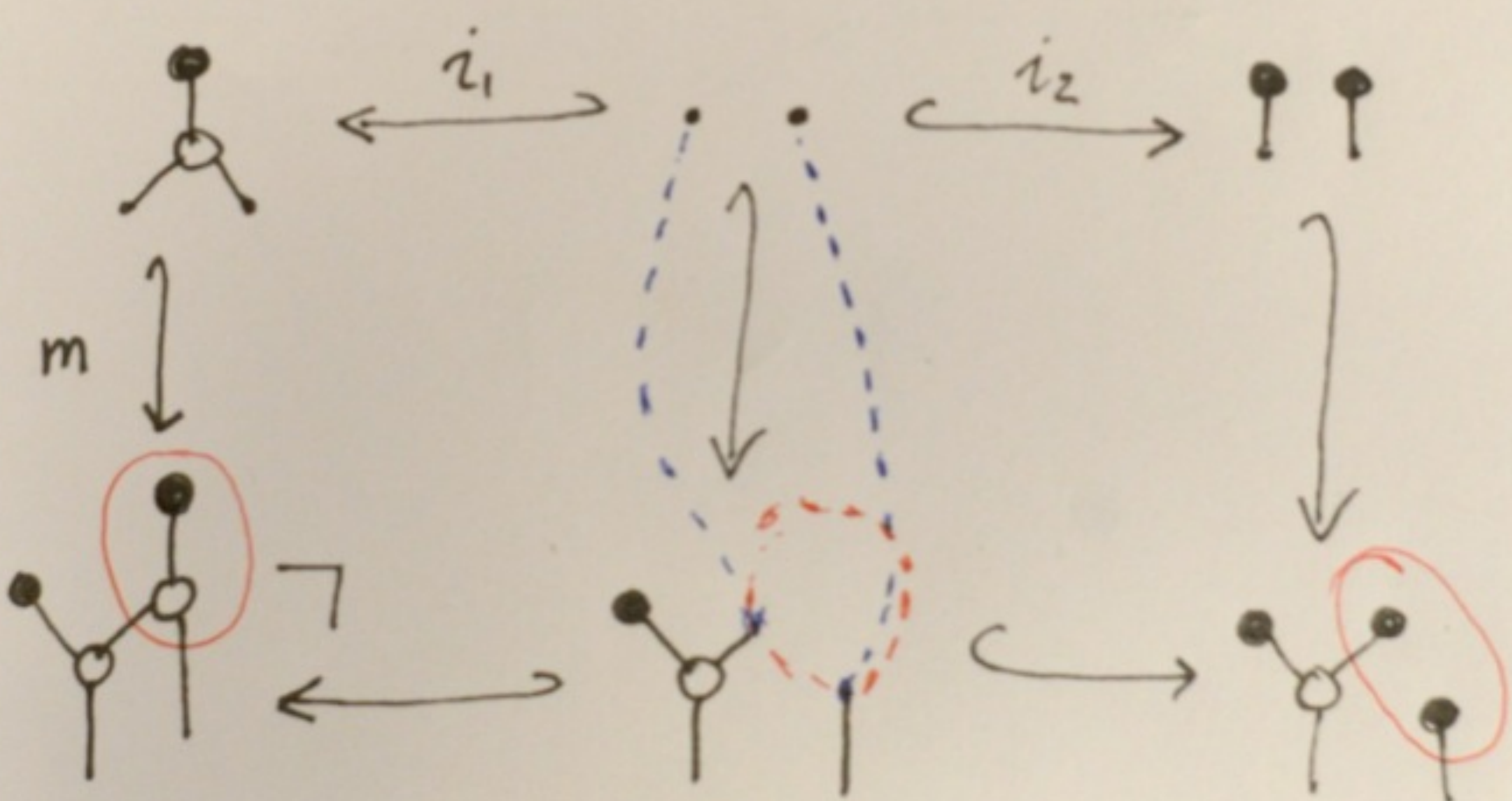
DPO REWRITING





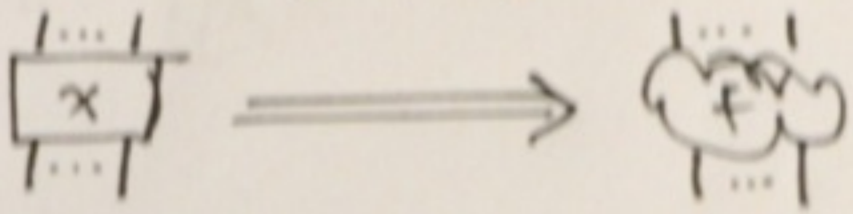
DPO REWRITING



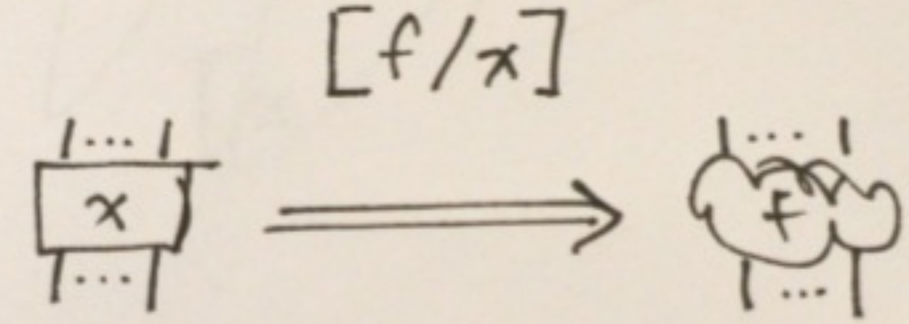


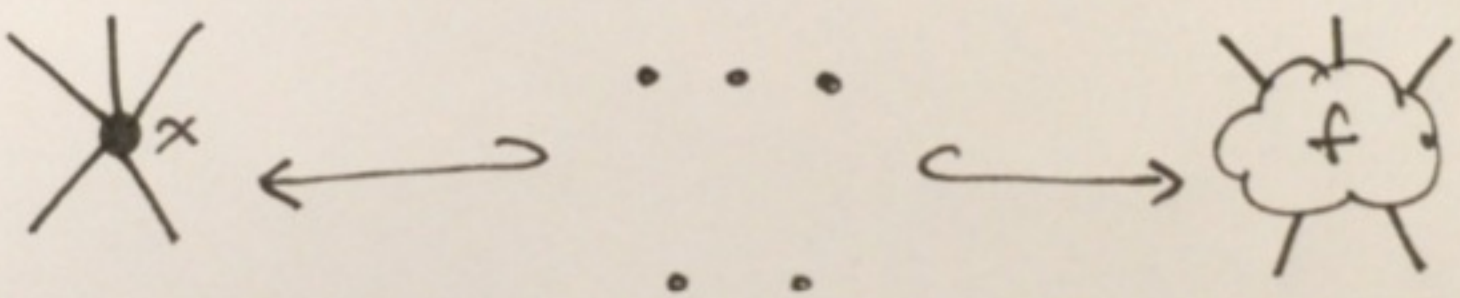
SUBSTITUTION VIA DPO

$[f/x]$



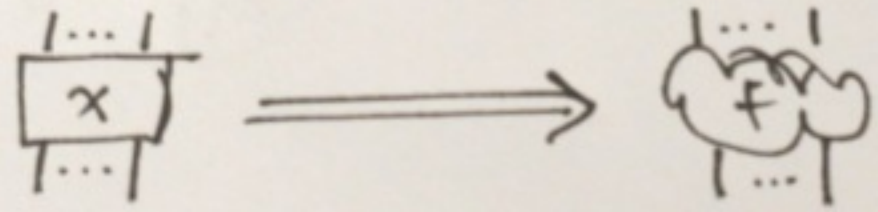
PLANE SUBSTITUTION VIA DPO

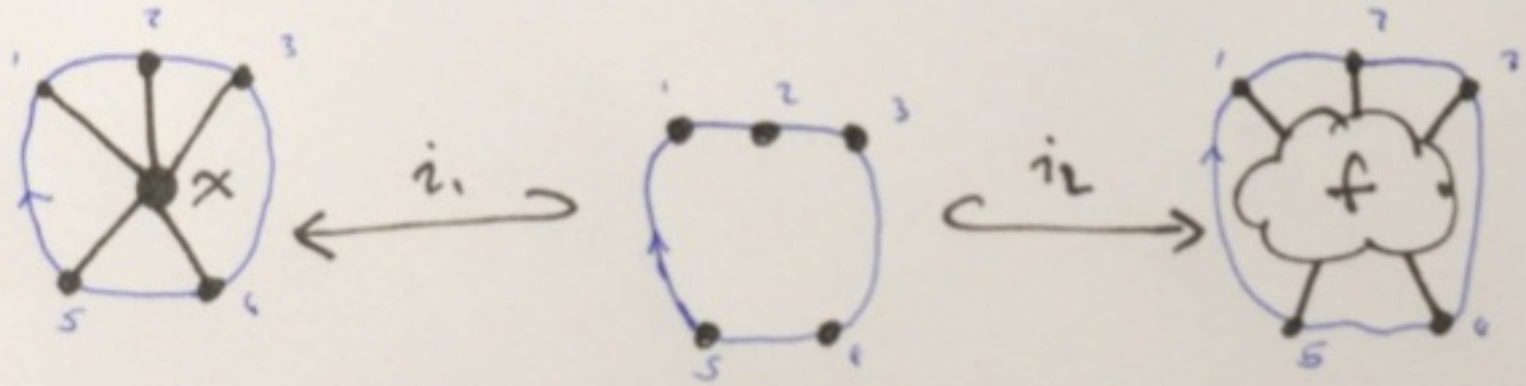




PLANE SUBSTITUTION VIA DPO

