Semantics of Higher-Order Quantum Computation via Geometry of Interaction

In: Proc. Logic in Computer Science (LICS), June 2011

Ichiro Hasuo
University of Tokyo (JP)

Naohiko Hoshino
RIMS, Kyoto University (JP)

東京大学
THE UNIVERSITY OF TOKYO

京都大学
KYOTO UNIVERSITY

Contribution

Denotational semantics of a **functional quantum programming language**

Contribution

Denotational semantics of a **functional quantum programming language**

Linear λ-calculus + quantum primitives

Contribution

One of the first to cover the full features!

- !-modality for duplicable data
- recursion

Denotational semantics of a **functional quantum programming language**

Linear λ-calculus + quantum primitives

Contribution

One of the first to cover the full features!

- !-modality for duplicable data
- recursion

Adequacy

Denotational semantics of a **functional quantum programming language**

Linear λ-calculus + quantum primitives

Contribution

One of the first to cover the full features!

- !-modality for duplicable data
- recursion

Adequacy

Denotational semantics of a **functional quantum programming language**

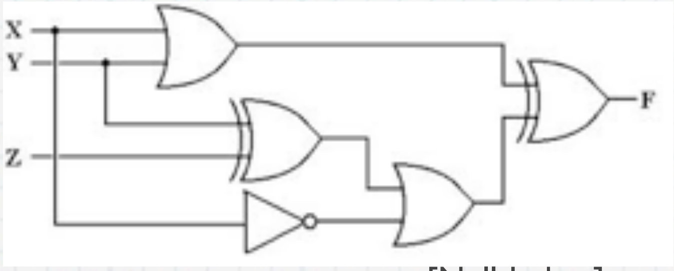
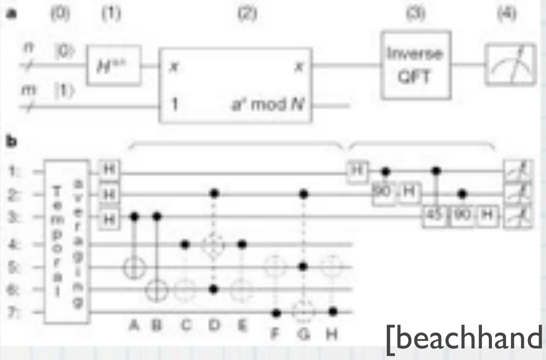
Linear λ-calculus + quantum primitives

... via **GoI** (Geometry of Interaction)

Part 1

Functional QPL: Some Contexts

Quantum Programming Language

Classical		Quantum	
(Boolean) circuit	 [Null-Lobur]	Quantum circuit	 [beachhandball.es]
Programming language	see code below		

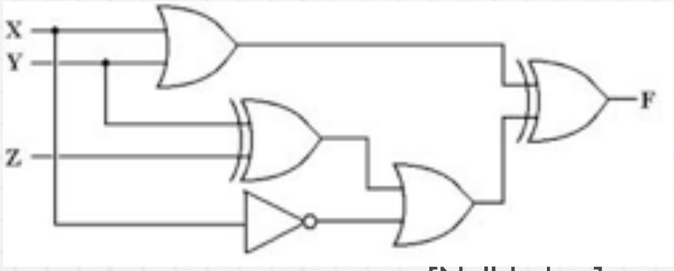
```
int i,j;
int factorial(int k)
{
    j=1;
    for (i=1; i<=k; i++)
        j=j*i;
    return j;
}
```

Hasuo (Tokyo)

Quantum Programming Language

Classical

(Boolean) circuit



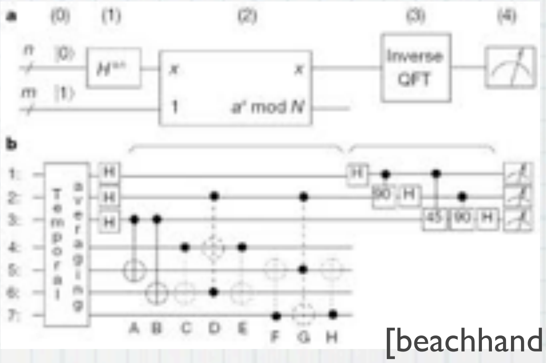
[Null-Lobur]

Programming language

```
int i,j;
int factorial(int k)
{
    j=1;
    for (i=1; i<=k; i++)
        j=j*i;
    return j;
}
```

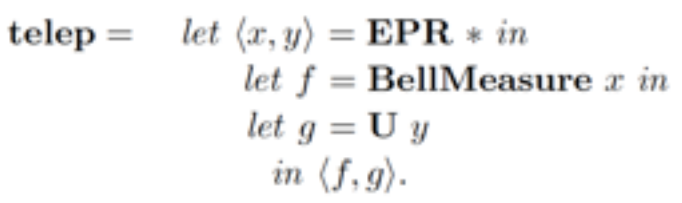
Quantum

Quantum circuit



[beachhandball.es]

Quantum programming language



$$\begin{aligned}\mathbf{telep} = \quad & \mathit{let}\ \langle x, y\rangle = \mathbf{EPR} * \mathit{in} \\ & \mathit{let}\ f = \mathbf{BellMeasure}\ x\ \mathit{in} \\ & \mathit{let}\ g = \mathbf{U}\ y \\ & \mathit{in}\ \langle f, g\rangle.\end{aligned}$$

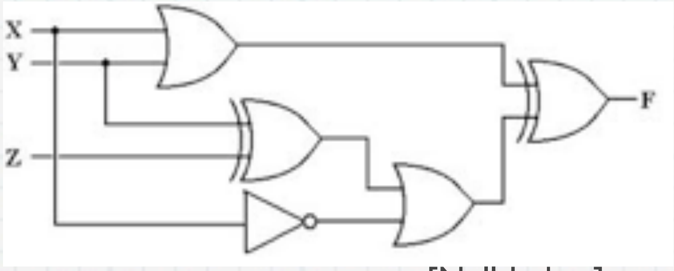
[Selinger-Valiron]

Hasuo (Tokyo)

Quantum Programming Language

Classical

(Boolean) circuit



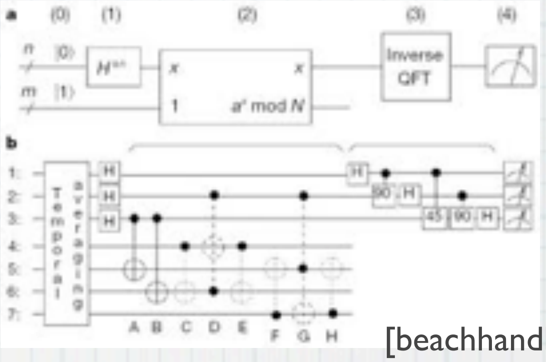
[Null-Lobur]

Programming language

```
int i,j;
int factorial(int k)
{
    j=1;
    for (i=1; i<=k; i++)
        j=j*i;
    return j;
}
```

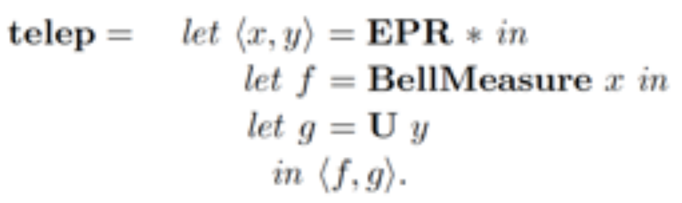
Quantum

Quantum circuit



[beachhandball.es]

Quantum programming language



$$
\begin{aligned}
\textbf{telep} = \quad & \mathit{let}\ \langle x, y\rangle = \mathbf{EPR} * \mathit{in} \\
& \quad \mathit{let}\ f = \mathbf{BellMeasure}\ x\ \mathit{in} \\
& \quad \mathit{let}\ g = \mathbf{U}\ y \\
& \quad \mathit{in}\ \langle f, g\rangle.
\end{aligned}
$$

[Selinger-Valiron]

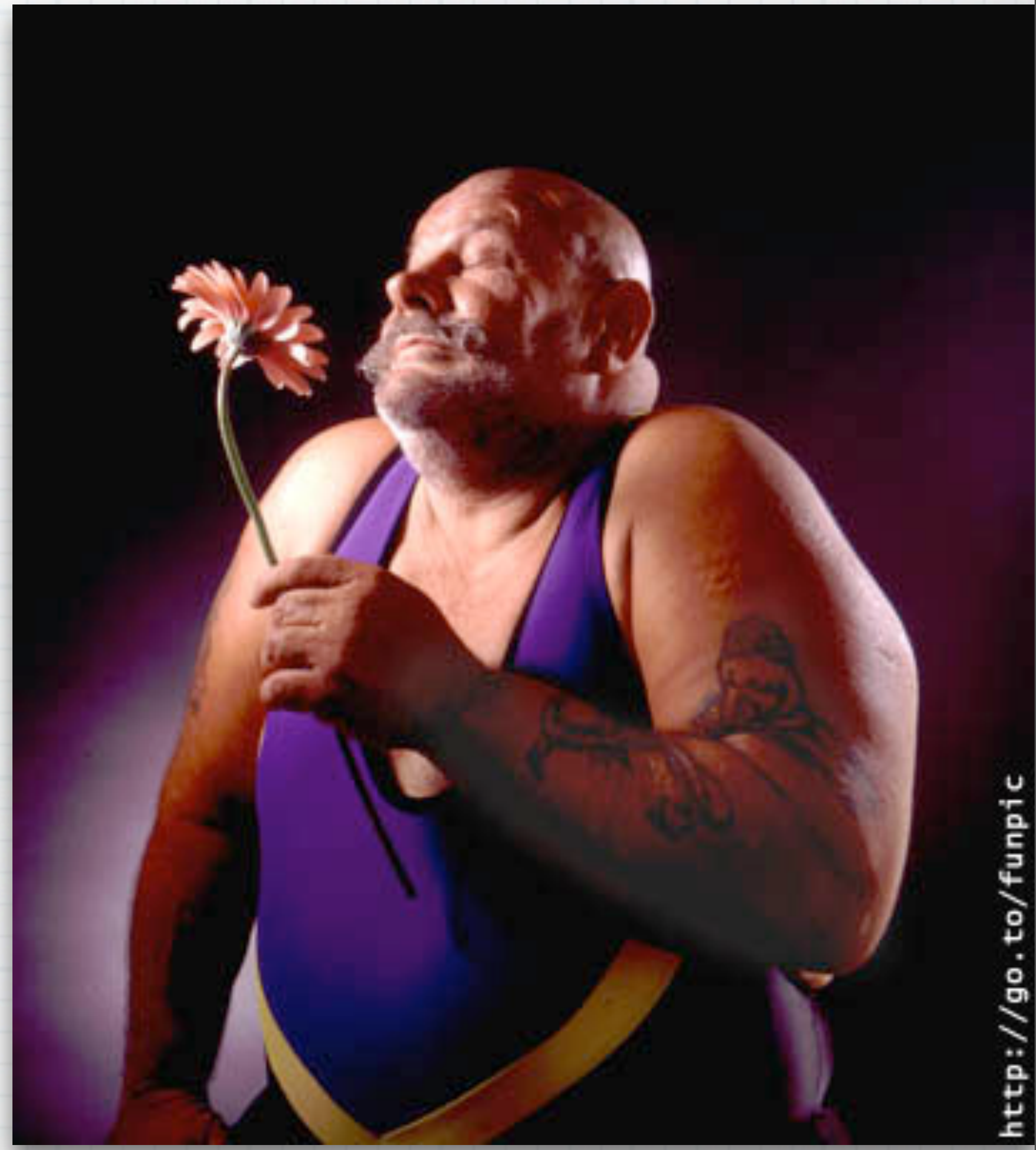
- For discovery of **algorithms**
- For **reasoning, verification**

Functional Quantum Programming Language

Hasuo (Tokyo)

Functional Quantum Programming Language

- A **real man's** programming style



Hasuo (Tokyo)

Functional Quantum Programming Language

- A **real man's** programming style
- Heavily used in the **financial sector**
- ...

ICFP'11 Sponsers (Tokyo, Sep 2011)

CiTRIX
CREDIT SUISSE
Quantitative Strategies
Erlang
SOLUTIONS
JANE STREET
galois
IIJ
Internet Initiative Japan
Microsoft Research
NII
National Institute of Informatics
Standard Chartered
TSURU CAPITAL
twitter

Functional Quantum Programming Language

- A **real man's** programming style
- Heavily used in the **financial sector**
- ...

- **Mathematically nice and clean**
  - Aids semantical study
  - **Transfer** from classical to quantum

ICFP'11 Sponsers (Tokyo, Sep 2011)

CITRIX
CREDIT SUISSE
Quantitative Strategies
Erlang SOLUTIONS
JANE STREET
galois
IIJ
Internet Initiative Japan
Microsoft Research
NII
National Institute of Informatics
Standard Chartered
TSURU CAPITAL
twitter

Functional QPL: Syntax

- **Linear λ-calculus**
  + quantum primitives [van Tonder, Selinger, Valiron, ...]
- Linearity for **no-cloning**
  - "Input can be used only once"
  - Not allowed/typable:
  - Duplicable (classical) data: by the **!-modality**

Functional QPL: Syntax

- **Linear λ-calculus**

  + quantum primitives

Preparation/Unitary transformation/Measurement

[van Tonder, Selinger, Valiron, ...]

- Linearity for **no-cloning**
  - "Input can be used only once"
  - Not allowed/typable:
  - Duplicable (classical) data: by the **!-modality**

Functional QPL: Syntax

- **Linear λ-calculus**

  + quantum primitives

Preparation/Unitary transformation/Measurement

[van Tonder, Selinger, Valiron, ...]

- Linearity for **no-cloning**
  - "Input can be used only once"
  - Not allowed/typable:

$$\lambda x.\ \langle \mathtt{meas}\, x,\ \mathtt{meas}\, x \rangle$$

Functional QPL: Syntax

- Linear λ-calculus

  + quantum primitives

Preparation/Unitary transformation/Measurement

[van Tonder, Selinger, Valiron, ...]

- Linearity for **no-cloning**
  - "Input can be used only once"
  - Not allowed/typable:

$$\boxed{\lambda x.\ \langle \mathtt{meas}\ x,\ \mathtt{meas}\ x\rangle}$$

Functional QPL: Syntax

- **Linear λ-calculus**

  + quantum primitives

Preparation/Unitary transformation/Measurement

[van Tonder, Selinger, Valiron, ...]

- Linearity for **no-cloning**
  - "Input can be used only once"
  - Not allowed/typable: $\lambda x.\ \langle \mathtt{meas}\, x,\ \mathtt{meas}\, x \rangle$
  - Duplicable (classical) data: by the **!-modality**

$$\vdash \mathtt{tt} : {!}\,\mathtt{bit}$$

Functional QPL: Syntax

- **Linear λ-calculus**

  + quantum primitives

Preparation/Unitary transformation/Measurement

[van Tonder, Selinger, Valiron, ...]

- Linearity for **no-cloning**
  - “Input can be used only once”
  - Not allowed/typable: $\lambda x.\ \langle \mathtt{meas}\ \overset{\searrow}{x},\ \mathtt{meas}\ \overset{\searrow}{x} \rangle$
  - Duplicable (classical) data: by the **!-modality**

$$\vdash \mathtt{tt} : {!}\,\mathtt{bit}$$

“arbitrary many copies”

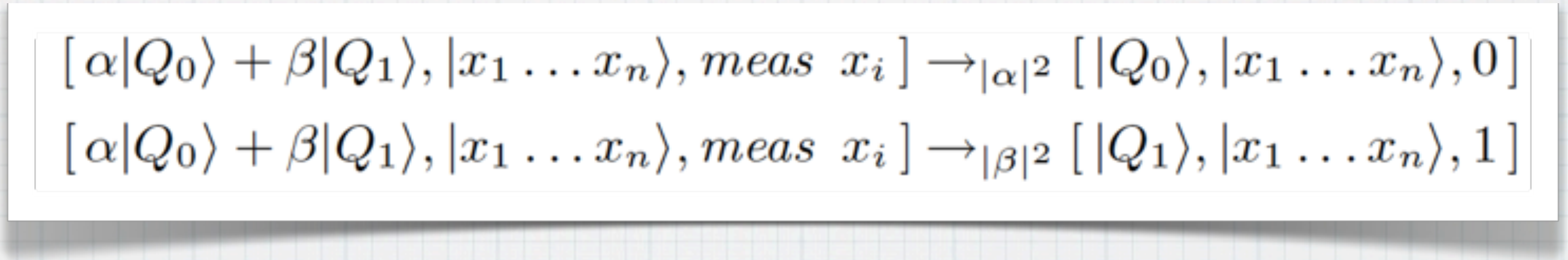
Functional QPL: Semantics

Functional QPL: Semantics

- Semantics = **mathematical model**

Functional QPL: Semantics

- Semantics = **mathematical model**
- Operational semantics: [Selinger & Valiron, '09]
  - "Quantum closure," reduction with probabilistic branching



$$[\,\alpha|Q_0\rangle + \beta|Q_1\rangle, |x_1 \dots x_n\rangle, meas\ \ x_i\,] \rightarrow_{|\alpha|^2} [\,|Q_0\rangle, |x_1 \dots x_n\rangle, 0\,]$$

$$[\,\alpha|Q_0\rangle + \beta|Q_1\rangle, |x_1 \dots x_n\rangle, meas\ \ x_i\,] \rightarrow_{|\beta|^2} [\,|Q_1\rangle, |x_1 \dots x_n\rangle, 1\,]$$

  - Allows to identify **linear logic** $\otimes$ and **quantum** $\otimes$ (feature of the Selinger-Valiron language; not in ours)

Functional QPL: Semantics

$[\![M]\!]$

Functional QPL: Semantics

- Denotational semantics
  - $[\![M]\!]$: a **function**, or an **arrow** of a category

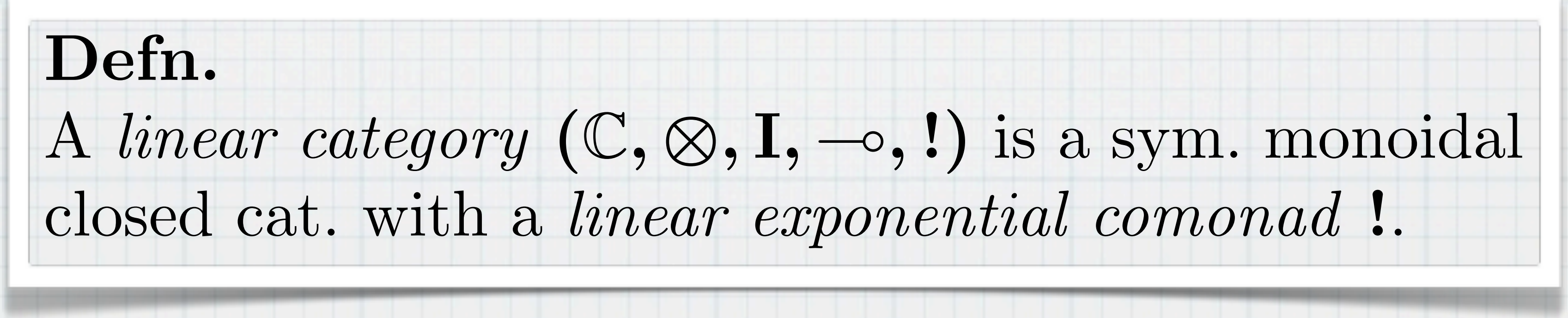
Functional QPL: Semantics

- Denotational semantics
  - $[\![M]\!]$: a **function**, or an **arrow** of a category
  - **Compositionality:** $[\![MN]\!] = [\![M]\!] \circ [\![N]\!]$

Hasuo (Tokyo)

Functional QPL: Semantics

- Denotational semantics
  - $[\![M]\!]$: a **function**, or an **arrow** of a category
  - **Compositionality:** $[\![MN]\!] = [\![M]\!] \circ [\![N]\!]$
  - **Linear category:** [Benton & Wadler, Bierman]
    (axioms for) a categorical model of linear λ-calculus



Defn.
A *linear category* $(\mathbb{C}, \otimes, \mathbf{I}, \multimap, \mathbf{!})$ is a sym. monoidal closed cat. with a *linear exponential comonad* $\mathbf{!}$.