$[f/x]$

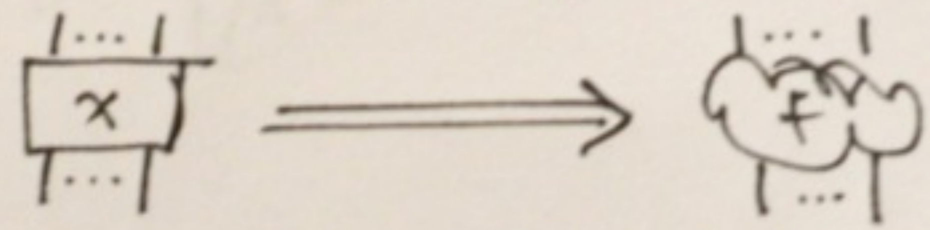


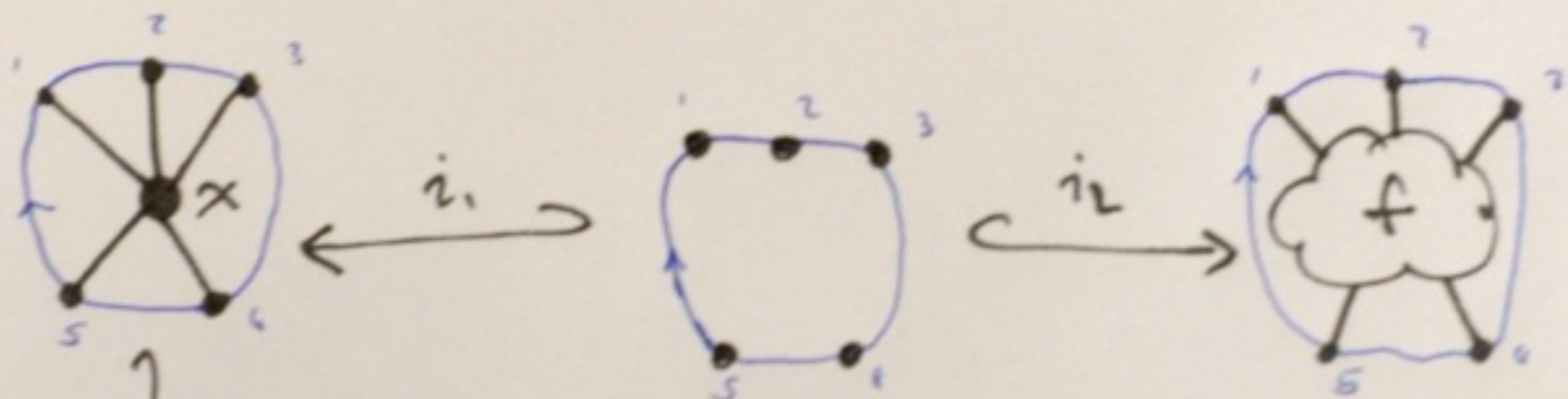


EMBEDDINGS BIJECTIVE ON THE BOUNDARY

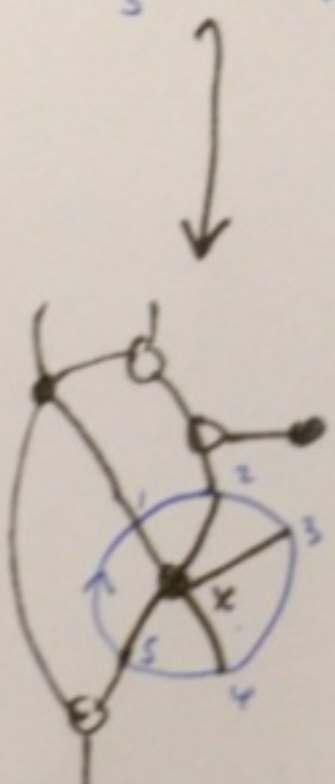
PLANE SUBSTITUTION VIA DPO

$[f/x]$



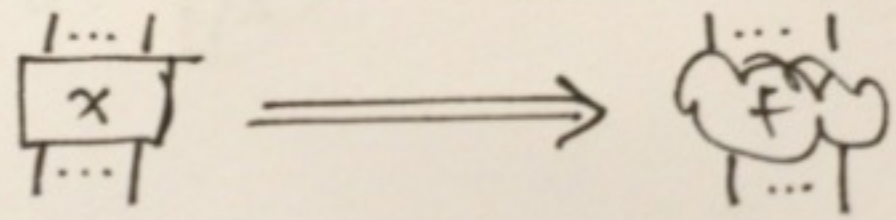


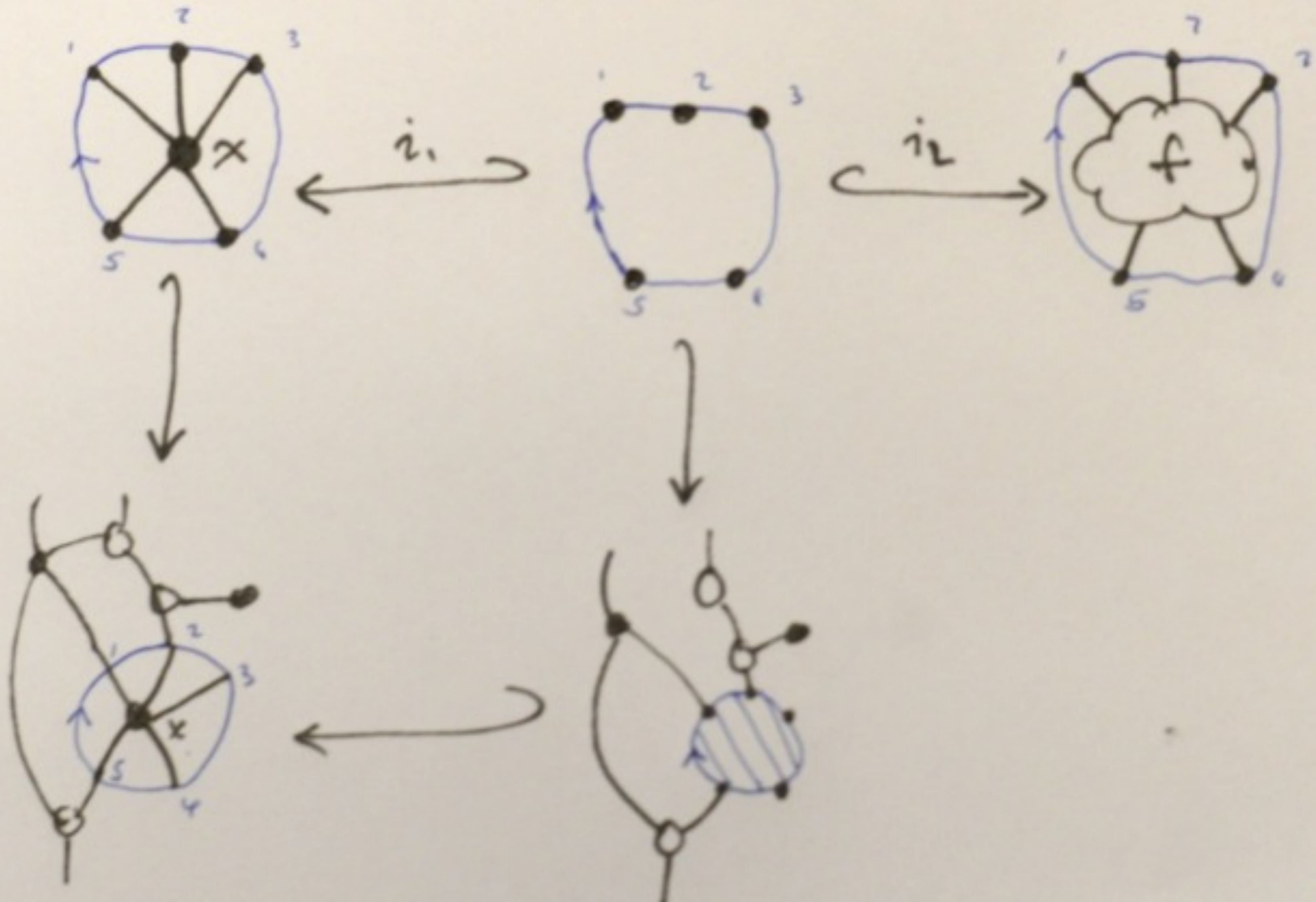
EMBEDDINGS BIJECTIVE ON THE BOUNDARY



PLANE SUBSTITUTION VIA DPO

$[f/x]$