Functional QPL: Semantics

- Denotational semantics
  - $[\![M]\!]$: a **function**, or an **arrow** of a category
  - **Compositionality**: $[\![MN]\!] = [\![M]\!] \circ [\![N]\!]$
  - **Linear category**: [Benton & Wadler, Bierman]
    (axioms for) a categorical model of linear λ-calculus

    > **Defn.**
    > A *linear category* $(\mathbb{C}, \otimes, \mathbf{I}, \multimap, !)$ is a sym. monoidal closed cat. with a *linear exponential comonad* $!$.

  - For functional QPL? Is **Hilb** (or alike) a linear cat.?

Functional QPL: Semantics

- **Hilb** (or alike) is **not** a linear category

- **Challenge**: coexistence of **quantum** and **classical** data
- Only partial results
  - [Selinger & Valiron, '08]: for strictly linear fragment (w/o !)

Functional QPL: Semantics

- **Hilb** (or alike) is **not** a linear category

monoidal closed str. $(\mathbb{C}, \otimes, \mathbf{I}, \multimap)$

! (for duplicable data)

- **Challenge**: coexistence of **quantum** and **classical** data
- Only partial results
  - [Selinger & Valiron, '08]: for strictly linear fragmant (w/o !)

Functional QPL: Semantics

- **Hilb** (or alike) is **not** a linear category

monoidal closed str. $(\mathbb{C}, \otimes, \mathbf{I}, \multimap)$

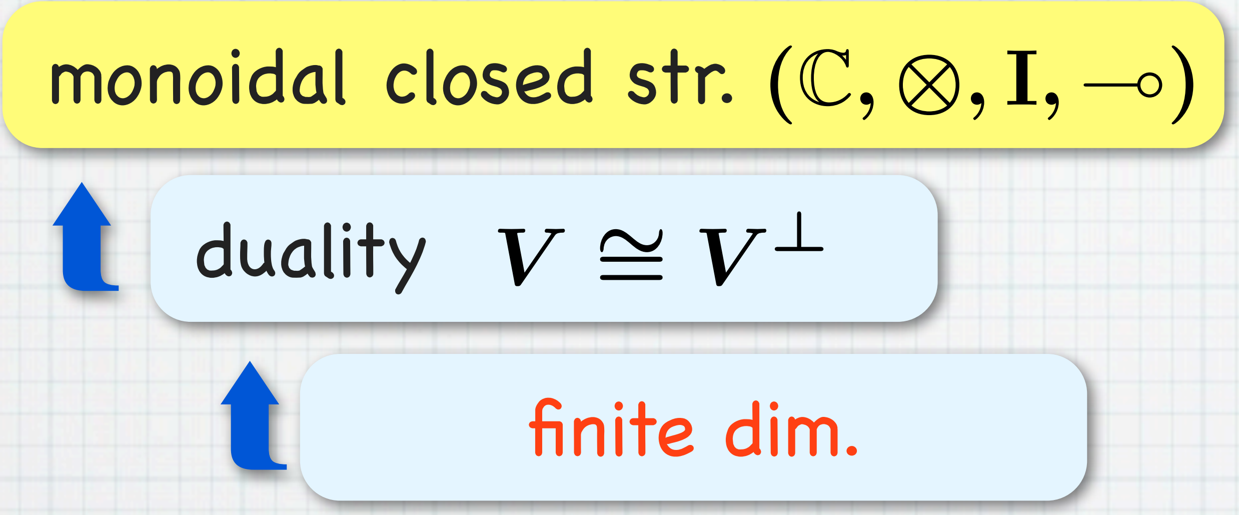
! (for duplicable data)

↑ duality $V \cong V^{\perp}$

- __**Challenge**__: coexistence of **quantum** and **classical** data
- Only partial results
  - [Selinger & Valiron, ’08]:
    for strictly linear fragmant (w/o **!**)

Functional QPL: Semantics

- **Hilb** (or alike) is **not** a linear category



monoidal closed str. $(\mathbb{C}, \otimes, \mathbf{I}, \multimap)$

duality $V \cong V^{\perp}$

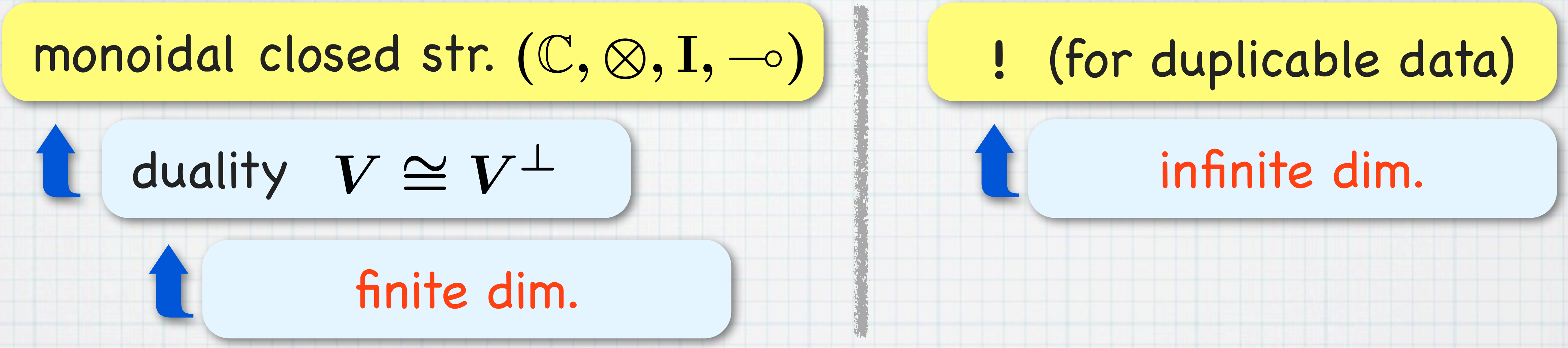
finite dim.

! (for duplicable data)

- **Challenge**: coexistence of **quantum** and **classical** data
- Only partial results
  - [Selinger & Valiron, '08]:
    for strictly linear fragmant (w/o **!**)

Functional QPL: Semantics

- **Hilb** (or alike) is **not** a linear category



monoidal closed str. $(\mathbb{C}, \otimes, \mathbf{I}, \multimap)$

duality $V \cong V^{\perp}$

finite dim.

! (for duplicable data)

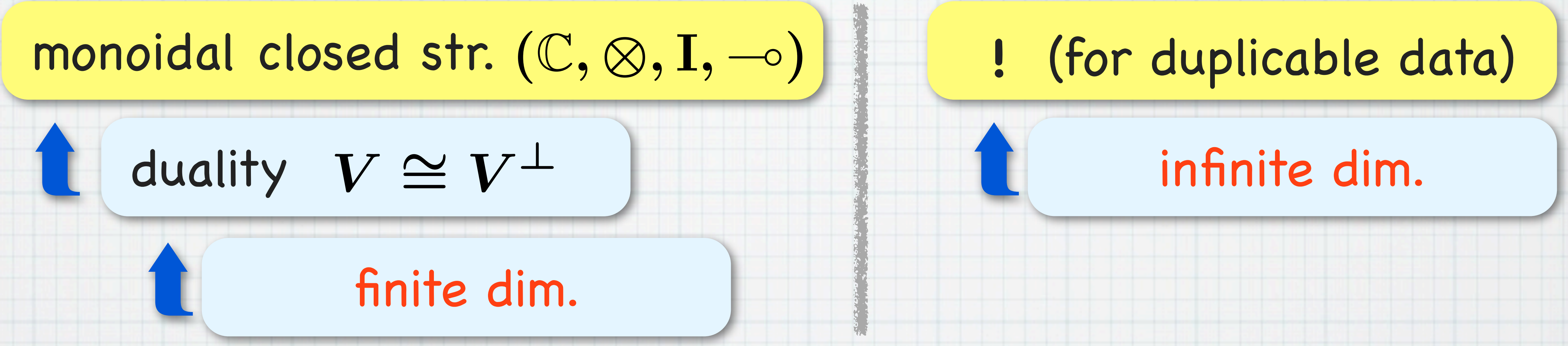
infinite dim.

- <u>**Challenge**</u>: coexistence of **quantum** and **classical** data
- Only partial results
  - [Selinger & Valiron, '08]:

    for strictly linear fragment (w/o !)

Functional QPL: Semantics

- **Hilb** (or alike) is **not** a linear category



monoidal closed str. $(\mathbb{C}, \otimes, \mathbf{I}, \multimap)$

duality $V \cong V^{\perp}$

finite dim.

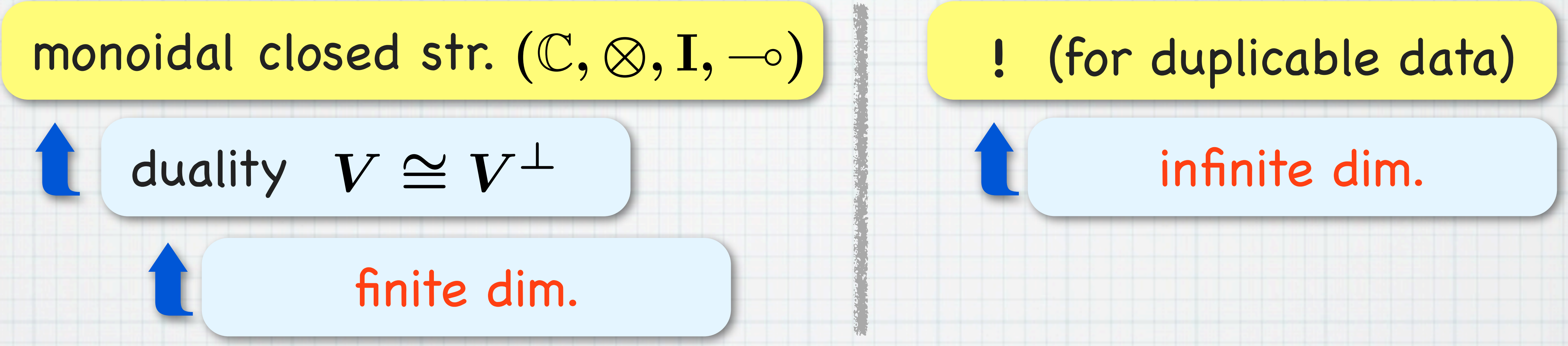
! (for duplicable data)

infinite dim.

- **Challenge**: coexistence of **quantum** and **classical** data

Functional QPL: Semantics

- **Hilb** (or alike) is **not** a linear category



monoidal closed str. $(\mathbb{C}, \otimes, \mathbf{I}, \multimap)$

duality $V \cong V^{\perp}$

finite dim.

! (for duplicable data)

infinite dim.

- **Challenge**: coexistence of **quantum** and **classical** data

- Only partial results

  - [Selinger & Valiron, '08]:

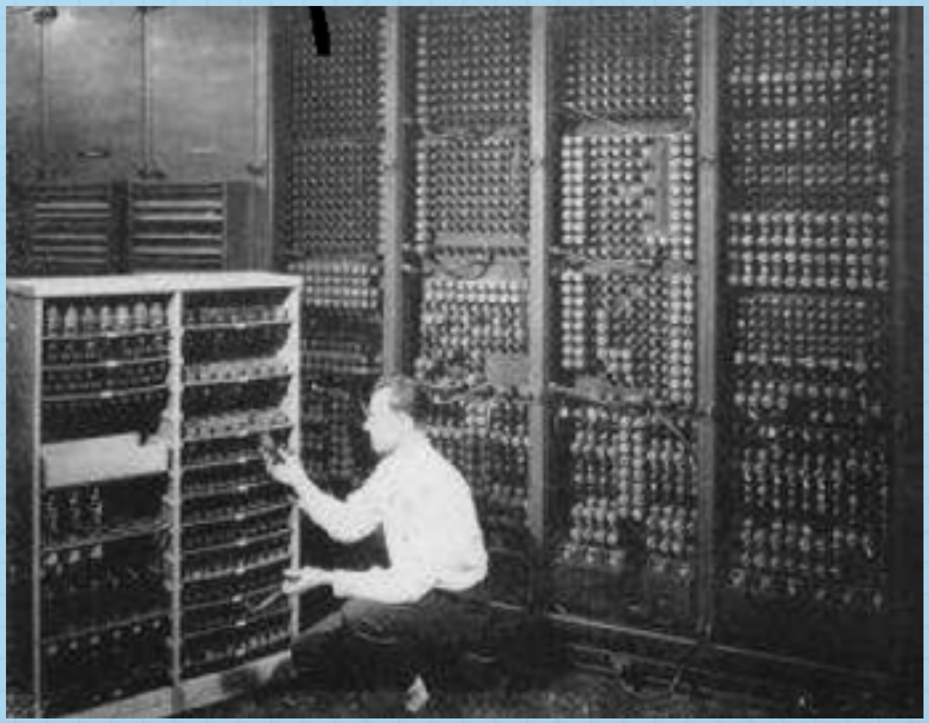
    for strictly linear fragmant (w/o **!**)

“Quantum Data, Classical Control”

Illustration by N. Hoshino

Quantum data

Classical control



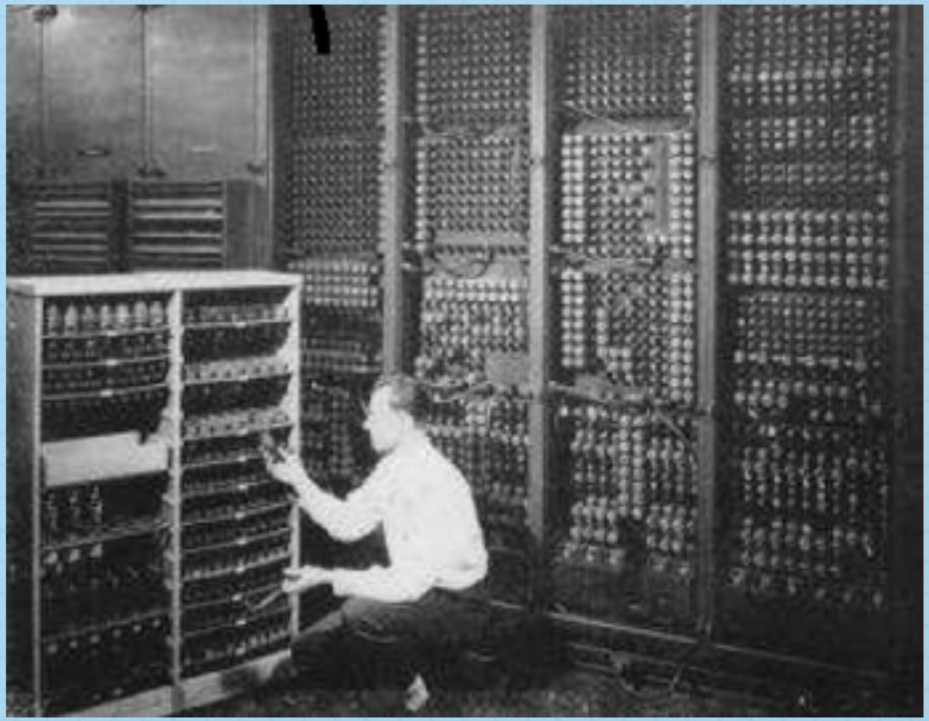
"Quantum Data, Classical Control"

Illustration by N. Hoshino

Quantum data

$$\frac{1}{\sqrt{2}}$$

Classical control



“Quantum Data, Classical Control”

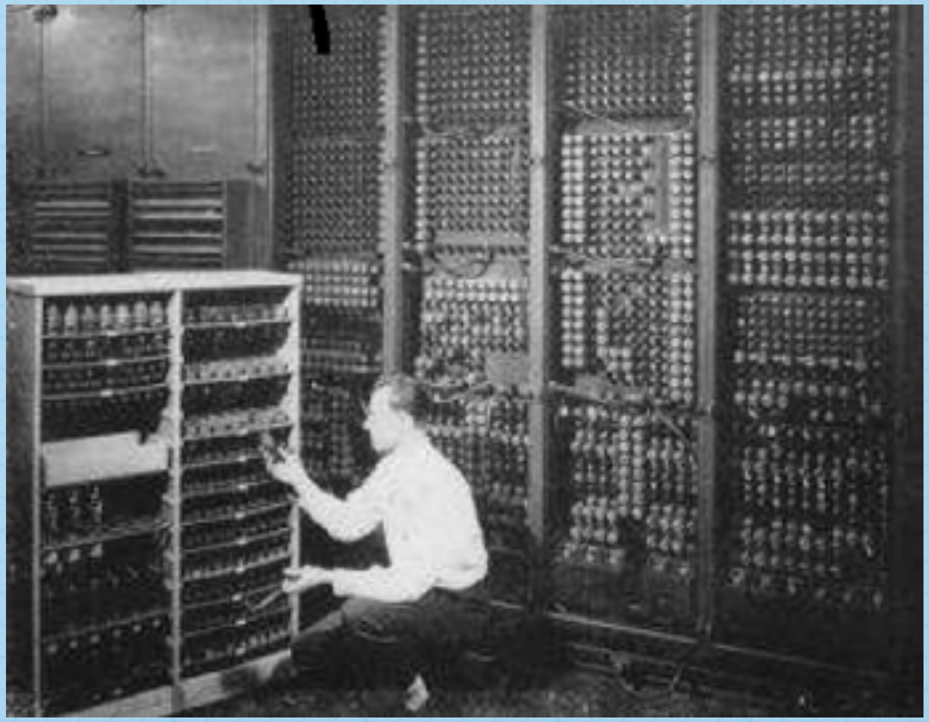
Illustration by N. Hoshino

Quantum data

$$\frac{1}{\sqrt{2}}$$

$$+\frac{1}{\sqrt{2}}$$

Classical control



What We Do

- **GoI (Geometry of Interaction)** [Girard '89]

  An "implementation" of **classical control**

$$\mathbf{tr}(f) =$$

$$f_{XY} \sqcup \left(\coprod_{n \in \mathbb{N}} f_{ZY} \circ (f_{ZZ})^{n} \circ f_{XZ} \right)$$

What We Do

- **GoI (Geometry of Interaction)** [Girard '89]

  An "implementation" of **classical control**

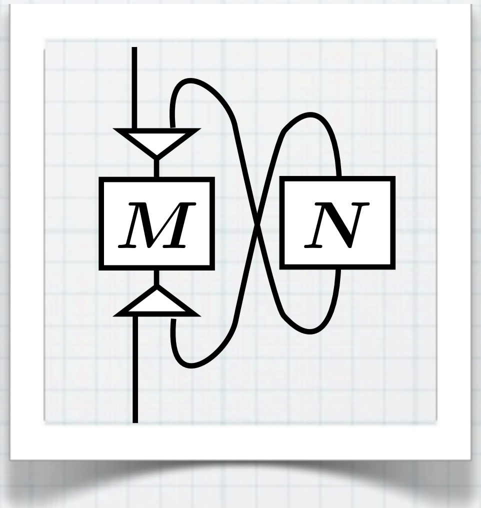
$$\mathbf{tr}(f) = f_{XY} \sqcup \left(\coprod_{n\in\mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \right)$$

What We Do

- **GoI (Geometry of Interaction)** [Girard '89]
  An "implementation" of **classical control**

$$\mathbf{tr}(f) = f_{XY} \sqcup \left(\coprod_{n \in \mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \right)$$

- **Categorical GoI** [Abramsky, Haghverdi, Scott '02]
  Its categorical axiomatics



What We Do

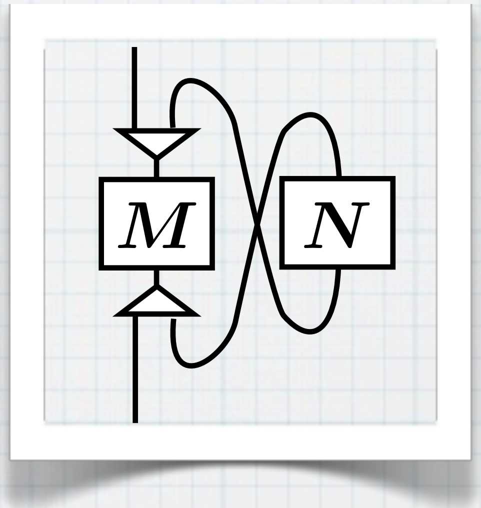
- **GoI (Geometry of Interaction)** [Girard '89]

  An "implementation" of **classical control**

$$\mathbf{tr}(f) = f_{XY} \sqcup \left(\coprod_{n \in \mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \right)$$

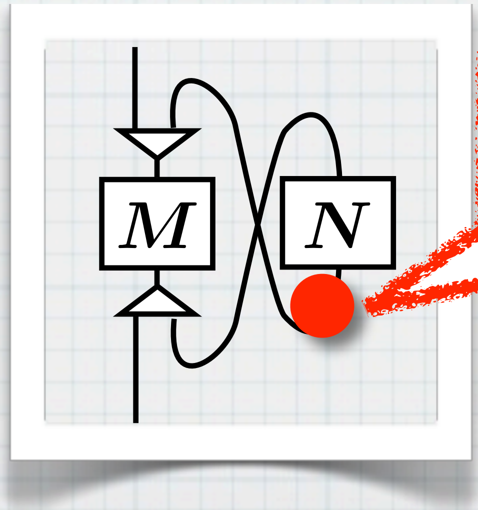
- **Categorical GoI** [Abramsky, Haghverdi, Scott '02]

  Its categorical axiomatics



- We add a **quantum layer** to GoI
  - ➜ "Quantum data, classical control"
  - Used: theory of coalgebra

    [Hasuo, Jacobs, Sokolova '07] [Jacobs '10]



quantum state

Part 2

The Categorical GoI Workflow

GoI:
Geometry of Interaction

- J.-Y. Girard, at Logic Colloquium '88

GoI:
Geometry of Interaction

- J.-Y. Girard, at Logic Colloquium '88

GoI:
Geometry of Interaction

- J.-Y. Girard, at Logic Colloquium '88

- **<u>But</u>** I'm no linear logician!

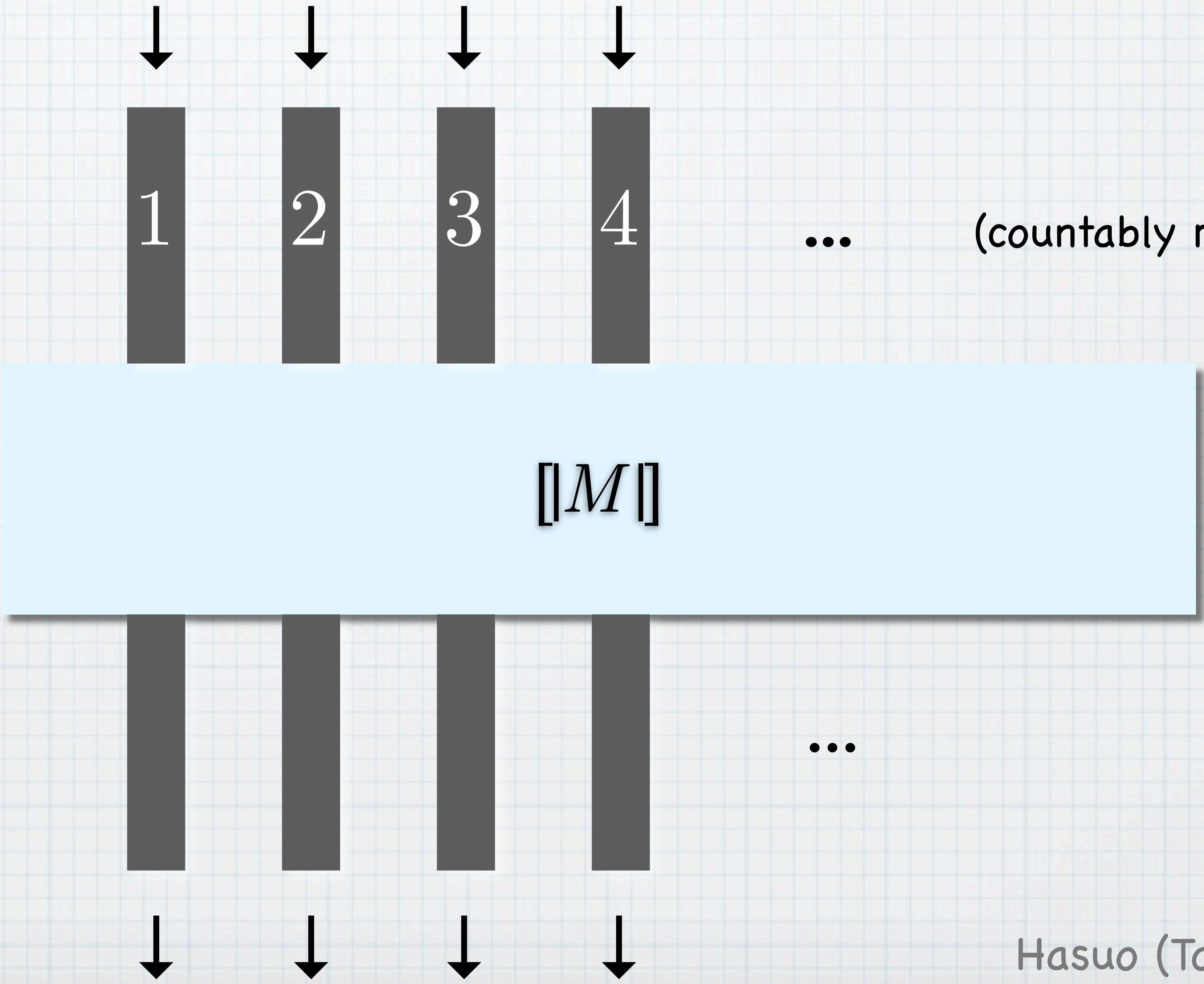
GoI:
Geometry of Interaction

- J.-Y. Girard, at Logic Colloquium '88
- **But** I'm no linear logician!
- In this talk:
  - Its categorical formulation
    [Abramsky, Haghverdi, Scott '02]
  - "The GoI Animation"

The GoI Animation

$$[\![\boldsymbol{M}]\!] \;=\; (\,\mathbb{N} \rightharpoonup \mathbb{N},\ \text{a partial function}\,)$$

$$=\ \text{“piping”}$$

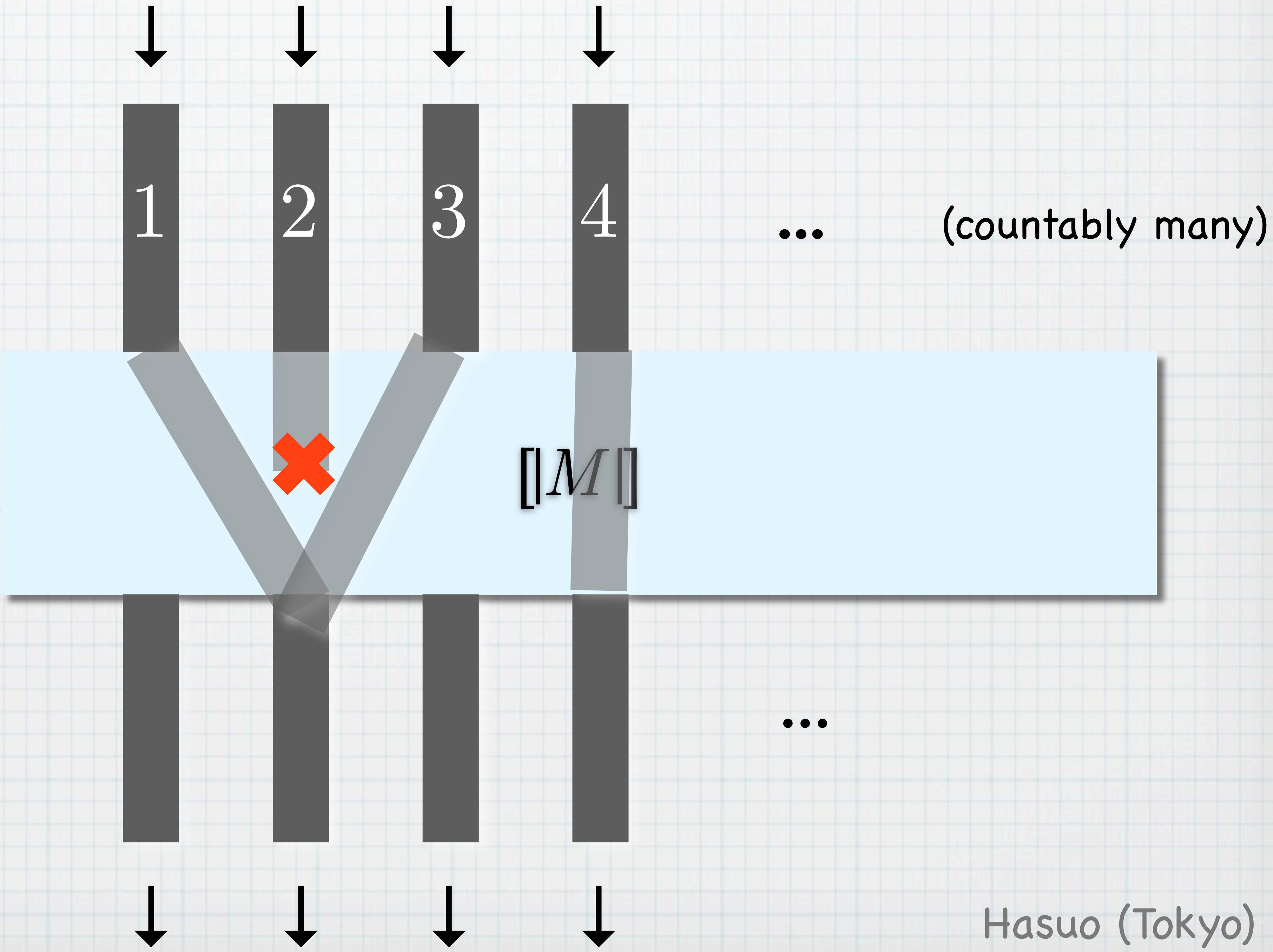


(countably many)

The GoI Animation

$$[\![M]\!] = (\,\mathbb{N} \rightharpoonup \mathbb{N},\ \text{a partial function}\,)$$

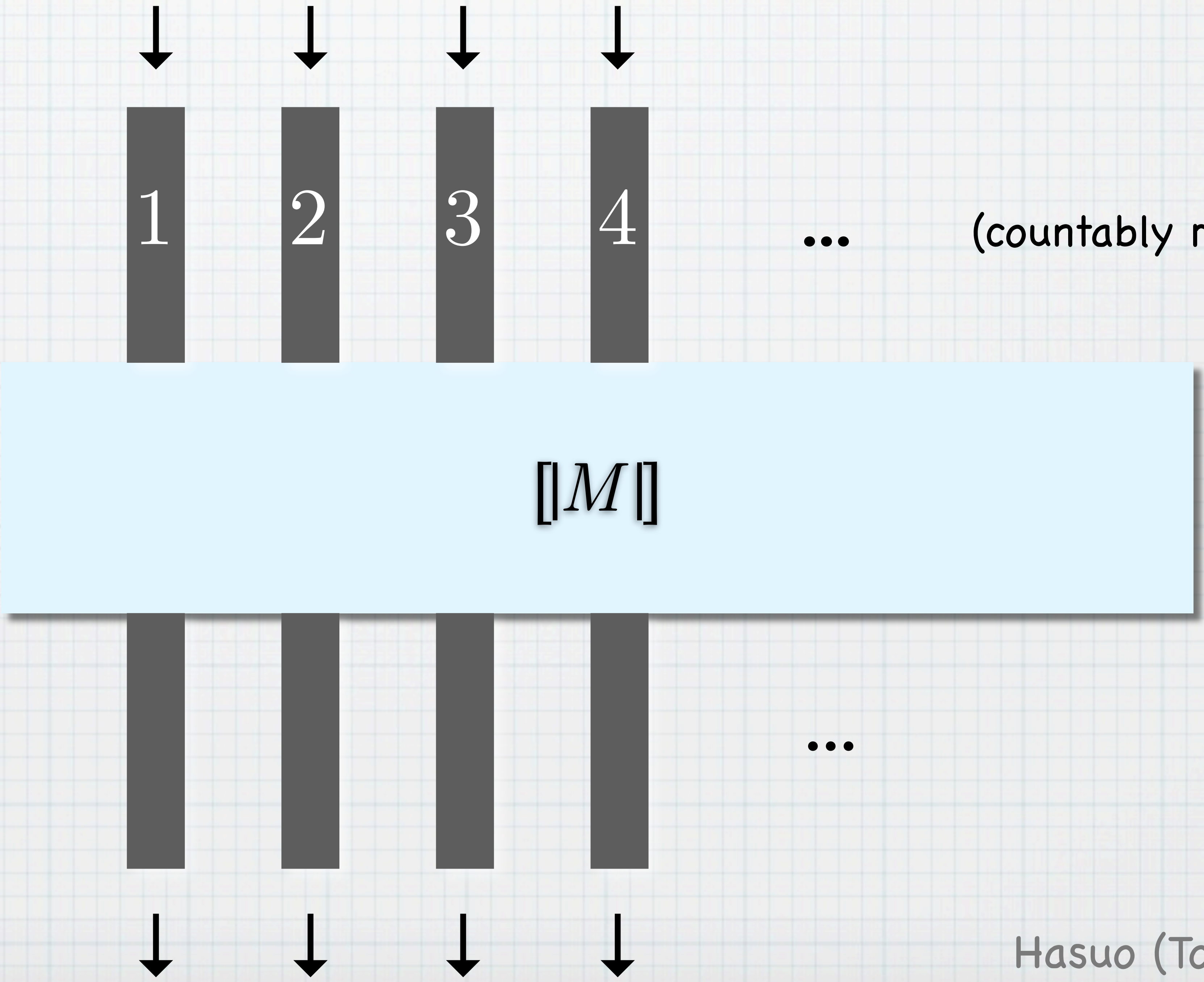
$= \text{“piping”}$



The GoI Animation

$$[\![M]\!] \;=\; (\, \mathbb{N} \rightharpoonup \mathbb{N},\ \text{a partial function}\,)$$