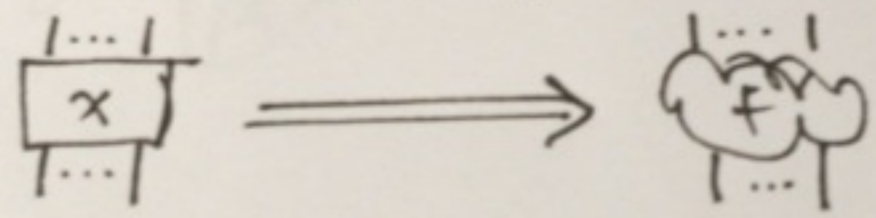


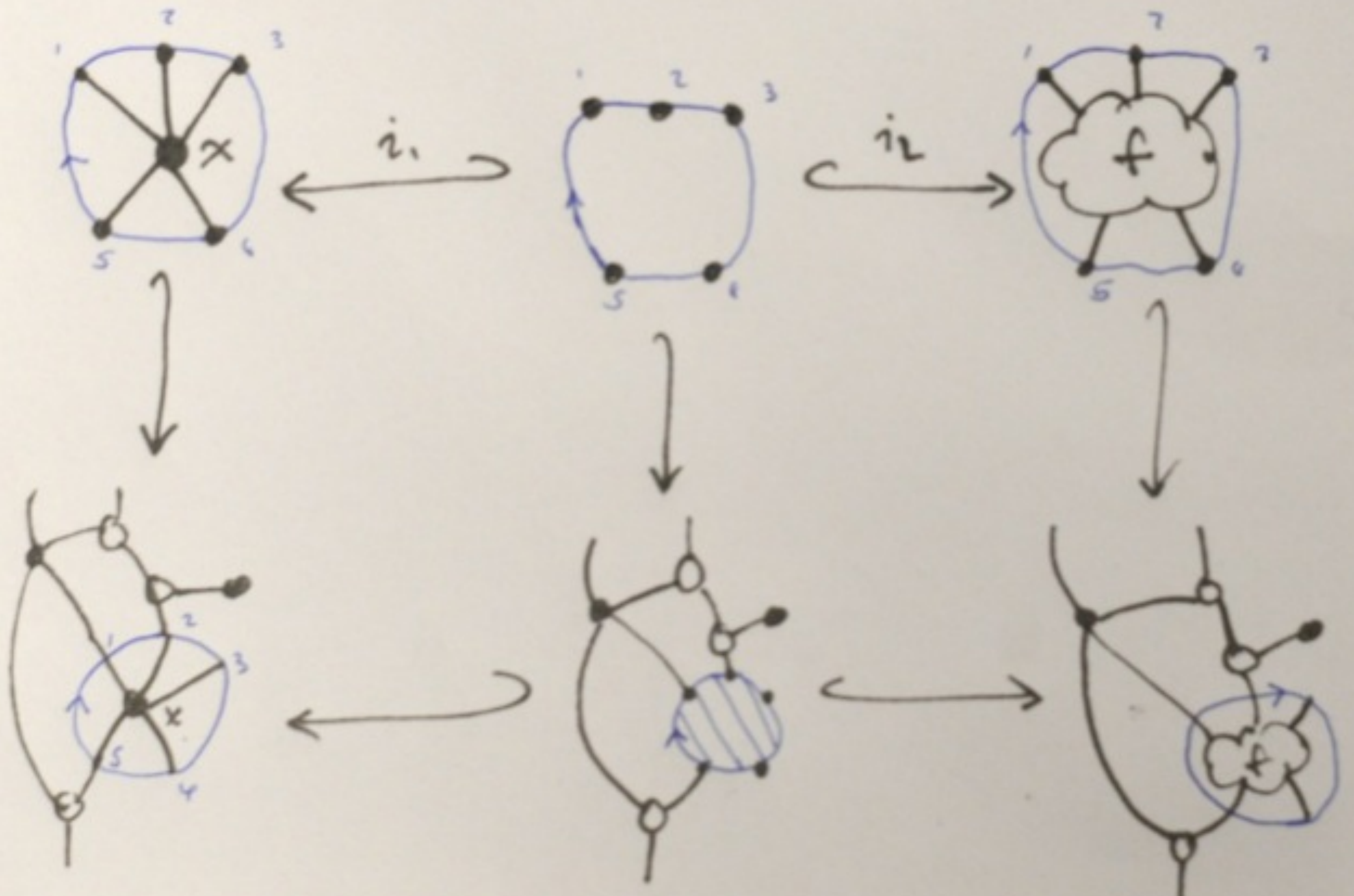


EMBEDDINGS BIJECTIVE ON THE BOUNDARY

PLANE SUBSTITUTION VIA DPO

$[f/x]$



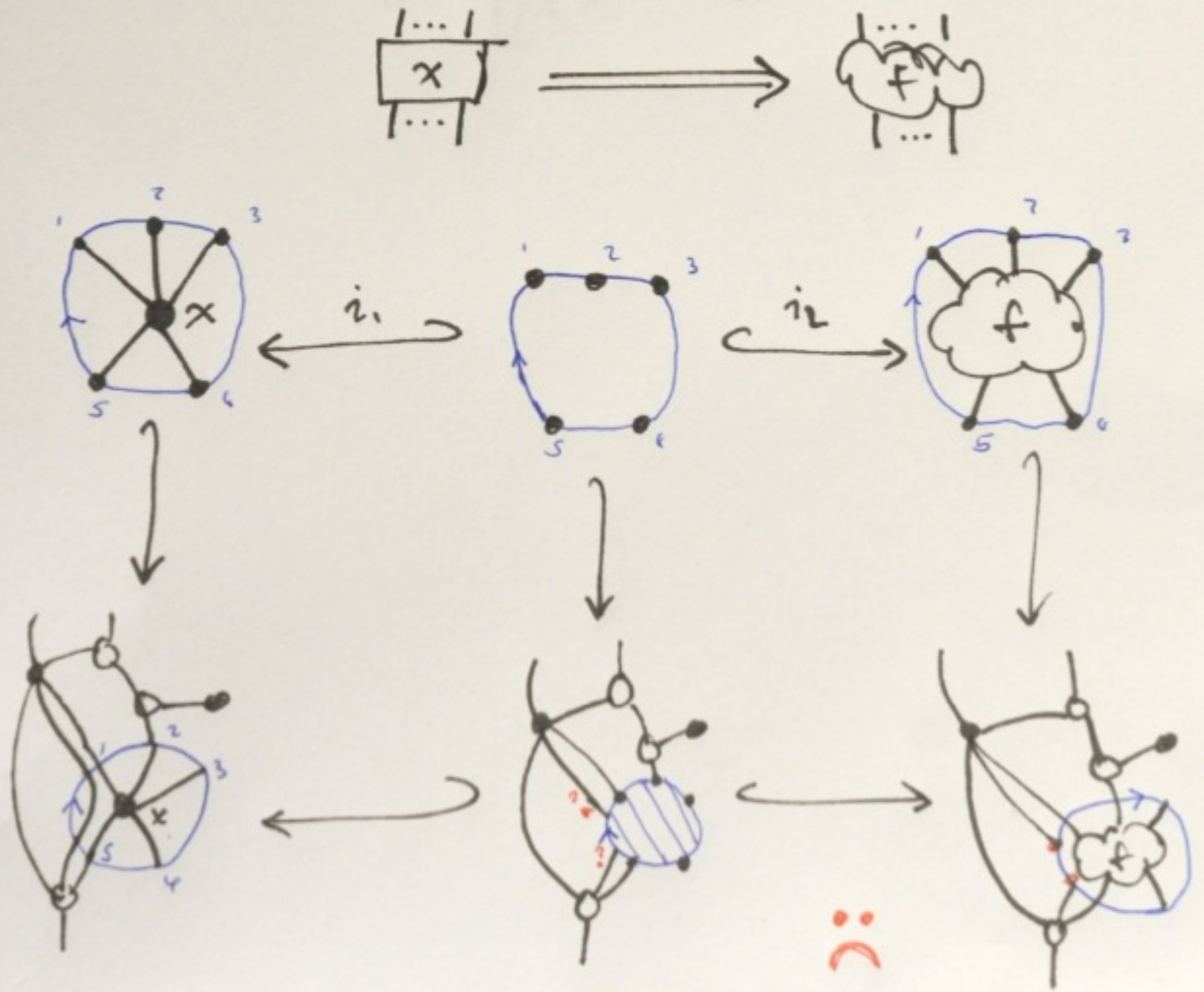


EMBEDDINGS BIJECTIVE ON THE BOUNDARY

See Slofstra

Plane Substitution via DPO

$[f/x]$



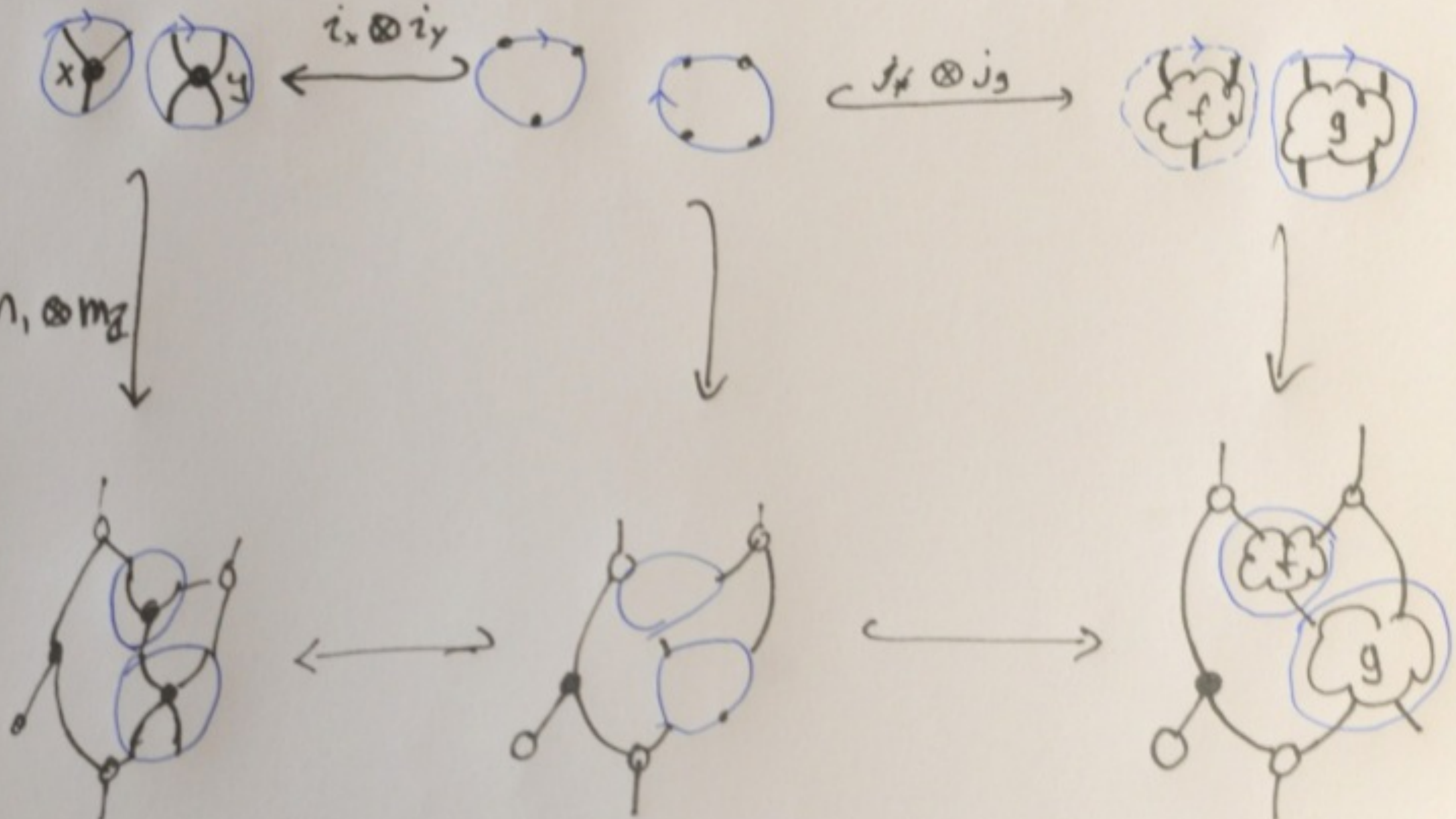
Embeddings bijective on the boundary

Bad!

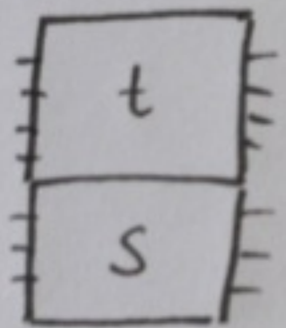
See Slofstra

PLANE SUBSTITUTION

$$[f/x,\ g/y]$$



Logical Rules

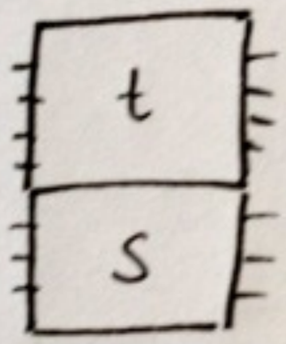


"tensor"

$$\frac{\bar{x}:\Delta \vdash t:A \qquad \bar{y}:\Gamma \vdash s:B}{\bar{x}:\Delta, \bar{y}:\Gamma \vdash t\otimes s : A\otimes B}$$

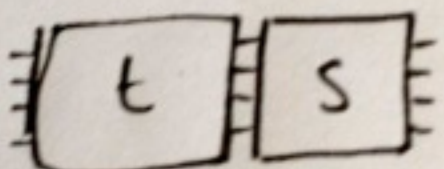
LOGICAL RULES

"tensor"



$$\frac{\bar{x}:\Delta \vdash t : A \qquad \bar{y}:\Gamma \vdash s : B}{\bar{x}:\Delta,\ \bar{y}:\Gamma \vdash t \otimes s : A \otimes B}$$

"composite"



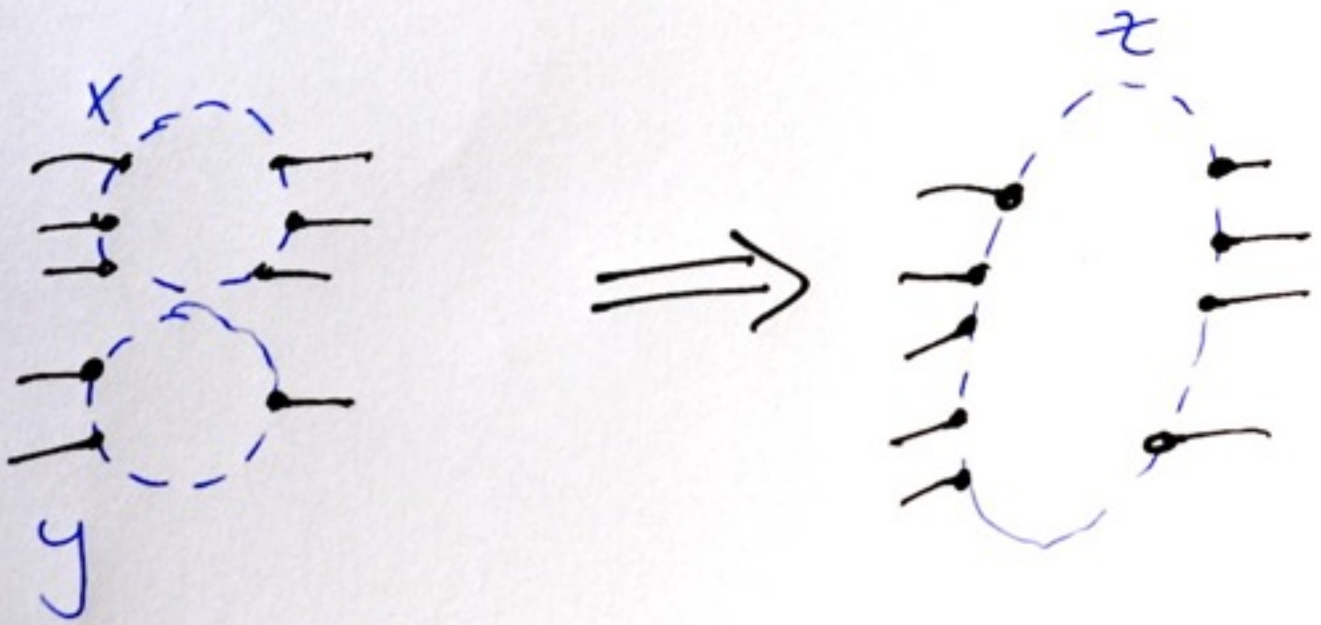
$$\frac{\bar{x}:\Delta \vdash t : (n,m) \qquad \bar{y}:\Gamma \vdash s : (m,k)}{\bar{x}:\Delta,\ \bar{y}:\Gamma \vdash s \circ t : (n,k)}$$

NOT ALLOWED

$$\frac{x : A,\ y : B \ \vdash\ t : C}{z : A \otimes B \ \vdash\ \text{let } z = x \otimes y \text{ in } t \ : C}$$

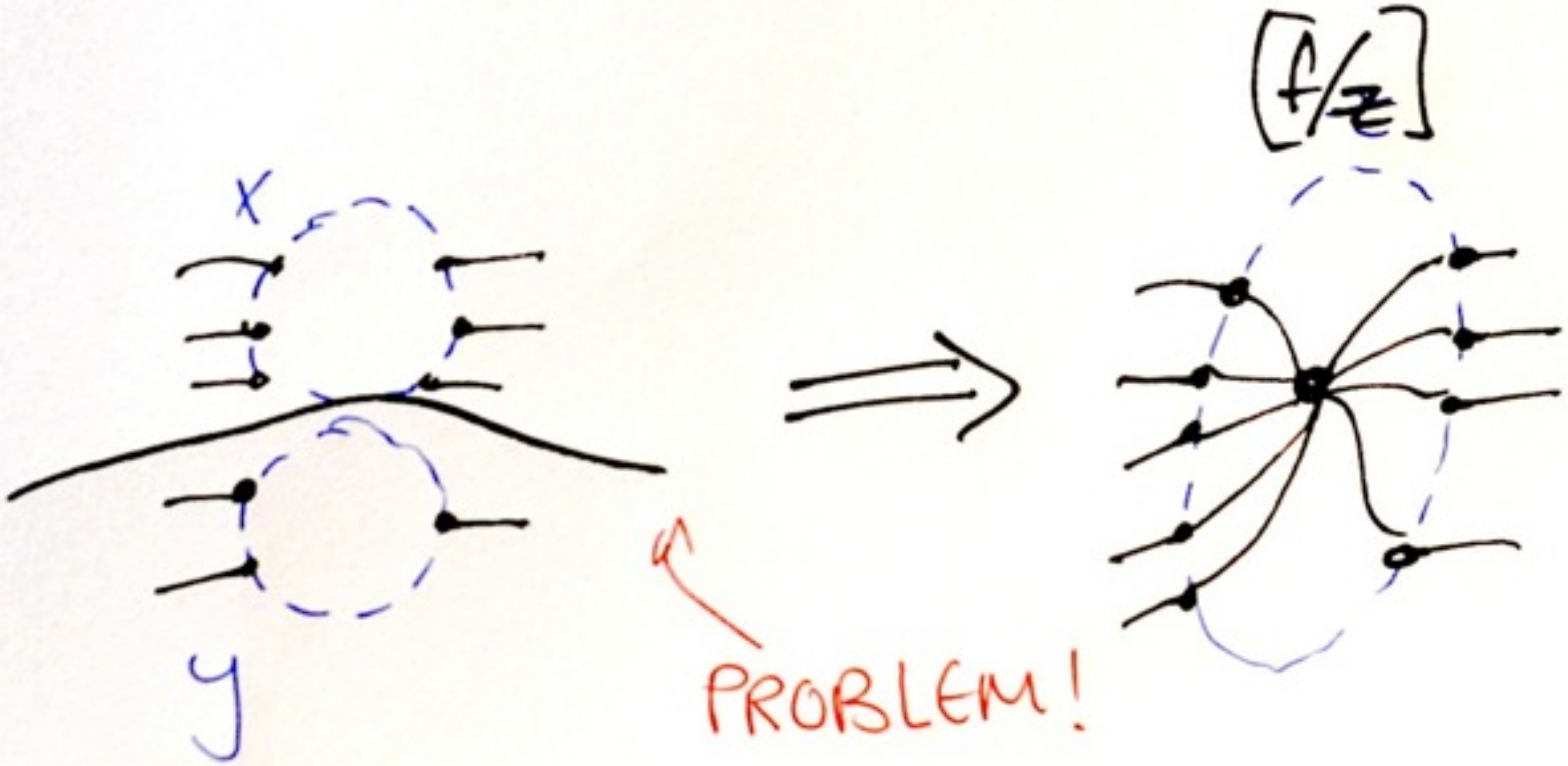
NOT ALLOWED

$$\frac{x : A,\ y : B \ \vdash\ t : C}{z : A \otimes B \ \vdash\ \text{let } z = x \otimes y \text{ in } t \ : C}$$



NOT ALLOWED

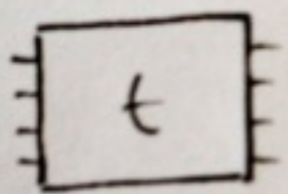
$$\frac{x : A,\ y : B \ \vdash\ t : C}{z : A \otimes B \ \vdash\ \mathrm{let}\ z = x \otimes y\ \mathrm{in}\ t \ : C}$$



DITCHING LINEARITY

LOGICAL RULES

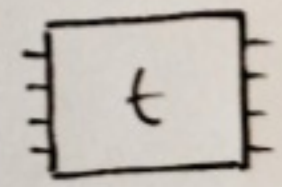
weakening



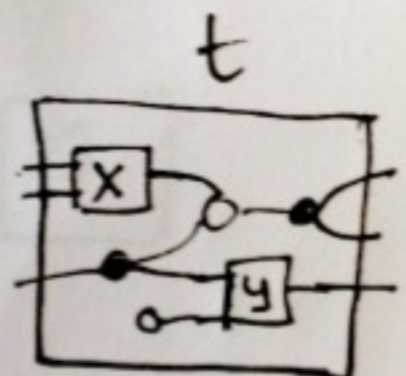
$$\frac{\bar{x} : \Delta \vdash t : A}{y : B, \bar{x} : \Delta \vdash t : A}$$

LOGICAL RULES

weakening



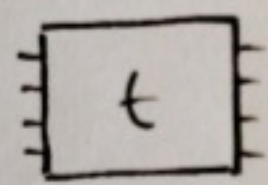
$$\frac{\bar{x} : \Delta \vdash t : A}{y : B, \bar{x} : \Delta \vdash t : A}$$



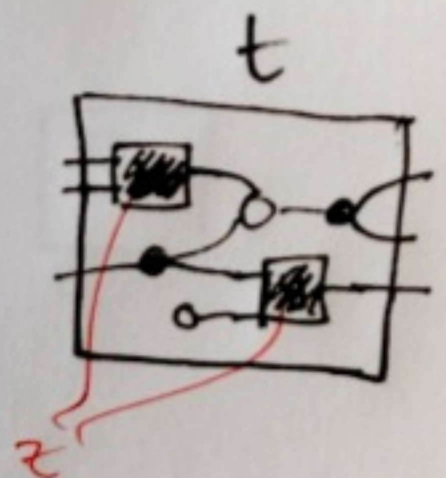
$$\frac{x : A, y : A \vdash t : B}{z : A \vdash t[z/x, z/y]}$$