$$=\ \text{“piping”}$$

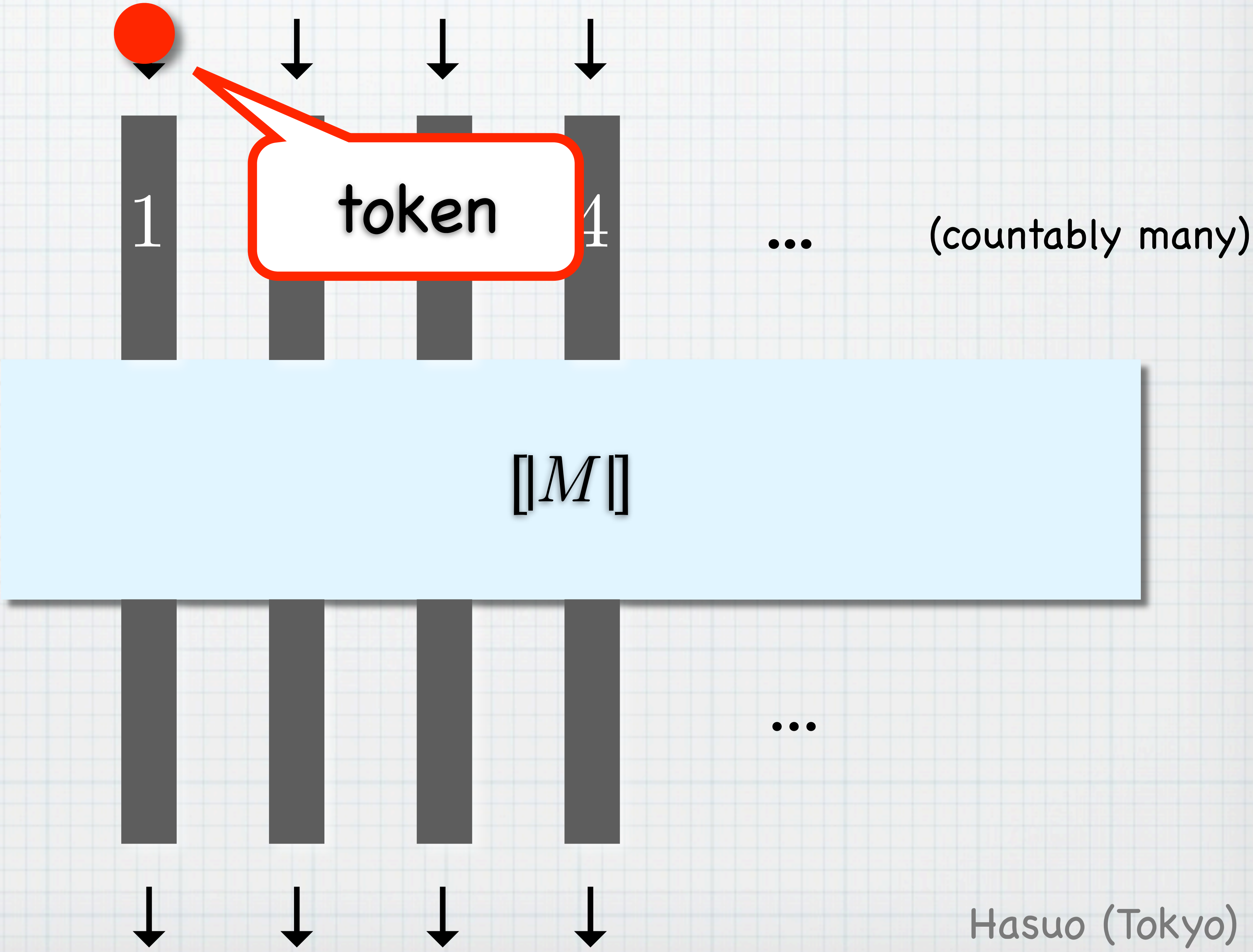


(countably many)

The GoI Animation

$[\![M]\!] \;=\; (\,\mathbb{N} \rightharpoonup \mathbb{N},\ \text{a partial function}\,)$

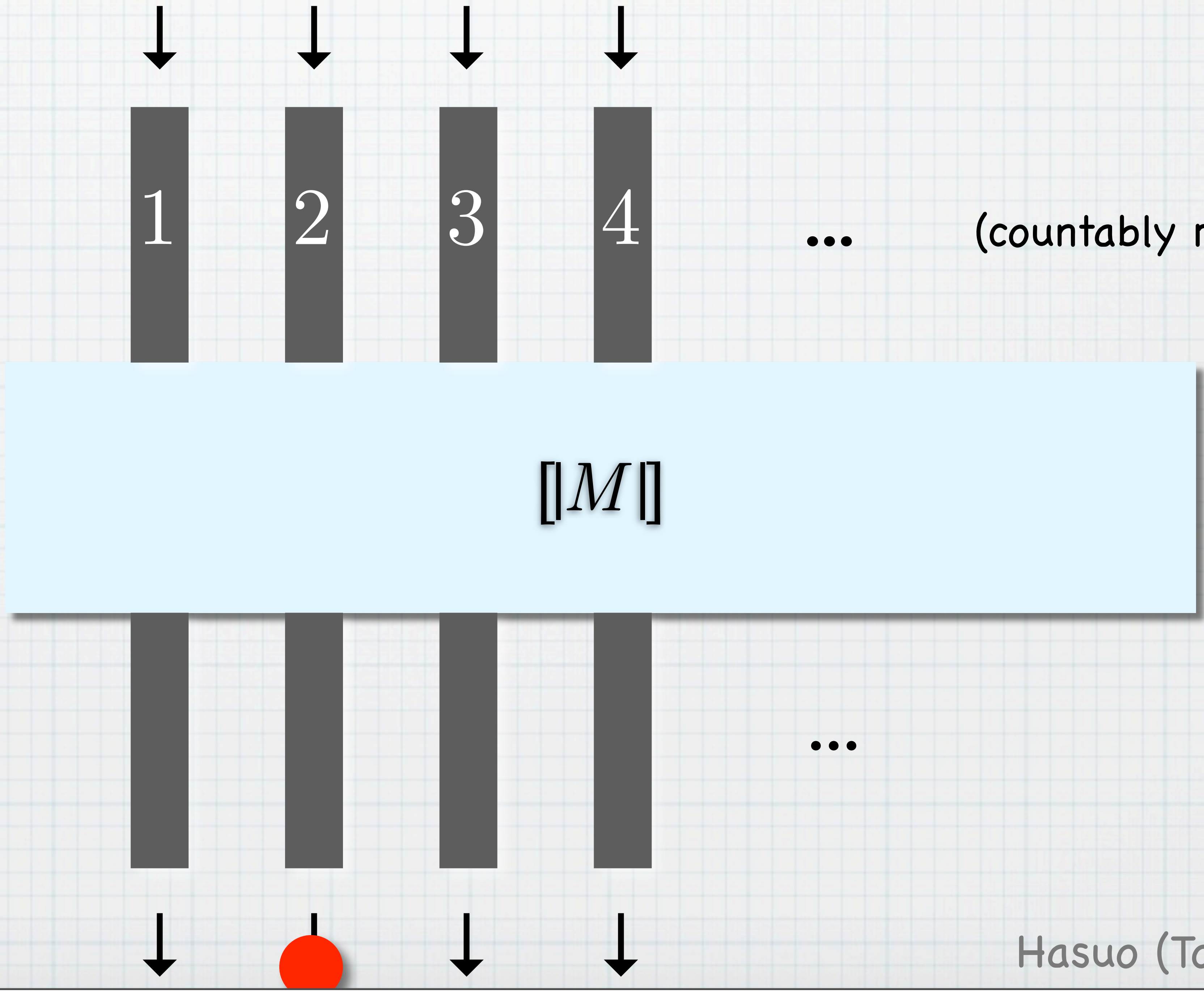
$= \;$ "piping"



The GoI Animation

$$[\![M]\!] = (\,\mathbb{N} \rightharpoonup \mathbb{N},\ \text{a partial function}\,)$$

$$= \text{“piping”}$$

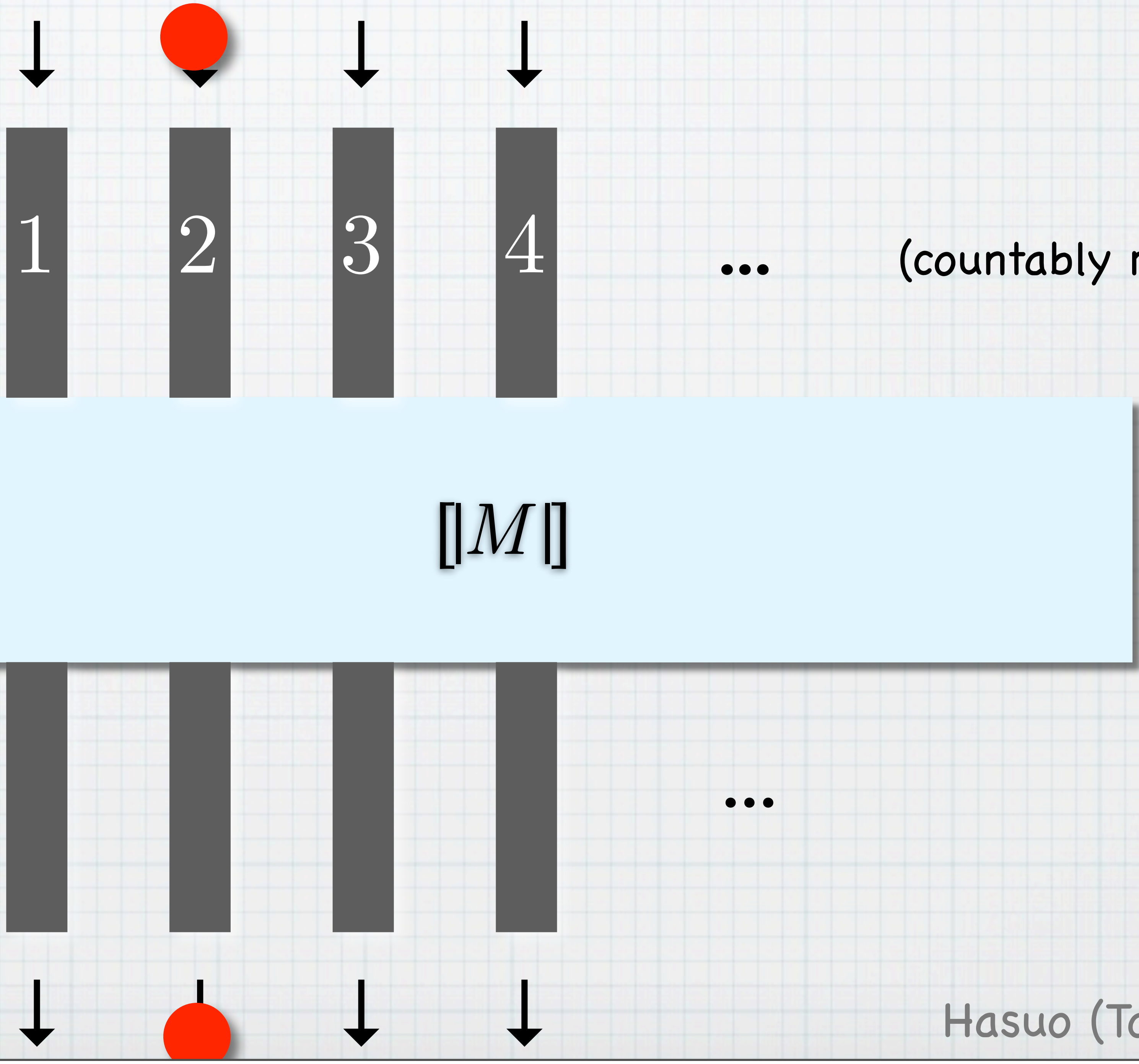


(countably many)

The GoI Animation

$$[\![M]\!] \;=\; (\,\mathbb{N} \rightharpoonup \mathbb{N},\ \text{a partial function}\,)$$

$$=\ \text{“piping”}$$



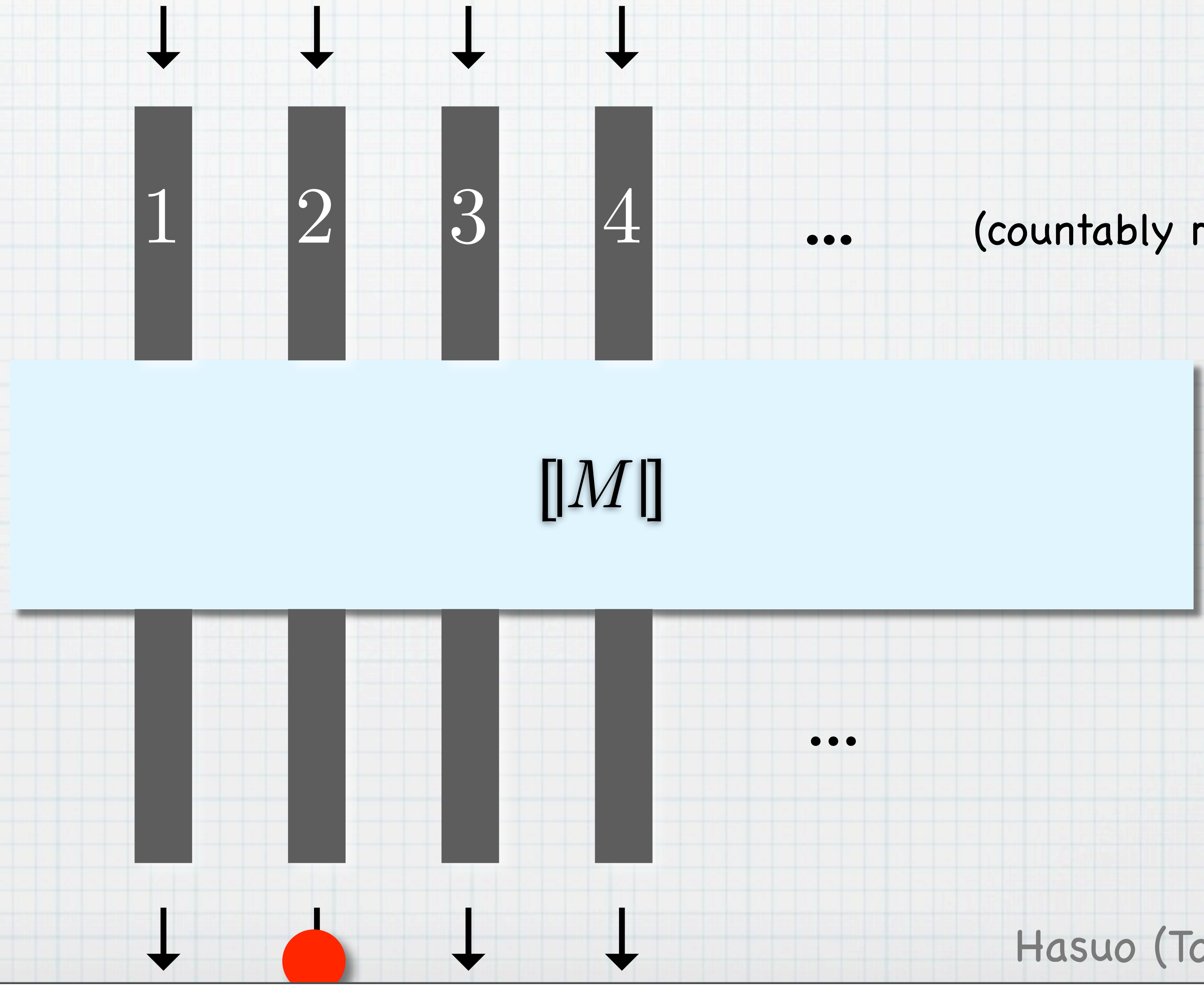
(countably many)

Hasuo (Tokyo)