LOGICAL RULES

weakening



$$\frac{\bar{x}:\Delta \vdash t:A}{y:B,\ \bar{x}:\Delta \vdash t:A}$$



$$\frac{x:A,\ y:A \vdash t:B}{z:A \quad \vdash t[z/x,\ z/y]}$$

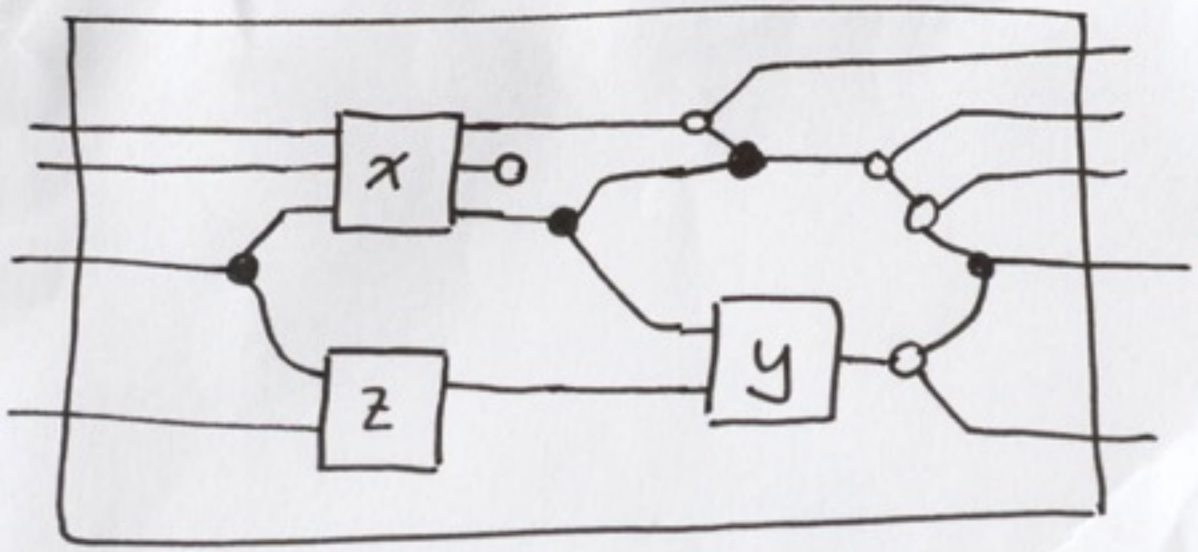
"CONTRACTION"

SUMMARY PT. 1

1. SUBSTITUTION AND OPERATIONS IN UNDERLYING PRO form a "monoidal++" operad.

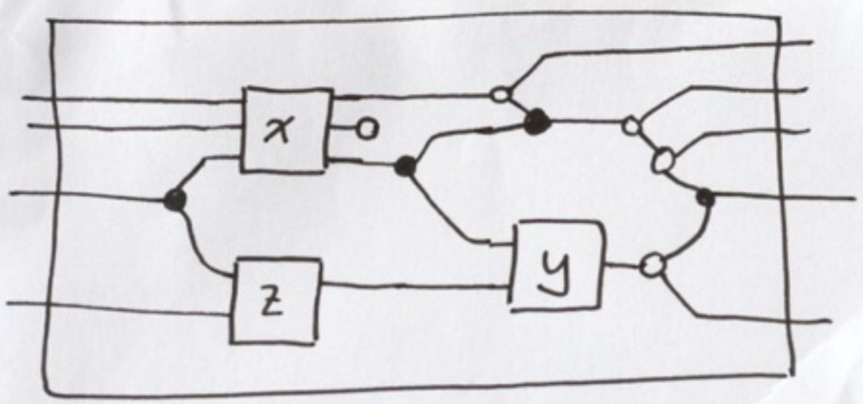
2. Variable manipulations give a cocommutative comonoid. But don't allow $\otimes$.

3. PATTERN-MATCHING.

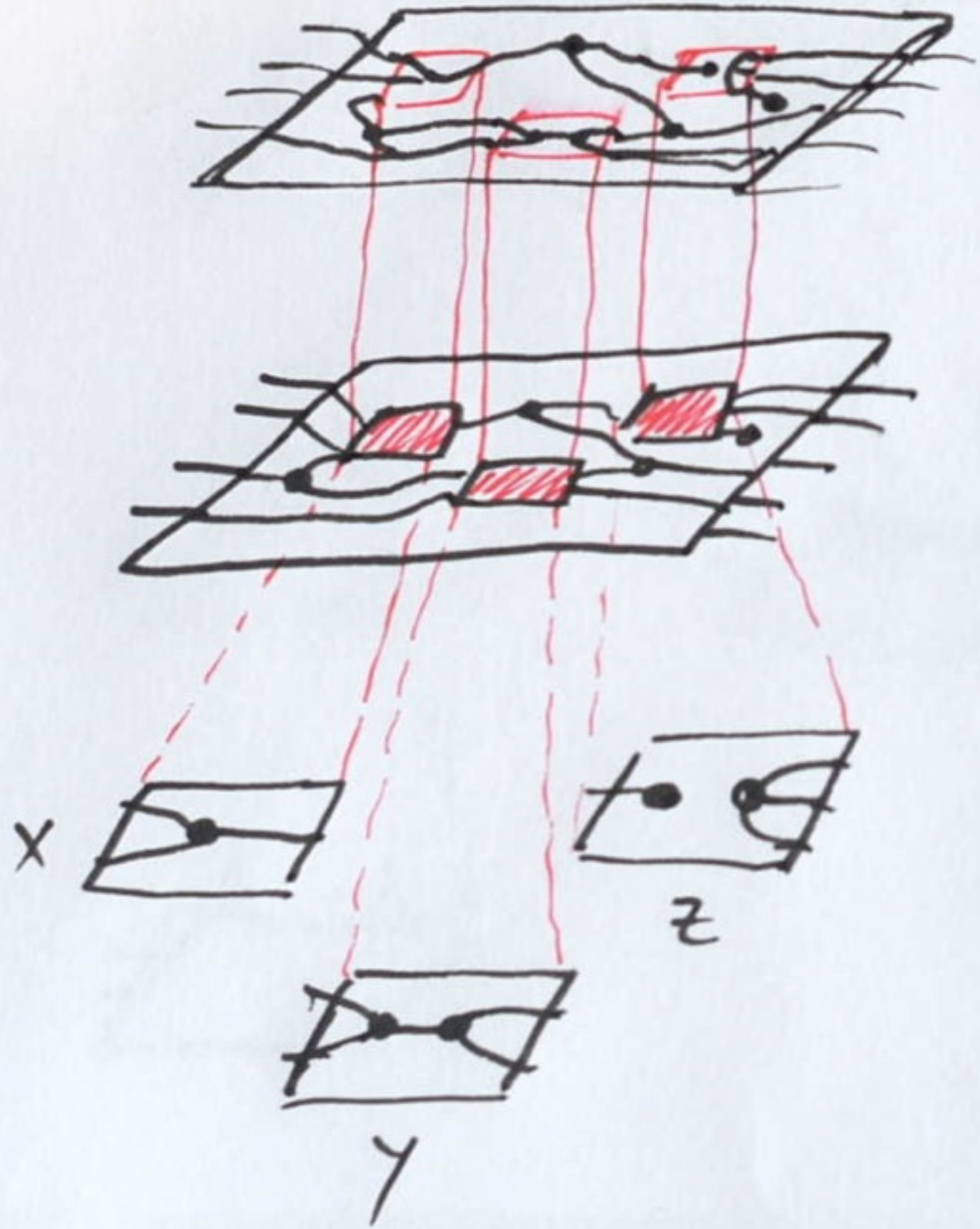


$x:(3,3),\ y:(2,1),\ z:(2,1) \vdash f:(4,5)$

3. PATTERN-MATCHING



$x:(3,3),\ y:(2,1),\ z:(2,1) \vdash f:(4,5)$

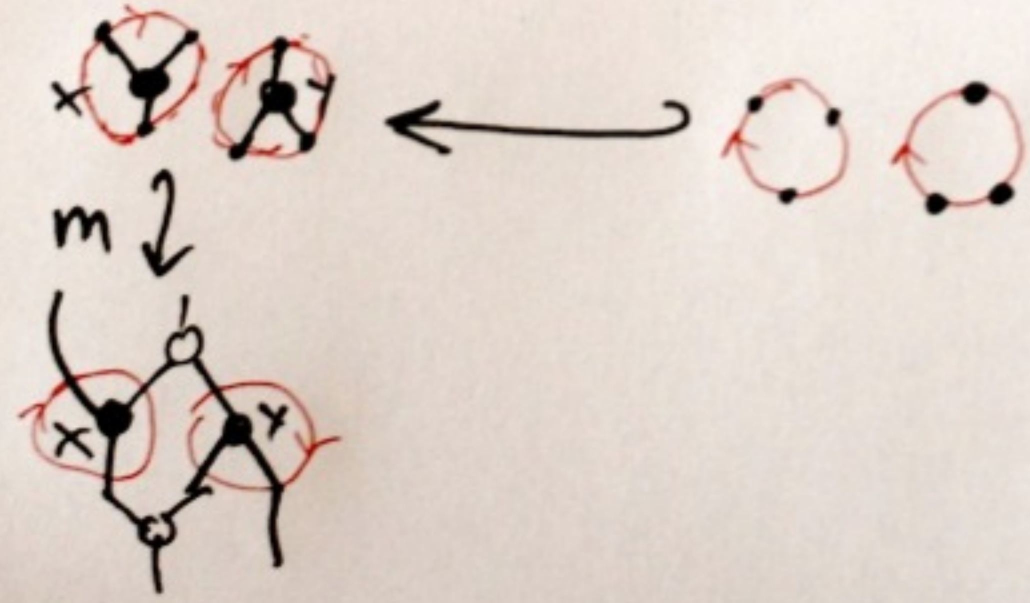


3. PATTERN MATCHING

$$\begin{tikzcd} V \arrow[d, hook, "m"'] & \partial \arrow[l, hook', "i"'] \\ G & \end{tikzcd}$$

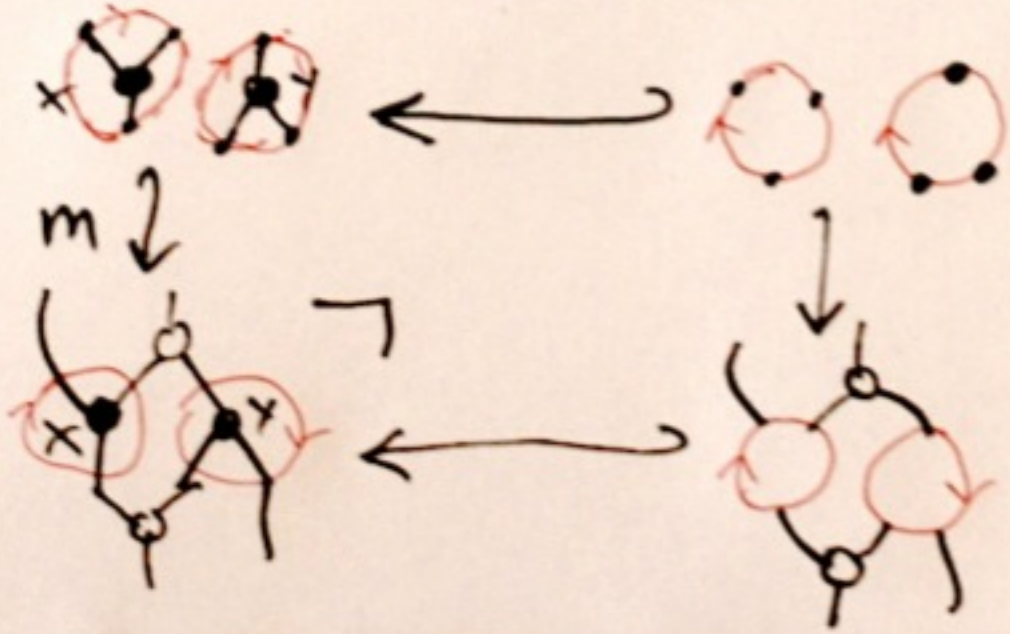
3. Pattern Matching

$$\begin{tikzcd} V \arrow[d, hook, "m"'] & \partial \arrow[l, hook', "i"'] \\ G \end{tikzcd}$$