The GoI Animation

$$[\![M]\!] \;=\; (\,\mathbb{N} \rightharpoonup \mathbb{N},\ \text{a partial function}\,)$$

$$=\ \text{"piping"}$$



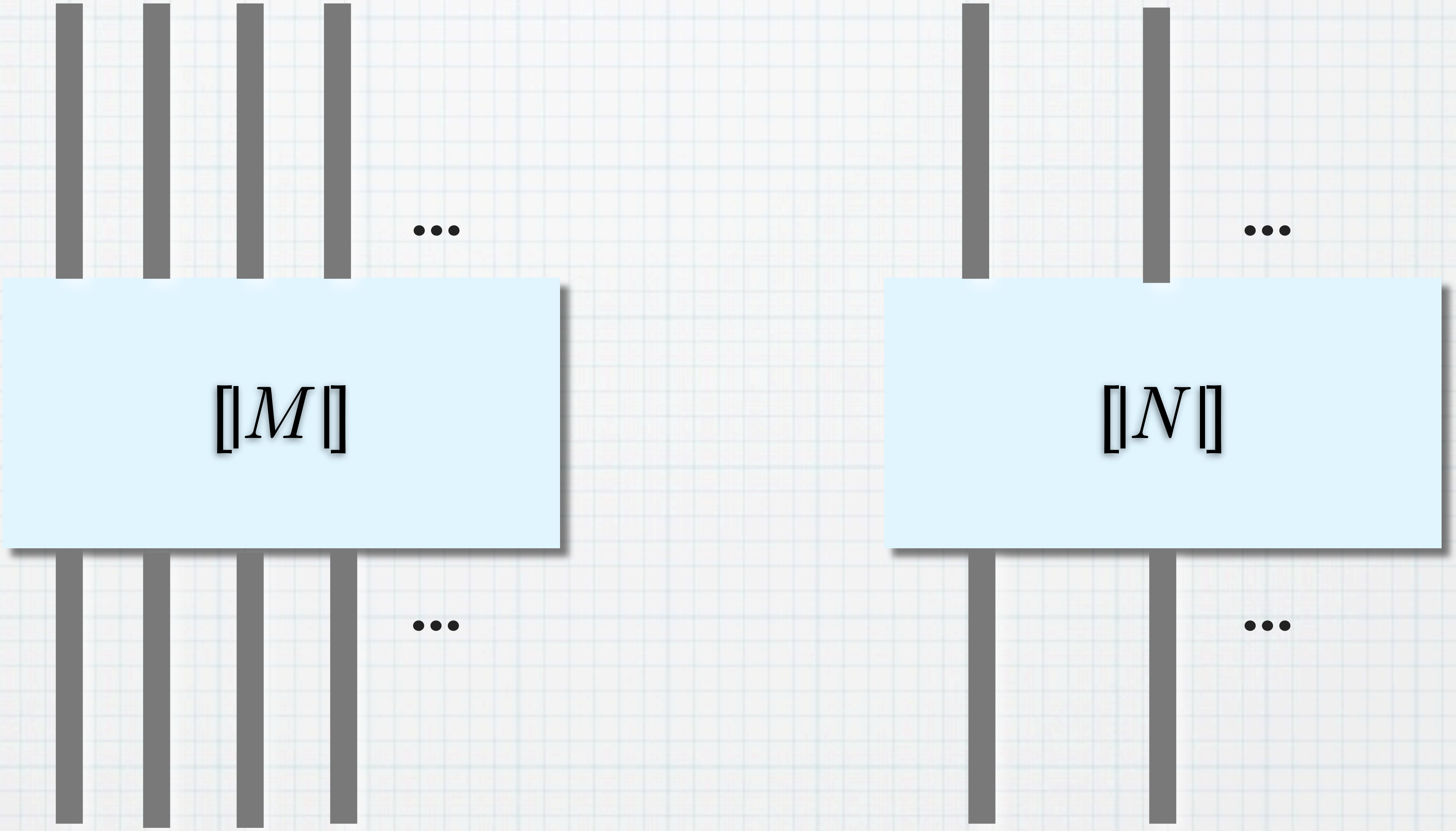
(countably many)

The GoI Animation

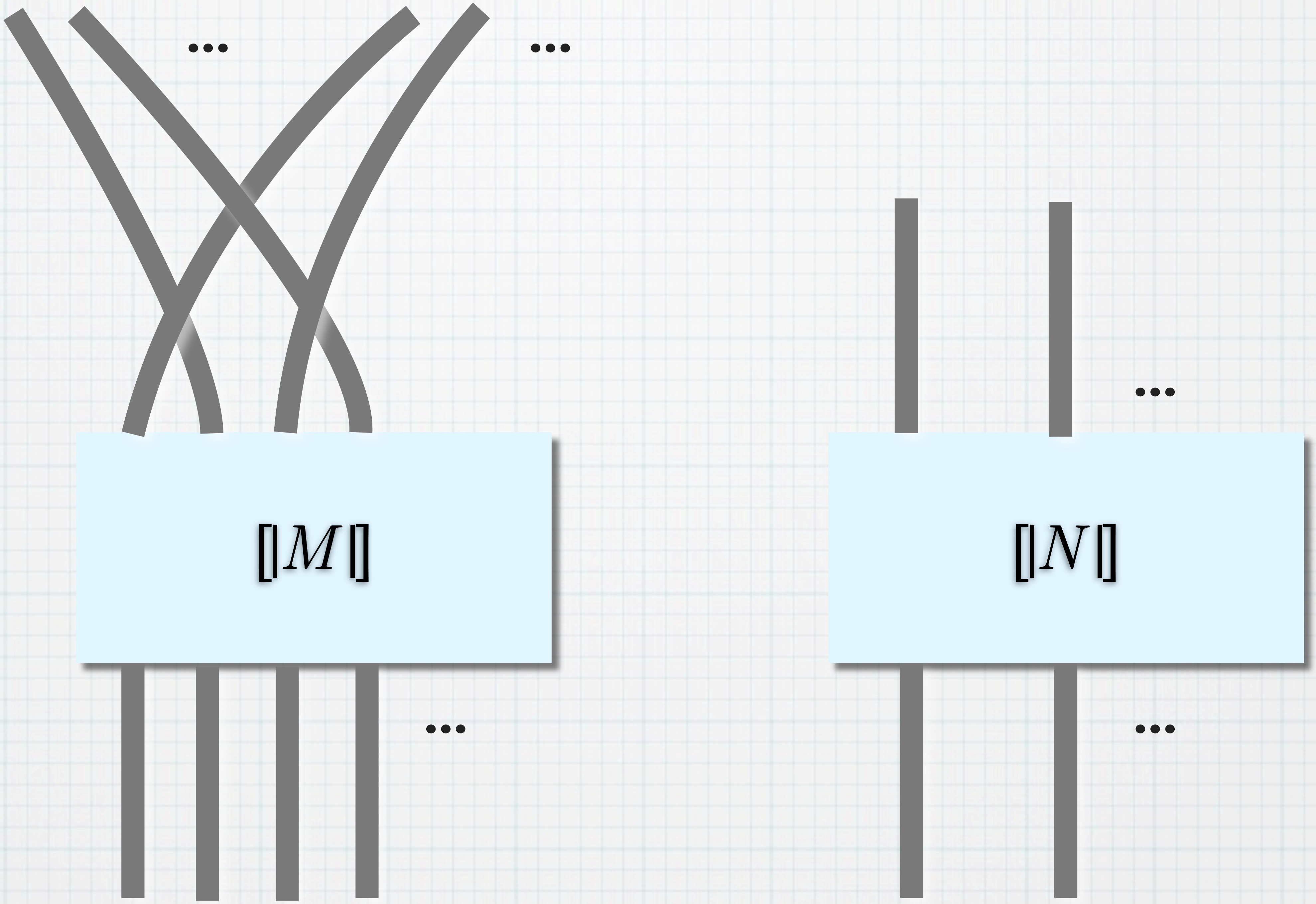
- Function application $[\![M N]\!]$
  - by “parallel composition + hiding”

Hasuo (Tokyo)

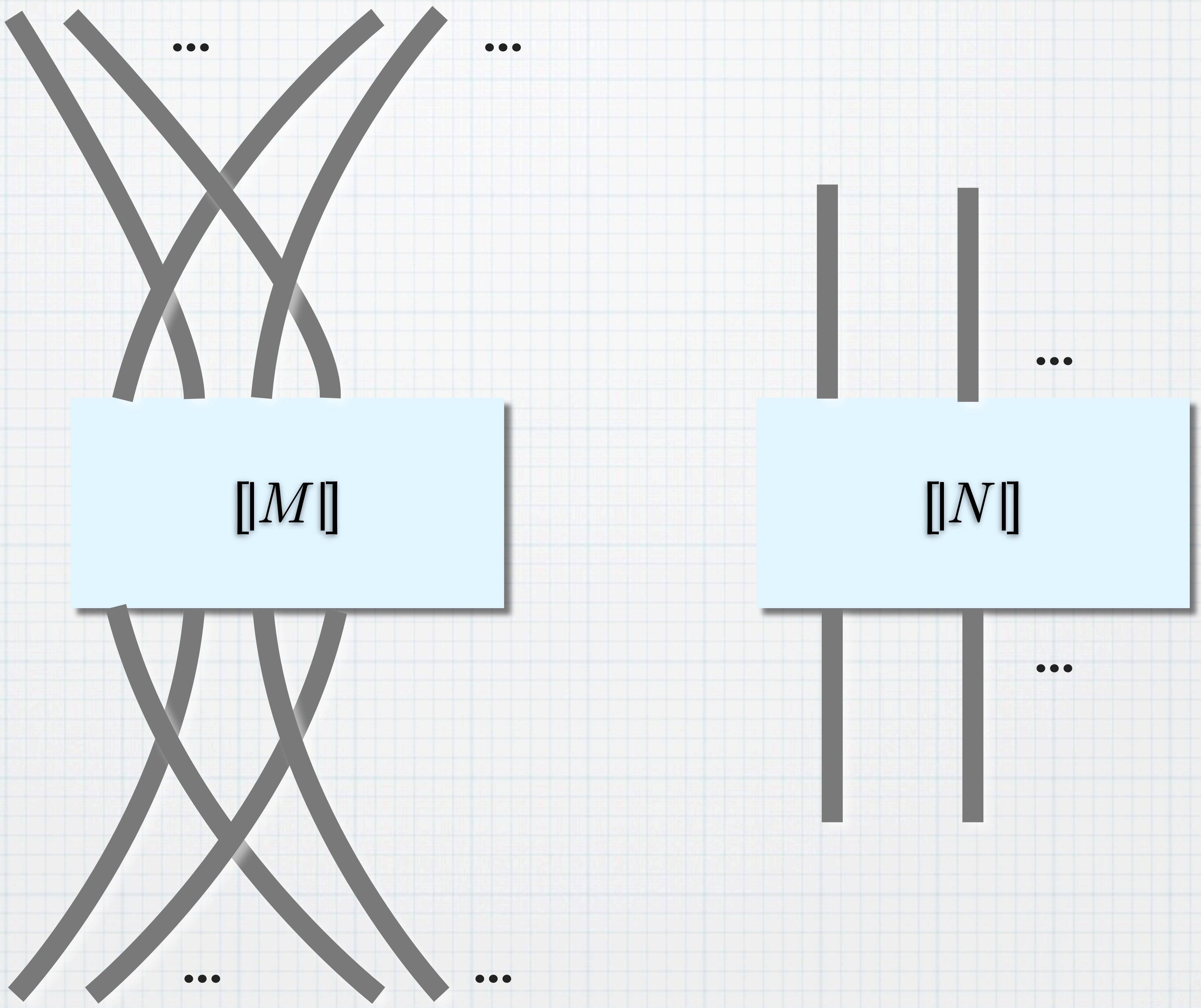
$$[\![\boldsymbol{M N}]\!] =$$



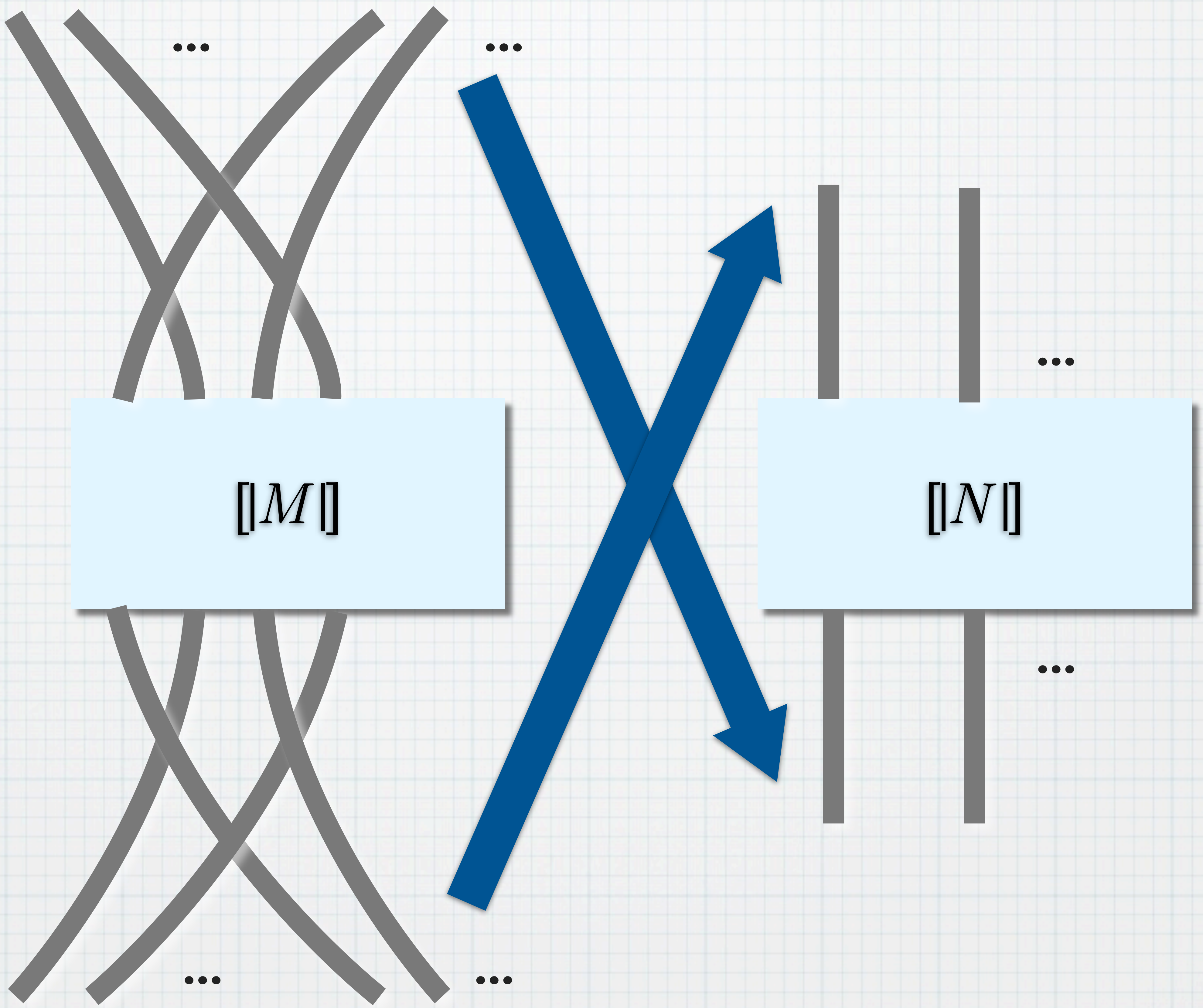
$$[\![\boldsymbol{M}\,\boldsymbol{N}]\!] =$$



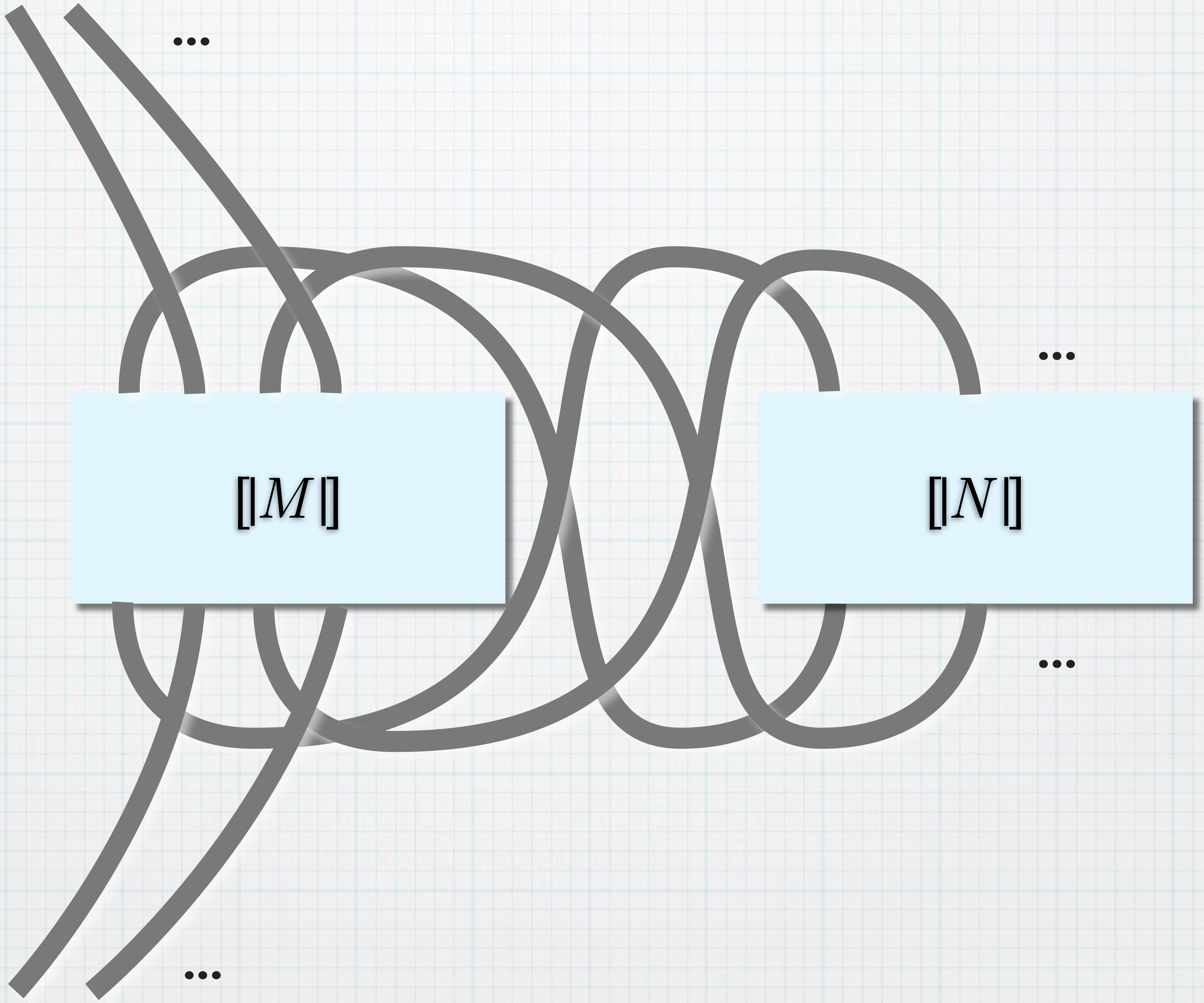
$$[\![\boldsymbol{M}\boldsymbol{N}]\!] =$$