3. Pattern Matching

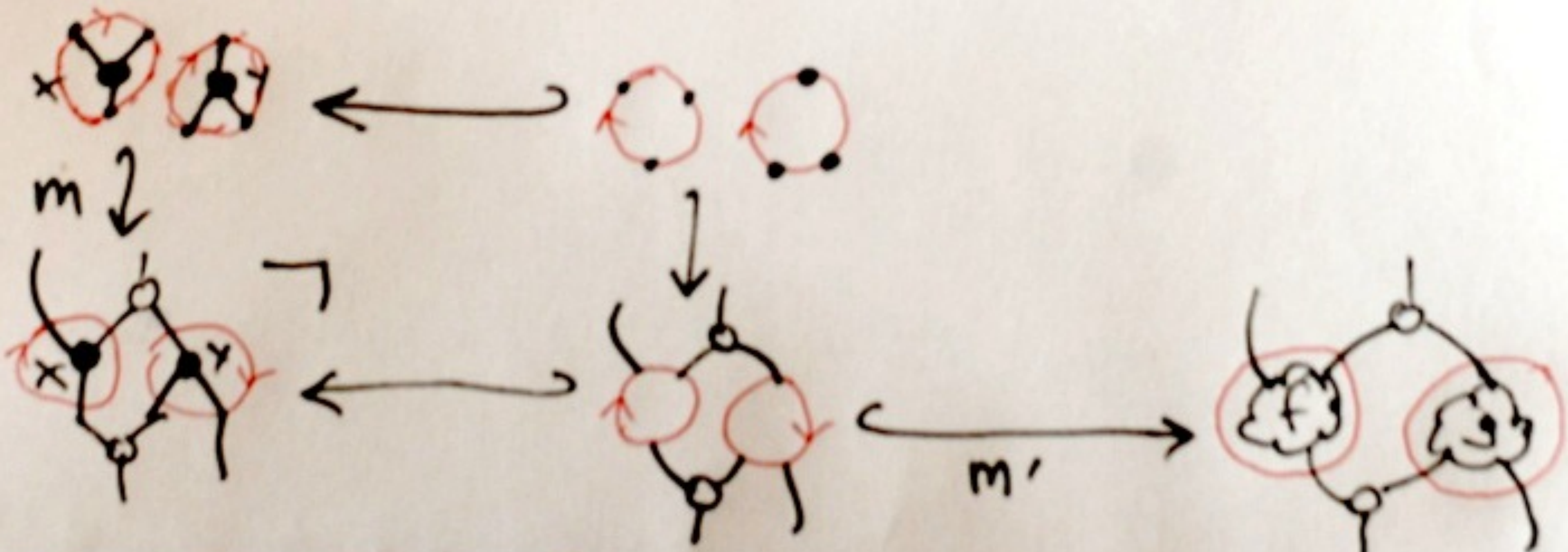
$$\begin{tikzcd}
V \arrow[d, hook, "m"'] & \partial \arrow[l, hook', "i"'] \arrow[d, hook] \\
G \arrow[ur, phantom, "\urcorner", very near start] & G \setminus V \arrow[l, hook']
\end{tikzcd}$$



3. Pattern Matching

$$\begin{tikzcd}
V \arrow[d, "m"'] & \partial \arrow[l, hook', "i"'] \arrow[d, hook] & \\
G \arrow[ur, phantom, very near start, "\urcorner"] & G \setminus V \arrow[l, hook'] \arrow[r, hook, "m'"'] & H
\end{tikzcd}$$

suppose this exists.

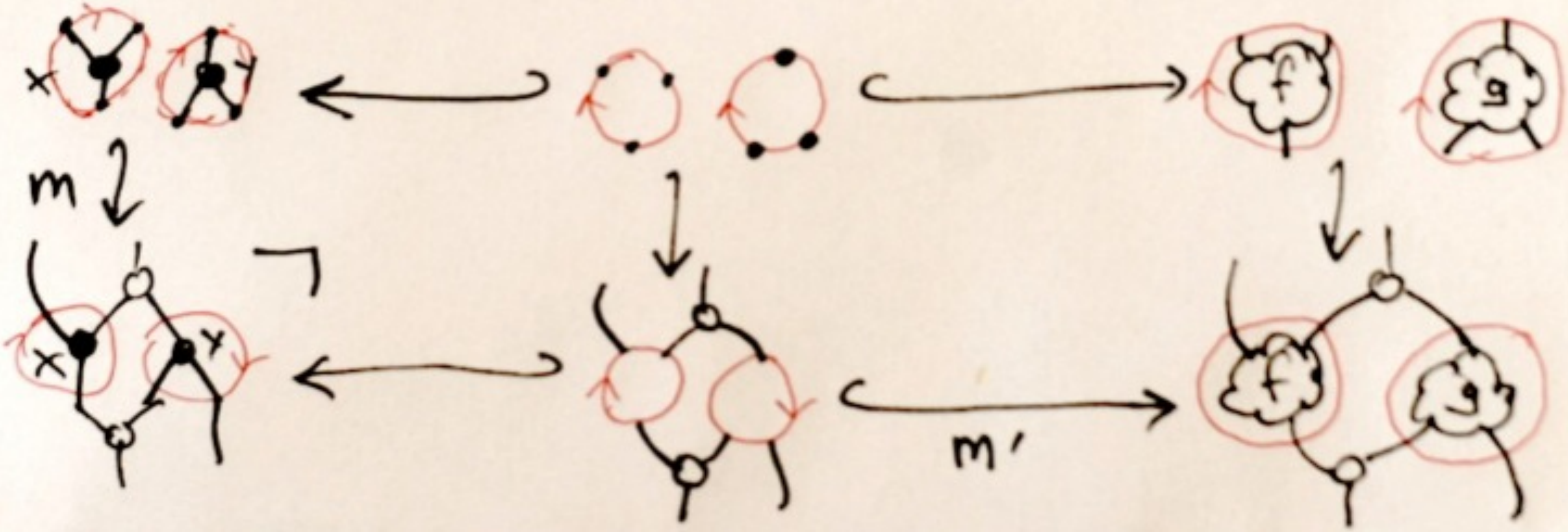


3. Pattern Matching

compute this pushout complement.

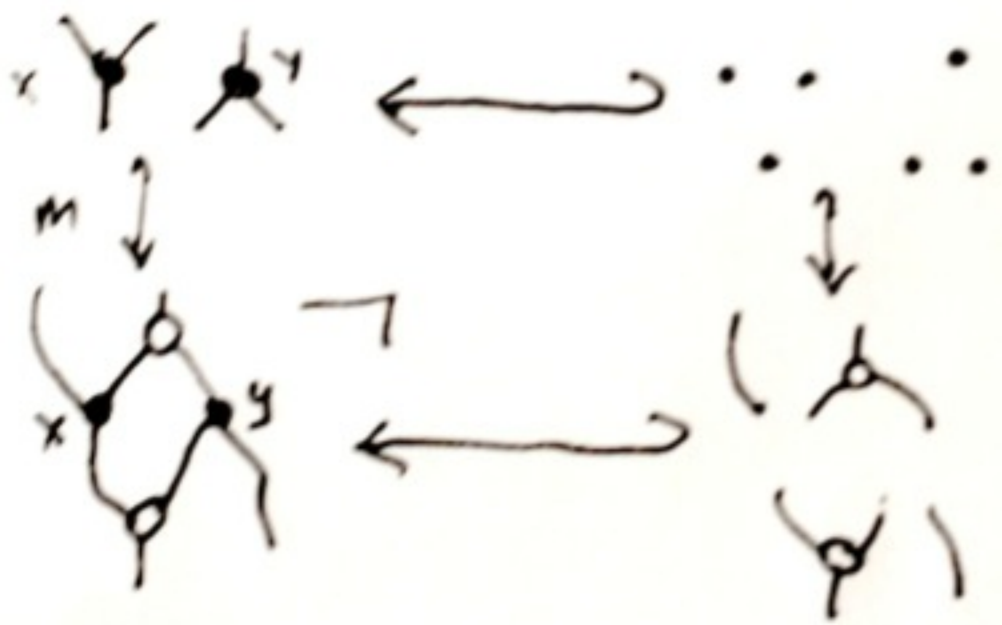
$$
\begin{tikzcd}
V \arrow[d, hook, "m"'] & \partial \arrow[l, hook', "i"'] \arrow[r, hook, "i'"] \arrow[d, hook] & H \setminus (G \setminus V) \arrow[d, hook] \\
G \arrow[ur, phantom, "\lrcorner", very near start] & G \setminus V \arrow[l, hook'] \arrow[r, hook, "m'"'] & H \arrow[ul, phantom, "\ulcorner", very near start]
\end{tikzcd}
$$

suppose this exists.