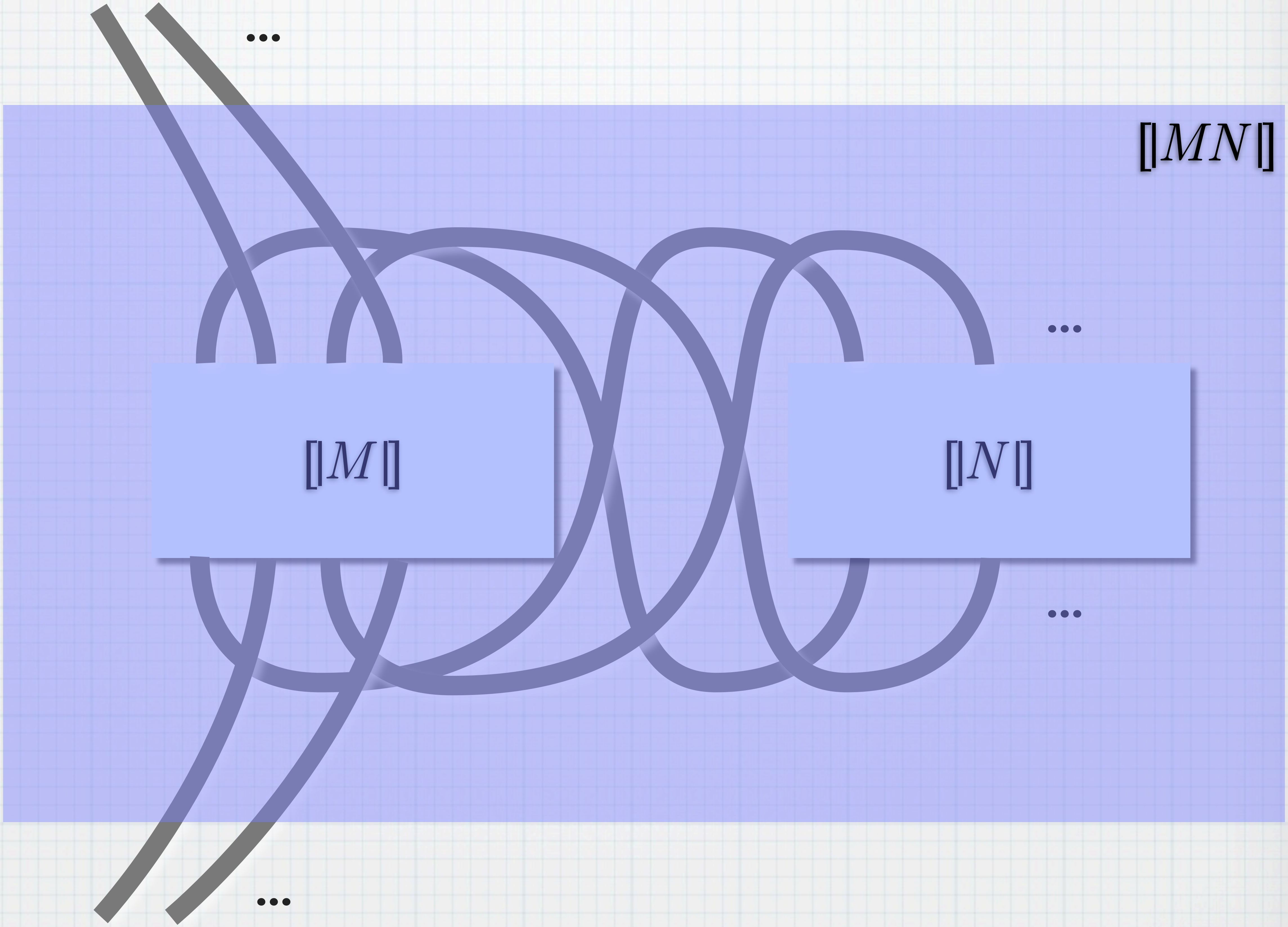
$$[\![MN]\!] =$$



$$[\![\boldsymbol{M} \boldsymbol{N}]\!] =$$

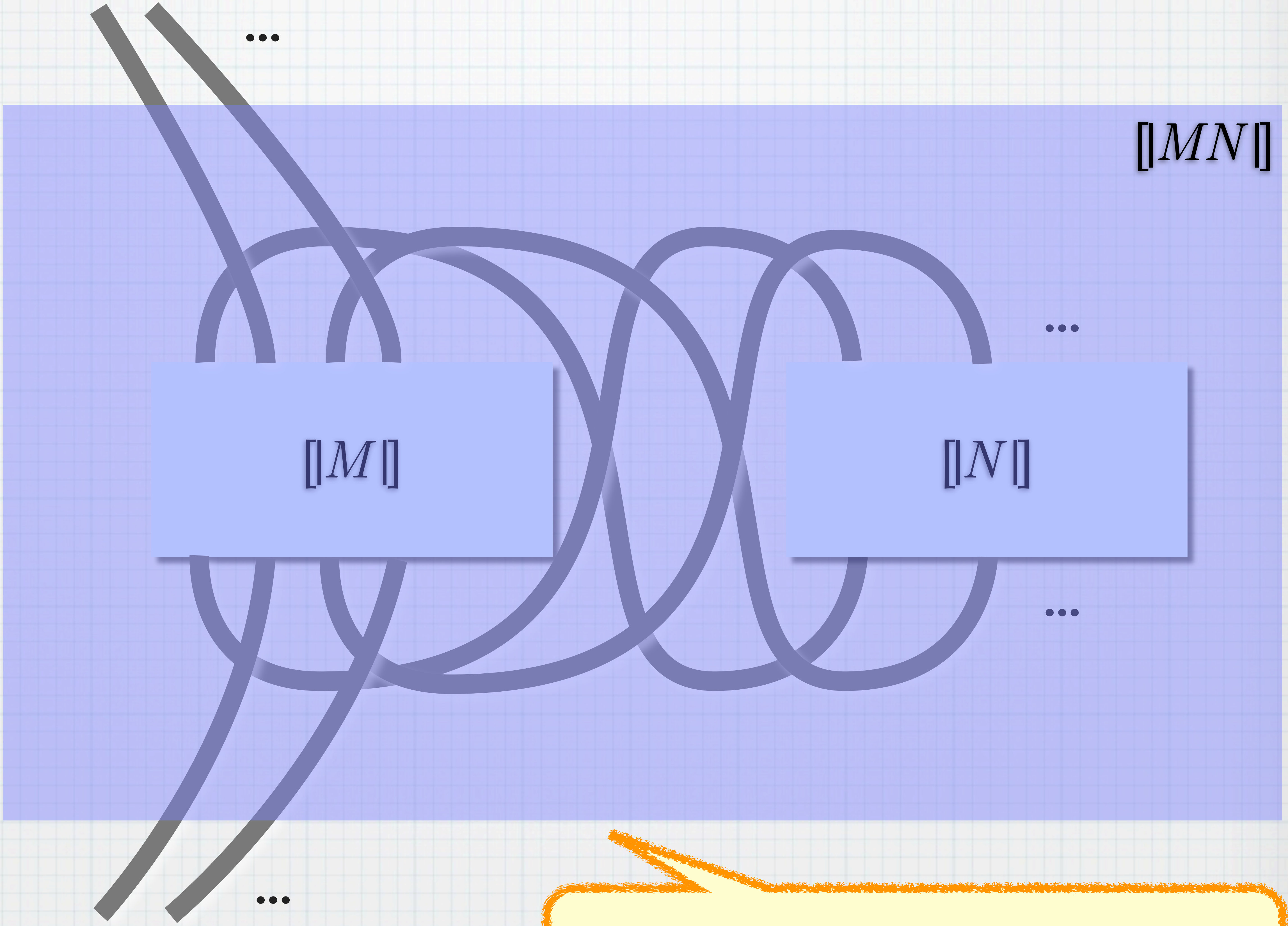


$$\llbracket \boldsymbol{M N} \rrbracket =$$


...
$\llbracket MN \rrbracket$
...
$\llbracket M \rrbracket$
$\llbracket N \rrbracket$
...
...

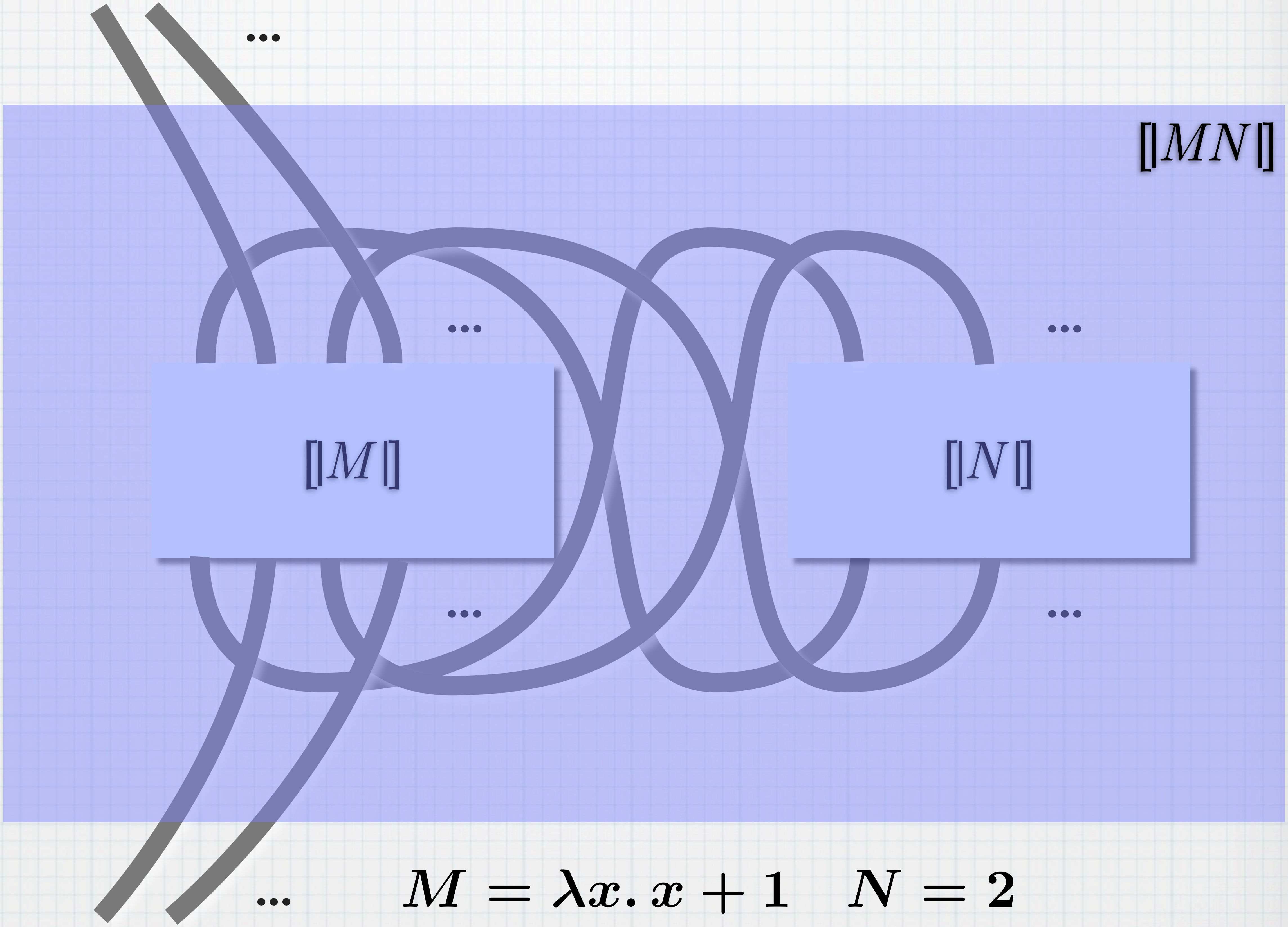
$[\![MN]\!]$

$=$



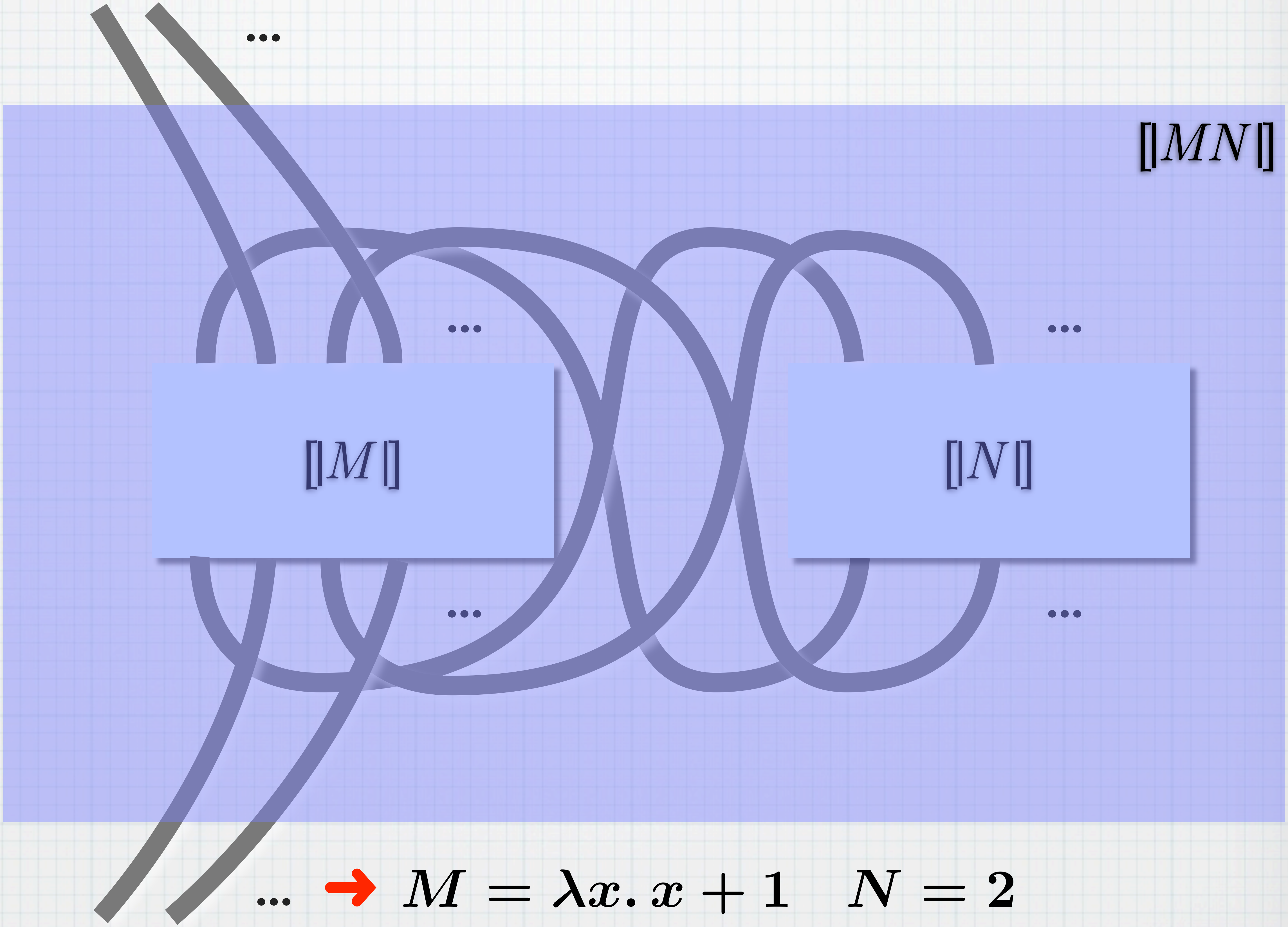
"parallel composition + hiding"
(cf. games)

$$[\![\boldsymbol{MN}]\!] =$$



$$\begin{array}{ll} \boldsymbol{M = \lambda x.\, x + 1} & \boldsymbol{N = 2} \\ \boldsymbol{M = \lambda x.\, 1} & \boldsymbol{N = 2} \\ \boldsymbol{M = \lambda f.\, f 1} & \boldsymbol{N = \lambda x.\, (x + 1)} \end{array}$$

$$[\![\boldsymbol{MN}]\!] =$$

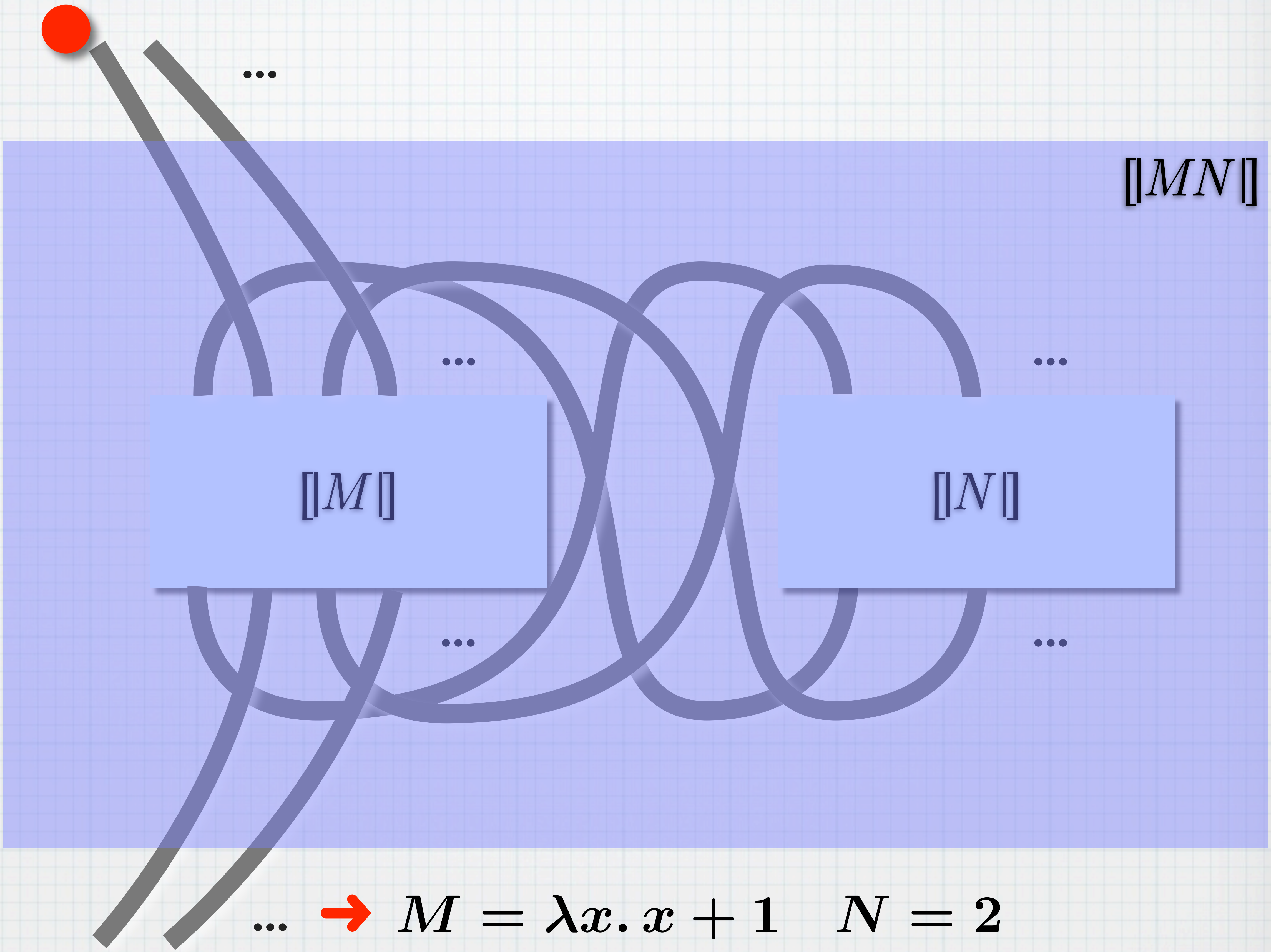


➜ $\boldsymbol{M = \lambda x.\, x + 1 \qquad N = 2}$

$\boldsymbol{M = \lambda x.\, 1 \qquad N = 2}$

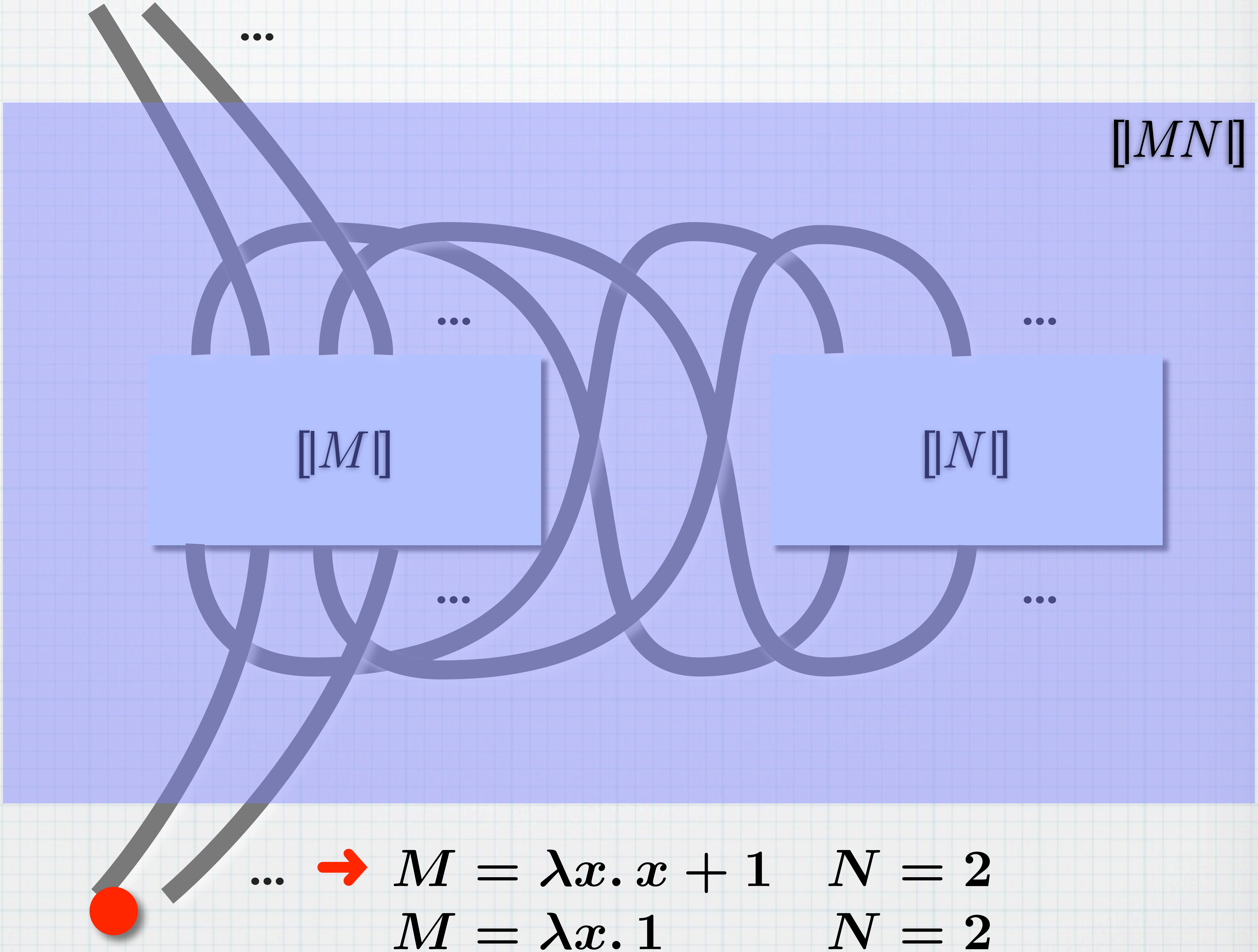
$\boldsymbol{M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)}$

$$[\![MN]\!] =$$



$$\begin{array}{llll} \rightarrow & M = \lambda x.\, x+1 & & N = 2 \\ & M = \lambda x.\, 1 & & N = 2 \\ & M = \lambda f.\, f 1 & & N = \lambda x.\, (x+1) \end{array}$$

$$[\![MN]\!] =$$

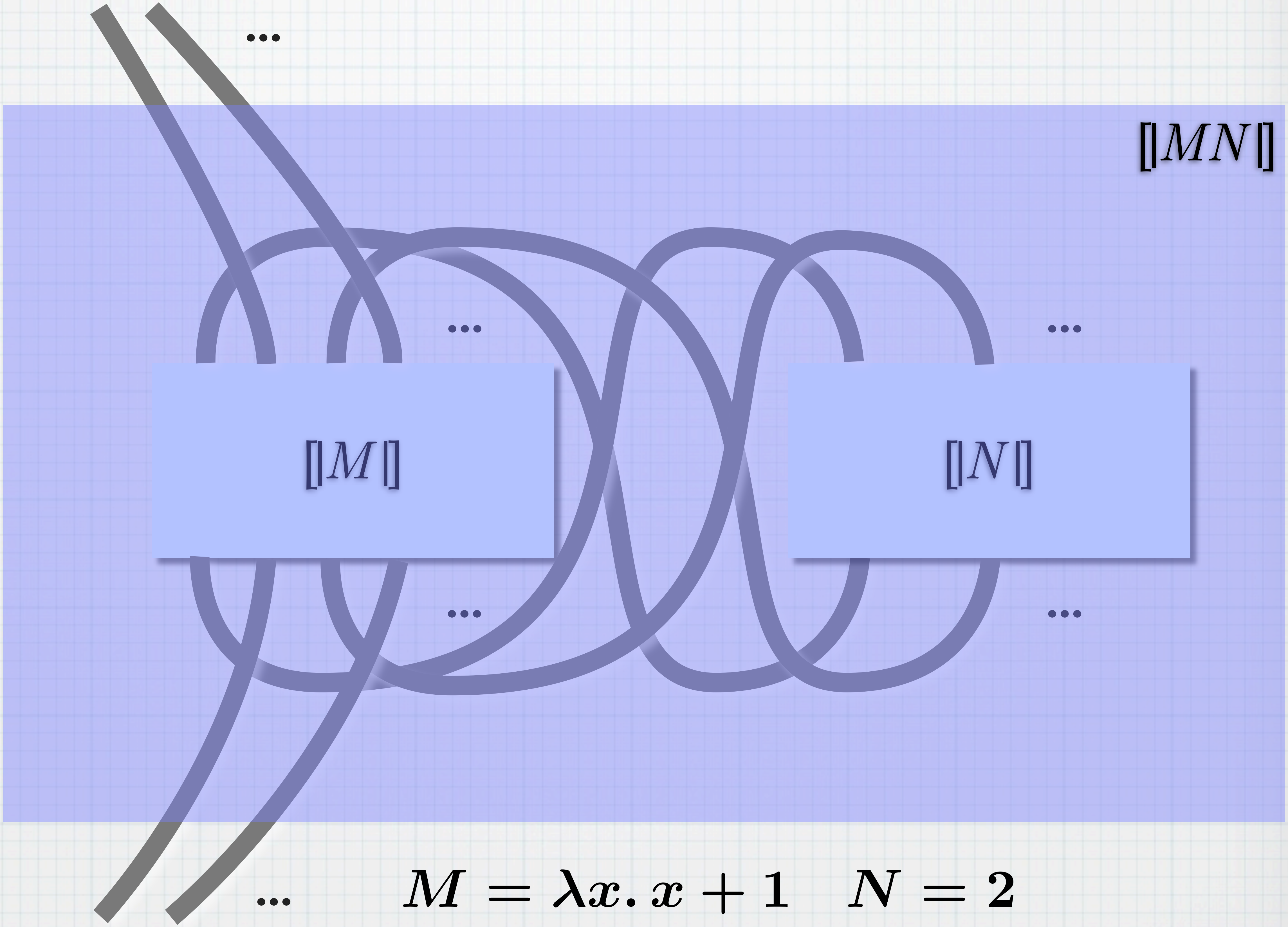


$\rightarrow$ $M = \lambda x.\, x + 1$ $N = 2$

$M = \lambda x.\, 1$ $N = 2$

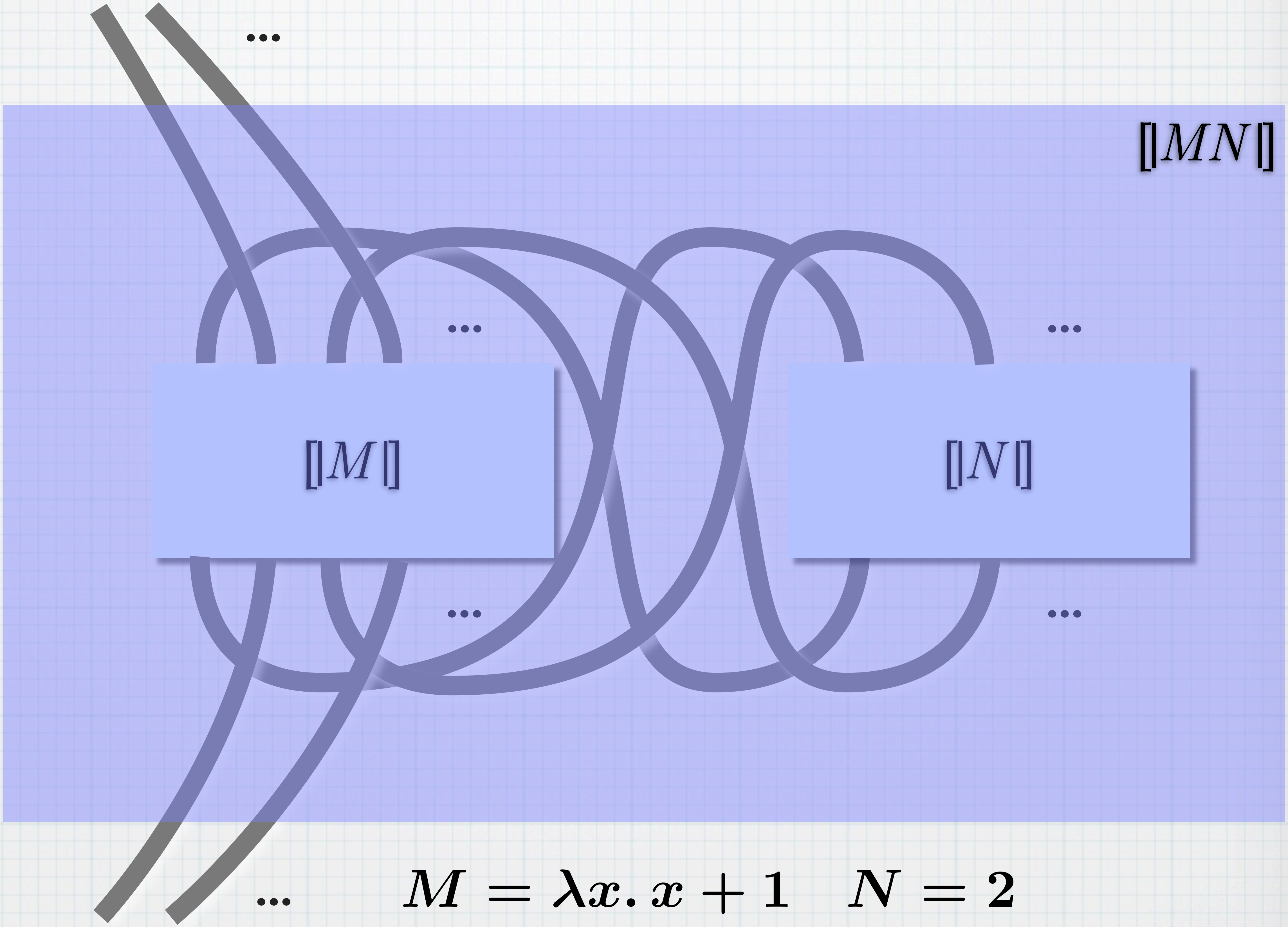
$M = \lambda f.\, f 1$ $N = \lambda x.\, (x + 1)$

$$[\![\boldsymbol{MN}]\!] =$$



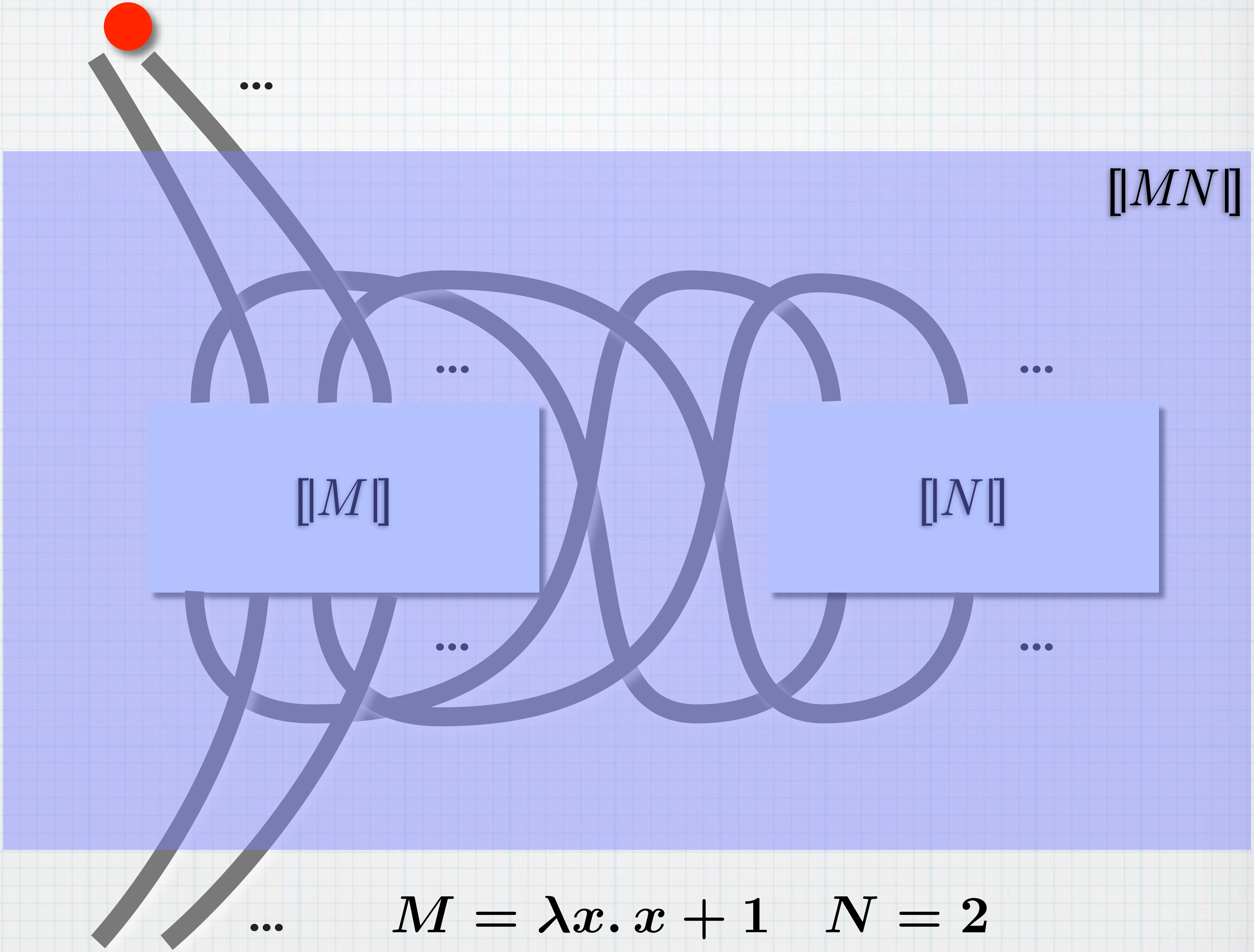
$$\begin{aligned}
\boldsymbol{M} &= \boldsymbol{\lambda x.\, x + 1} & \boldsymbol{N} &= \boldsymbol{2} \\
\boldsymbol{M} &= \boldsymbol{\lambda x.\, 1} & \boldsymbol{N} &= \boldsymbol{2} \\
\boldsymbol{M} &= \boldsymbol{\lambda f.\, f 1} & \boldsymbol{N} &= \boldsymbol{\lambda x.\, (x + 1)}
\end{aligned}$$