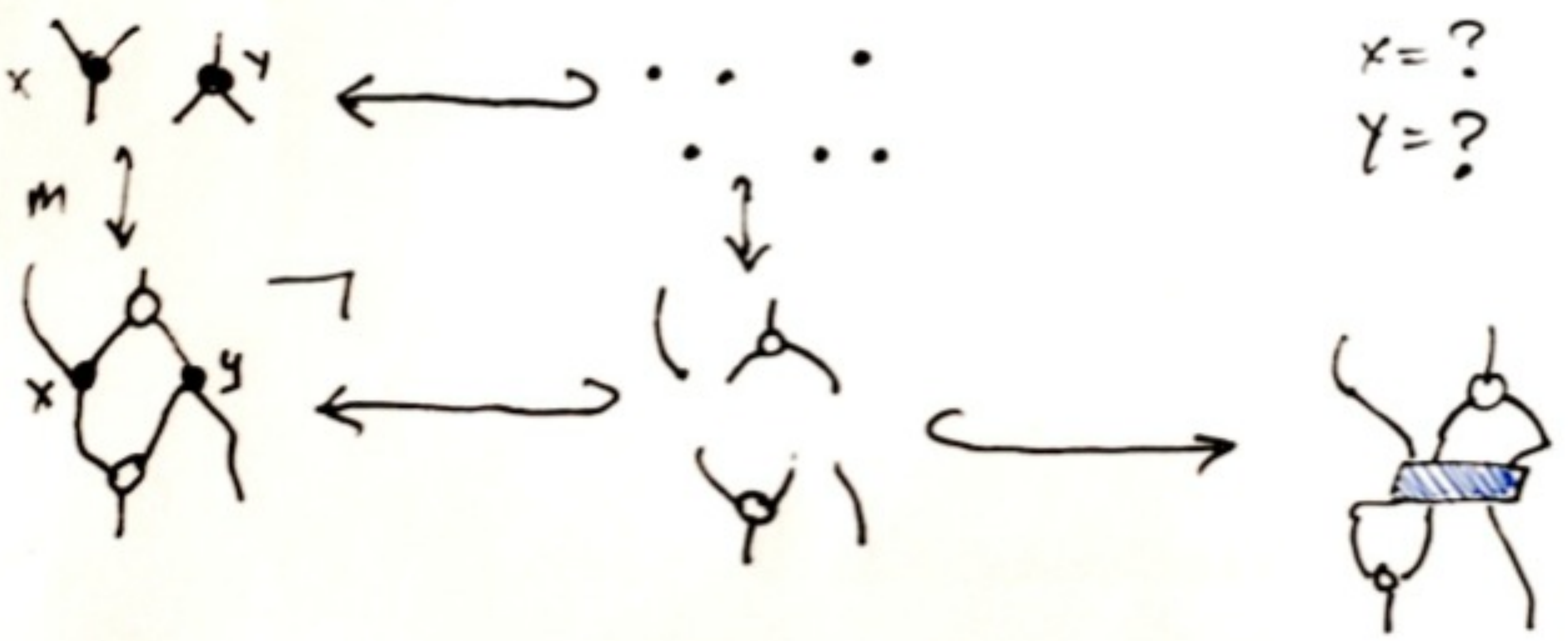
3. PATTERN MATCHING

NOTE PRESERVATION OF BOUNDARY CURVE IS ESSENTIAL:



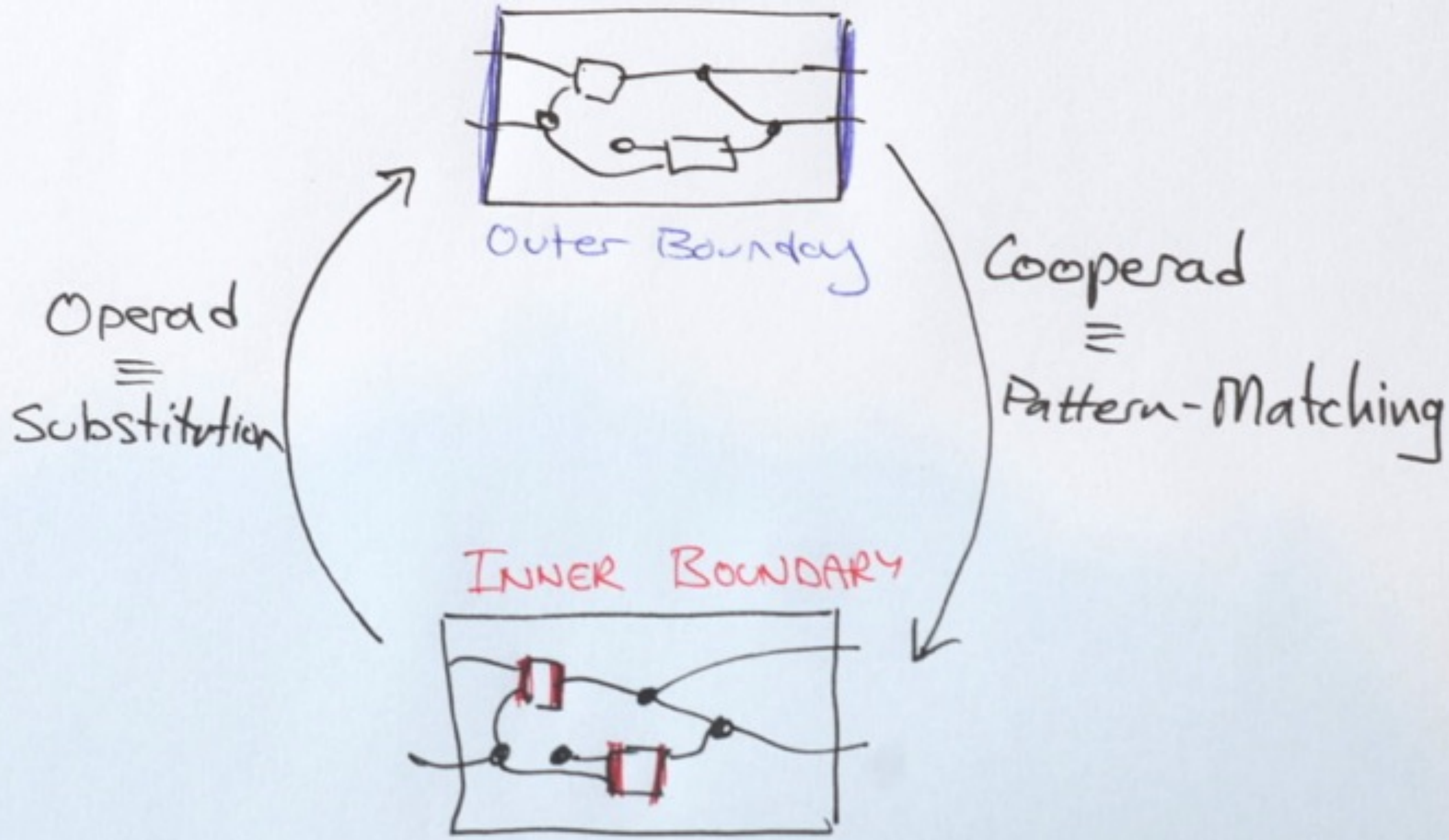
3. PATTERN MATCHING

NOTE PRESERVATION OF BOUNDARY <u>CURVE</u> IS ESSENTIAL:



PUTTING IT TOGETHER

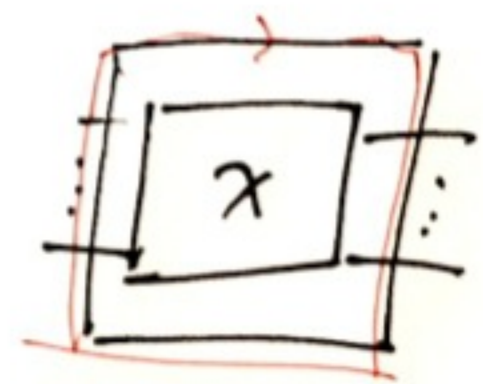
STRING DIAGRAMS w/ VARIABLES



Putting it together

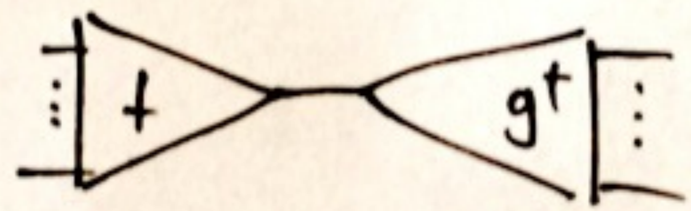
Identities are same in operad and cooperad

$$\frac{}{x : A \vdash x : A}$$



PUTTING IT TOGETHER

Composing like this

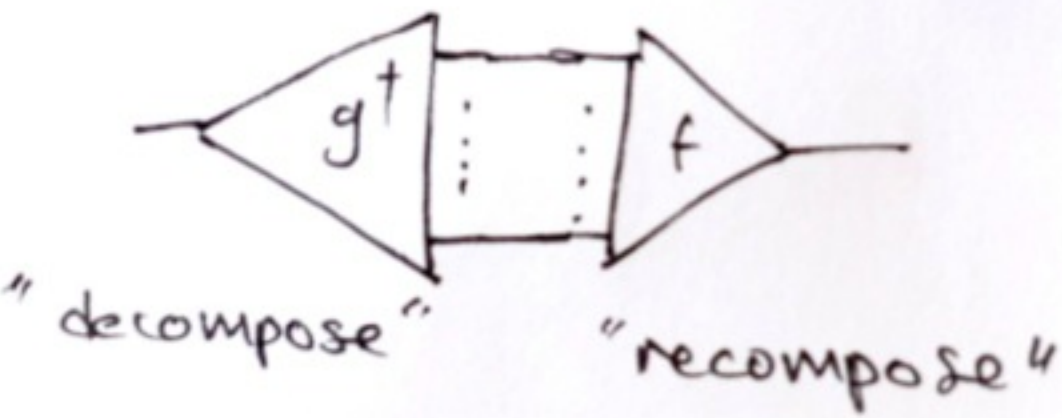


"construct" "deconstruct"

makes sense.

PUTTING IT TOGETHER

Composing like this



makes sense.

PUTTING IT TOGETHER

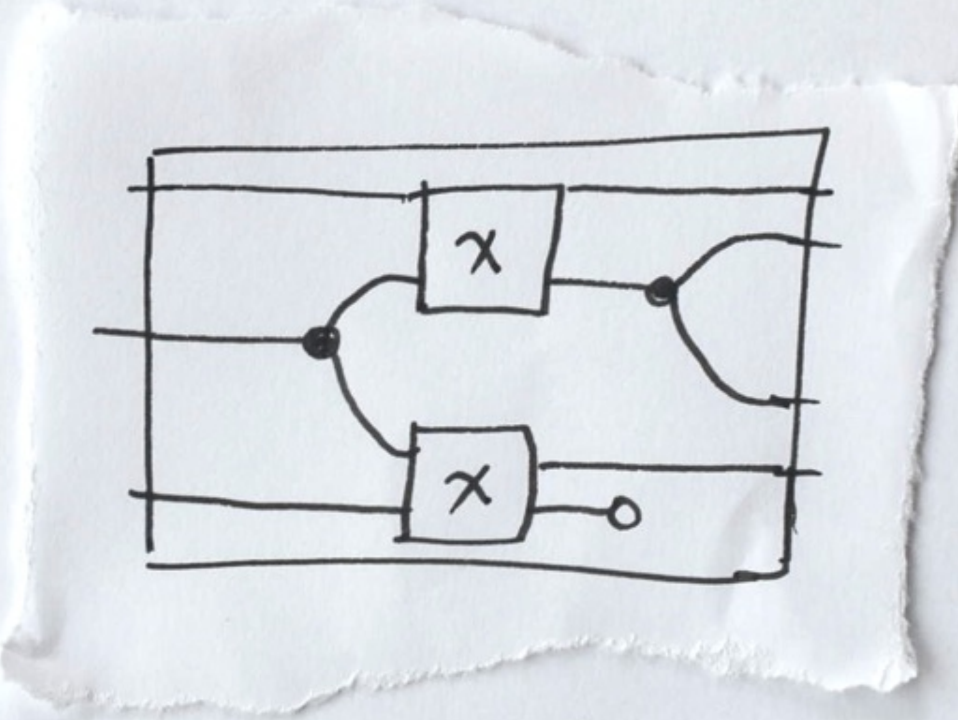
STRING DIAGRAMS WITH VARIABLES
FORM A { COMPUTAD / POLYCATEGORY }

OF MANY-TO-MANY, DIAGRAM TRANSFORMATIONS
PARTIAL.

(WITH THE MIX RULE)

4. Ditching Linearity

$$\frac{\Delta \vdash t : A,\ t' : A}{\Delta \vdash t'' : A}\ \text{Contraction}$$

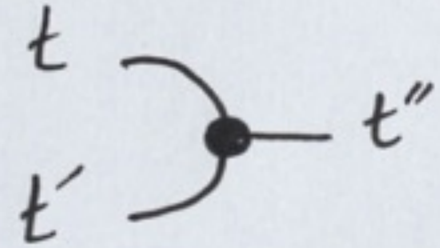


$$\frac{\Delta \vdash t : A}{\Delta \vdash t : A,\ t'' : B}\ \text{Weakening}$$

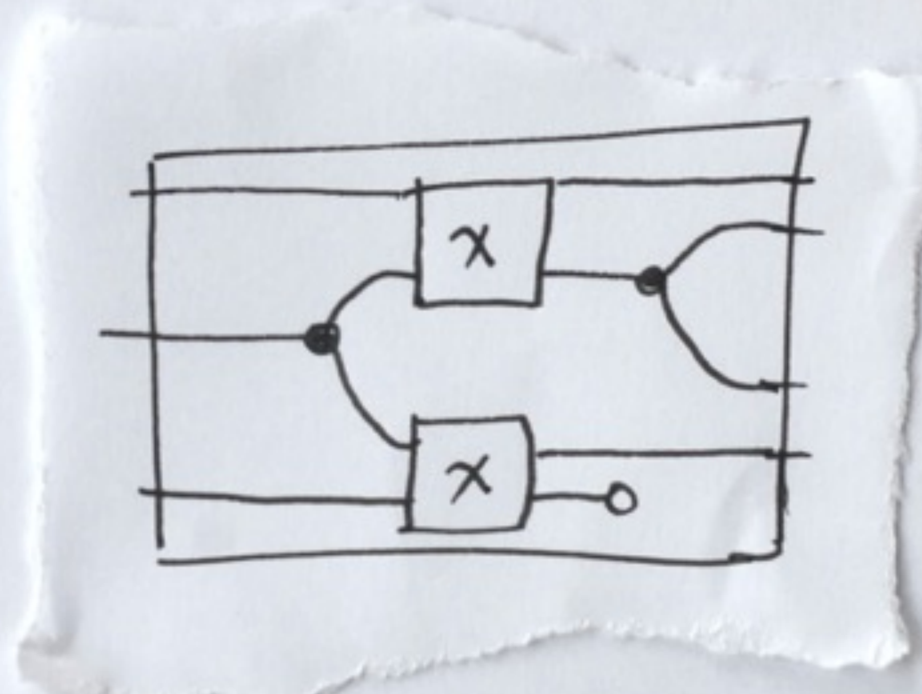
4. Ditching Linearity

$$\frac{\Delta \vdash t : A,\ t' : A}{\Delta \vdash t'' : A}\ \text{Contraction}$$

with $t = t' = t''$



$$\frac{\Delta \vdash t : A}{\Delta \vdash t : A,\ t'' : B}\ \text{Weakening}$$



4. Ditching Linearity

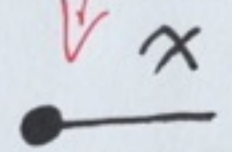
$$\frac{\Delta \vdash t : A,\ t' : A}{\Delta \vdash t'' : A}\ \text{Contraction}$$

$$\frac{\Delta \vdash t : A}{\Delta \vdash t : A,\ t'' : B}\ \text{Weakening}$$

with $\cancel{t = t' = t''}$

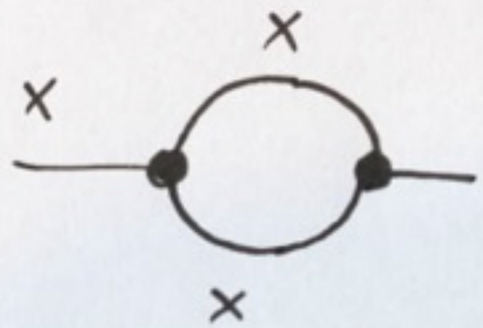
$t'' = \text{M.C.U.}(t, t')$

Commutative Monoid!*



where x is a fresh variable.

4. DITCHING LINEARITY.

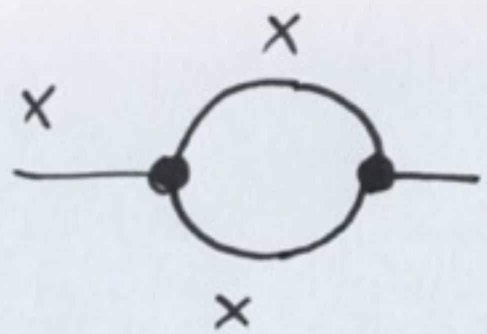
SPECIAL! 

MGU(x, x) = x

= 

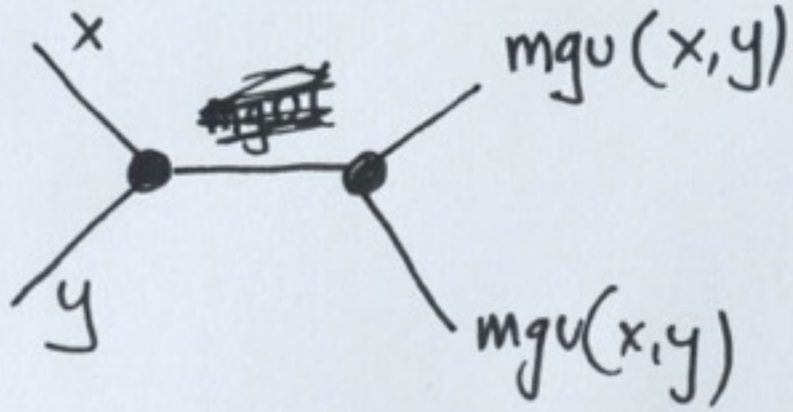
4. DITCHING LINEARITY.

SPECIAL!

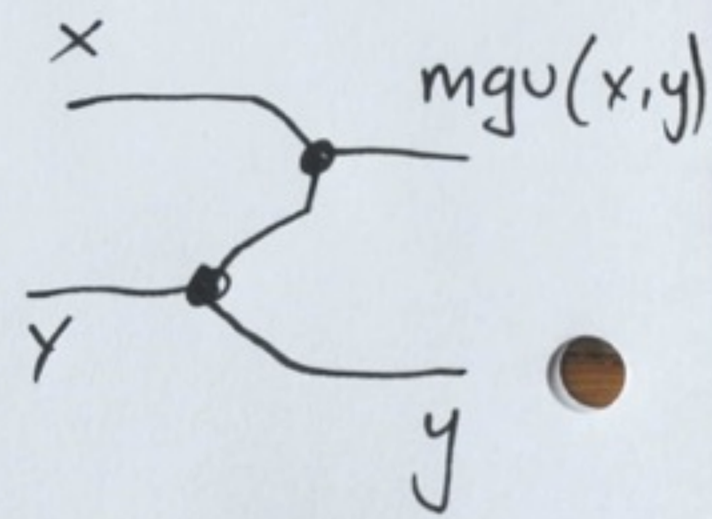


MGU(x, x) = x

=



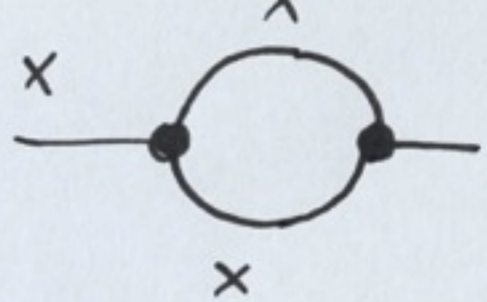
$\neq$



NOT FROB :(

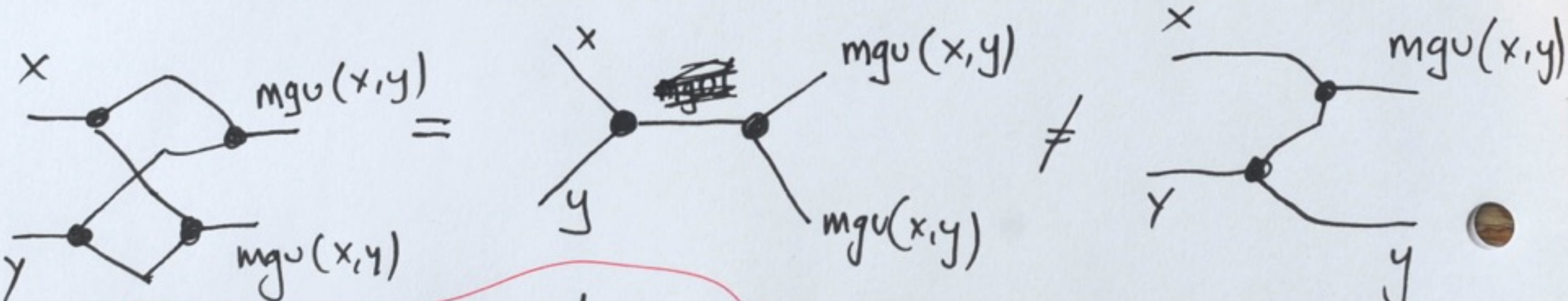
4. Ditching Linearity.

SPECIAL!



$\mathrm{MGU}(x, x) = x$

=



IS BIALGEBRA! :)

NOT FROB :(

Open Problems

- How to compute MGU for two diagrams?
    - Trickier than expected because the category does not ave many push-outs!
- Cut-elimination for the whole computad?
- Can we we express the separation condition for combinatorial planar graphs?


THANKS!