$[\![MN]\!] =$



$$\begin{array}{rll} & M = \lambda x.\, x+1 & N = 2 \\ \rightarrow & M = \lambda x.\, 1 & N = 2 \\ & M = \lambda f.\, f 1 & N = \lambda x.\, (x+1) \end{array}$$

$$\llbracket MN \rrbracket =$$



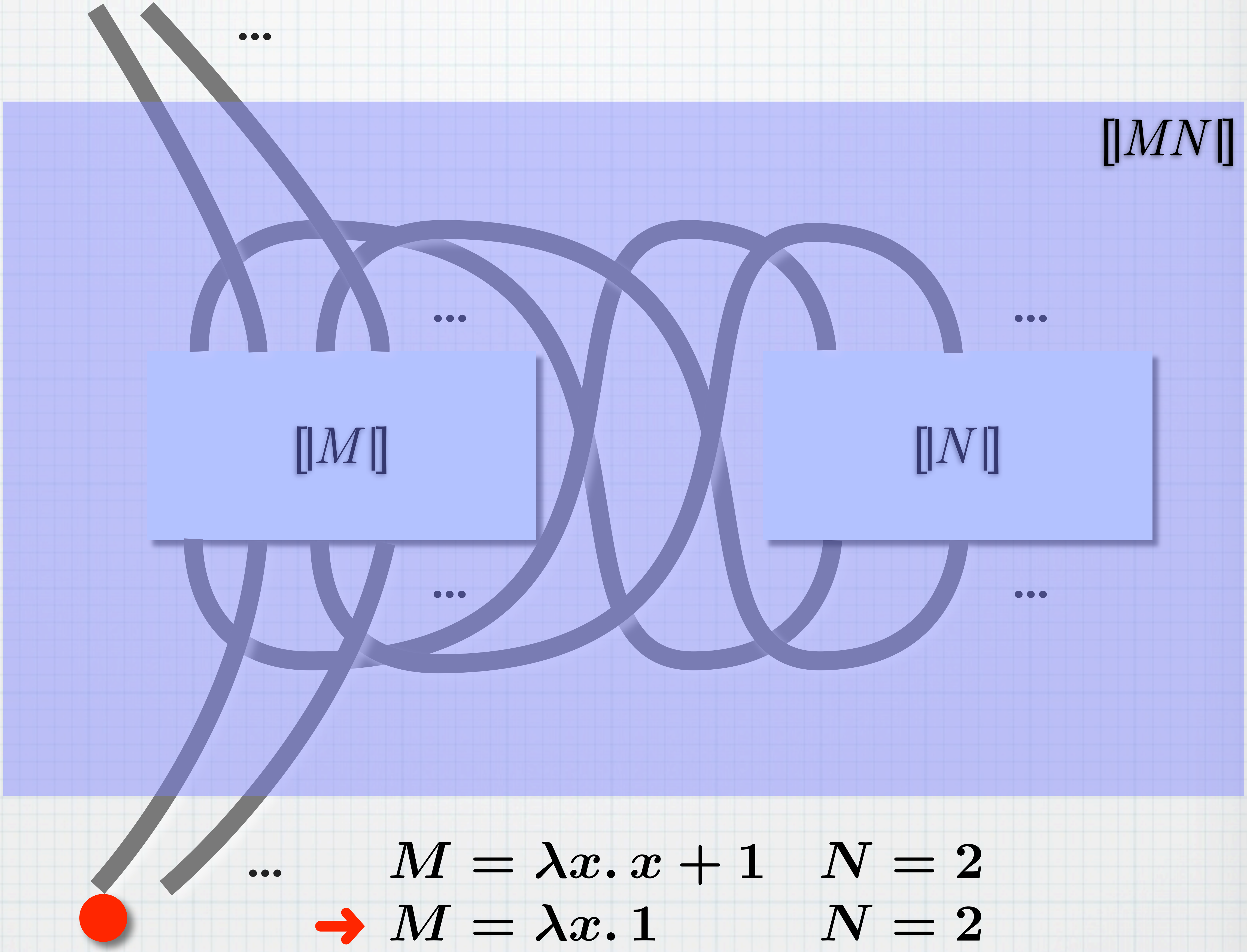
$\llbracket MN \rrbracket$
$\llbracket M \rrbracket$
$\llbracket N \rrbracket$
...
...
...
...
...
...

$$\boldsymbol{M = \lambda x.\, x + 1 \qquad N = 2}$$

→ $\boldsymbol{M = \lambda x.\, 1 \qquad N = 2}$

$$\boldsymbol{M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)}$$

$$[\![MN]\!] =$$



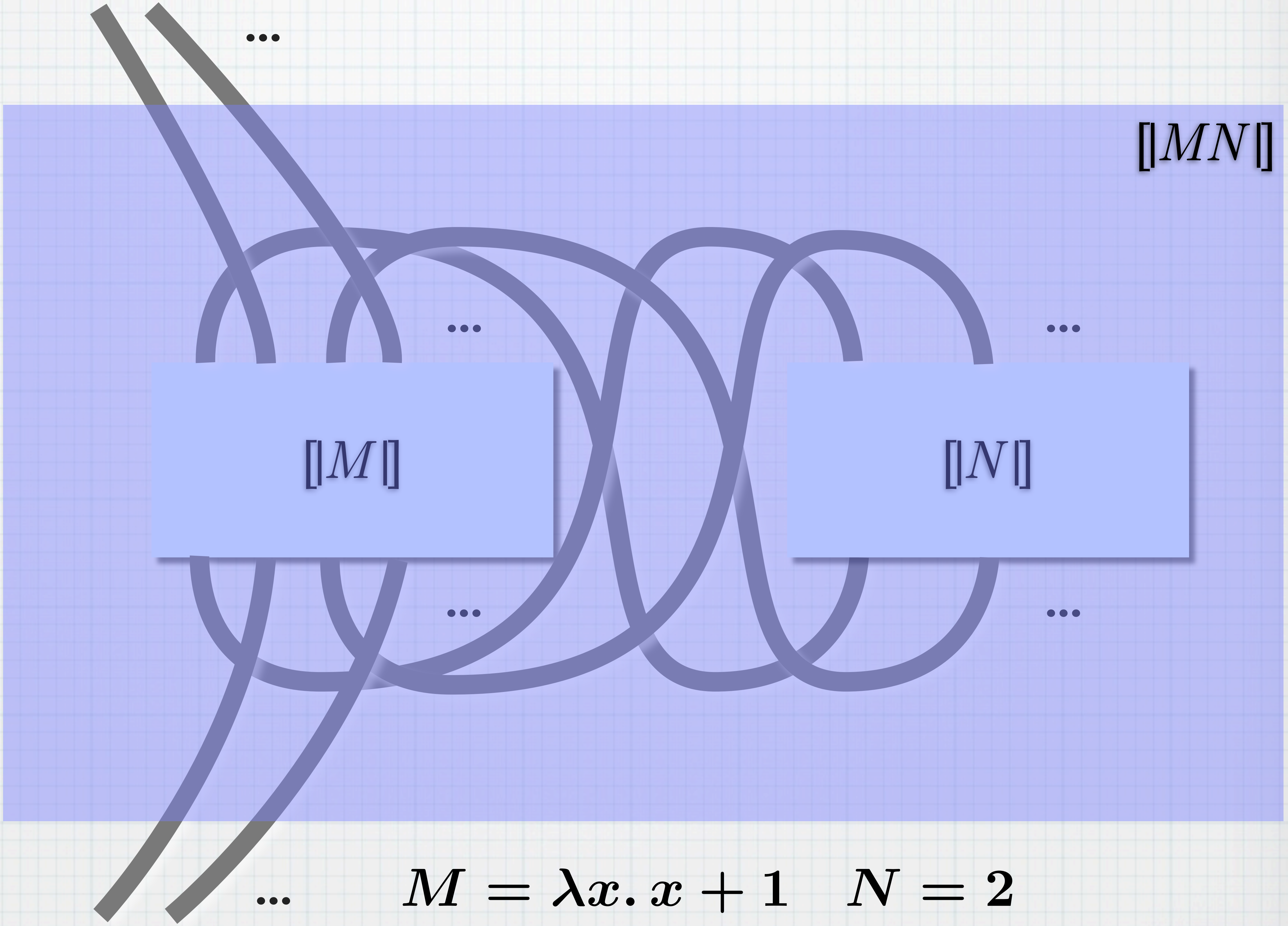
$M = \lambda x.\, x + 1 \qquad N = 2$

➜ $M = \lambda x.\, 1 \qquad N = 2$

$M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)$

$[\![MN]\!]$

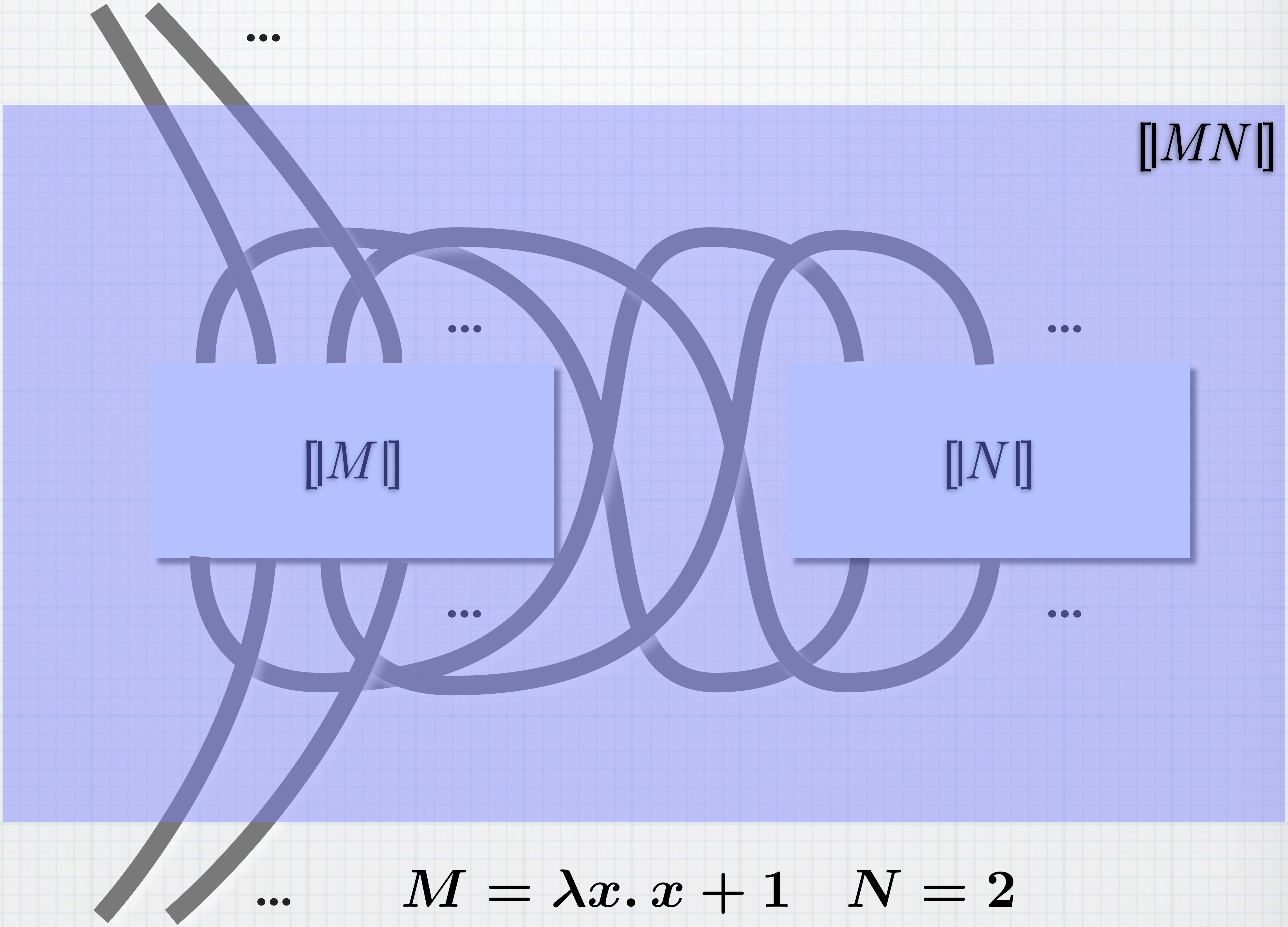
$=$



$$\begin{aligned} &M = \lambda x.\, x+1 && N = 2 \\ &M = \lambda x.\, 1 && N = 2 \\ &M = \lambda f.\, f1 && N = \lambda x.\, (x+1) \end{aligned}$$

$\llbracket MN \rrbracket$

$=$

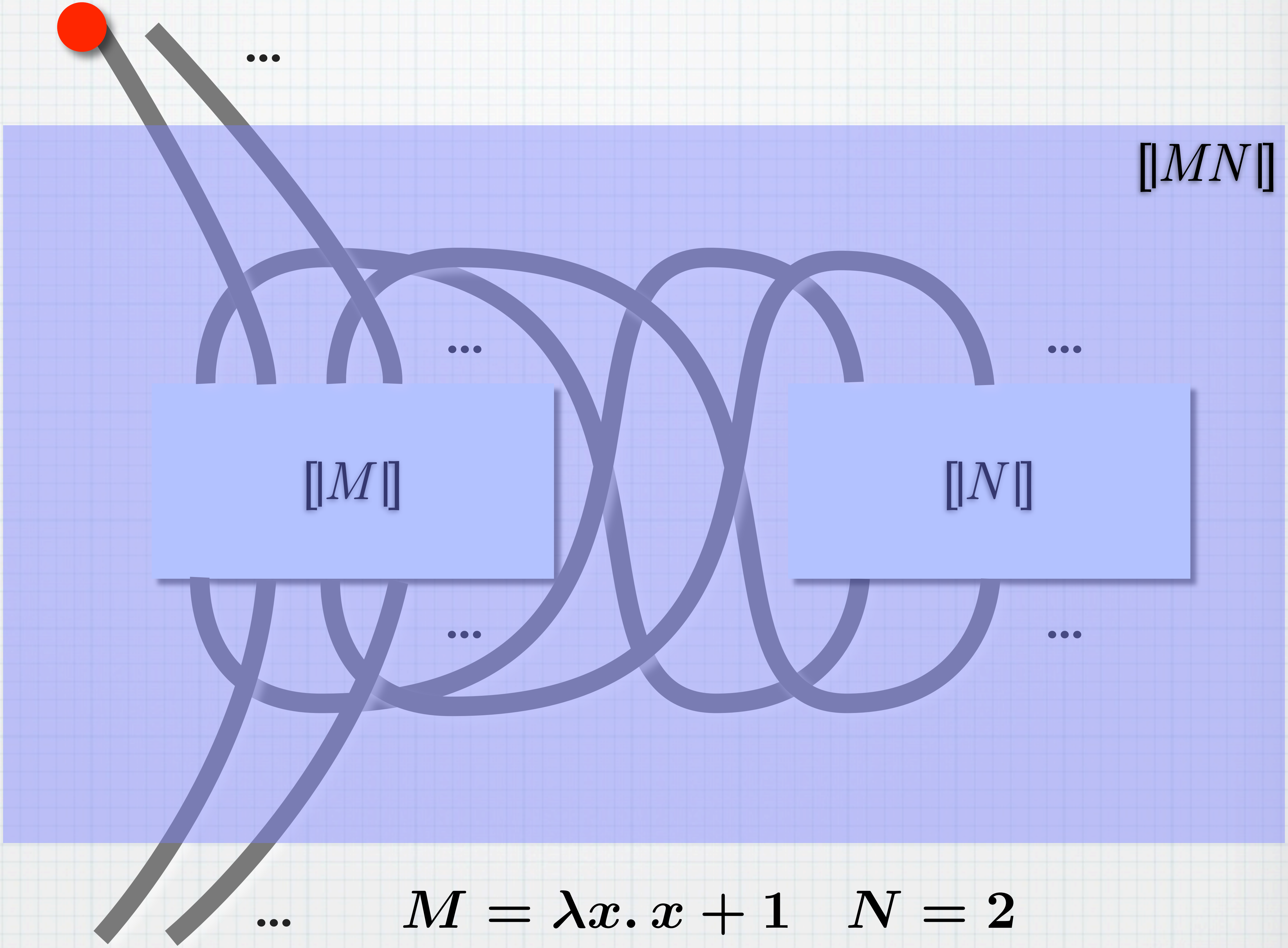

...
$\llbracket MN \rrbracket$
...
$\llbracket M \rrbracket$
...
$\llbracket N \rrbracket$
...
...
...

$M = \lambda x.\, x + 1 \qquad N = 2$

$M = \lambda x.\, 1 \qquad N = 2$

➜ $M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)$

$$[\![MN]\!] =$$



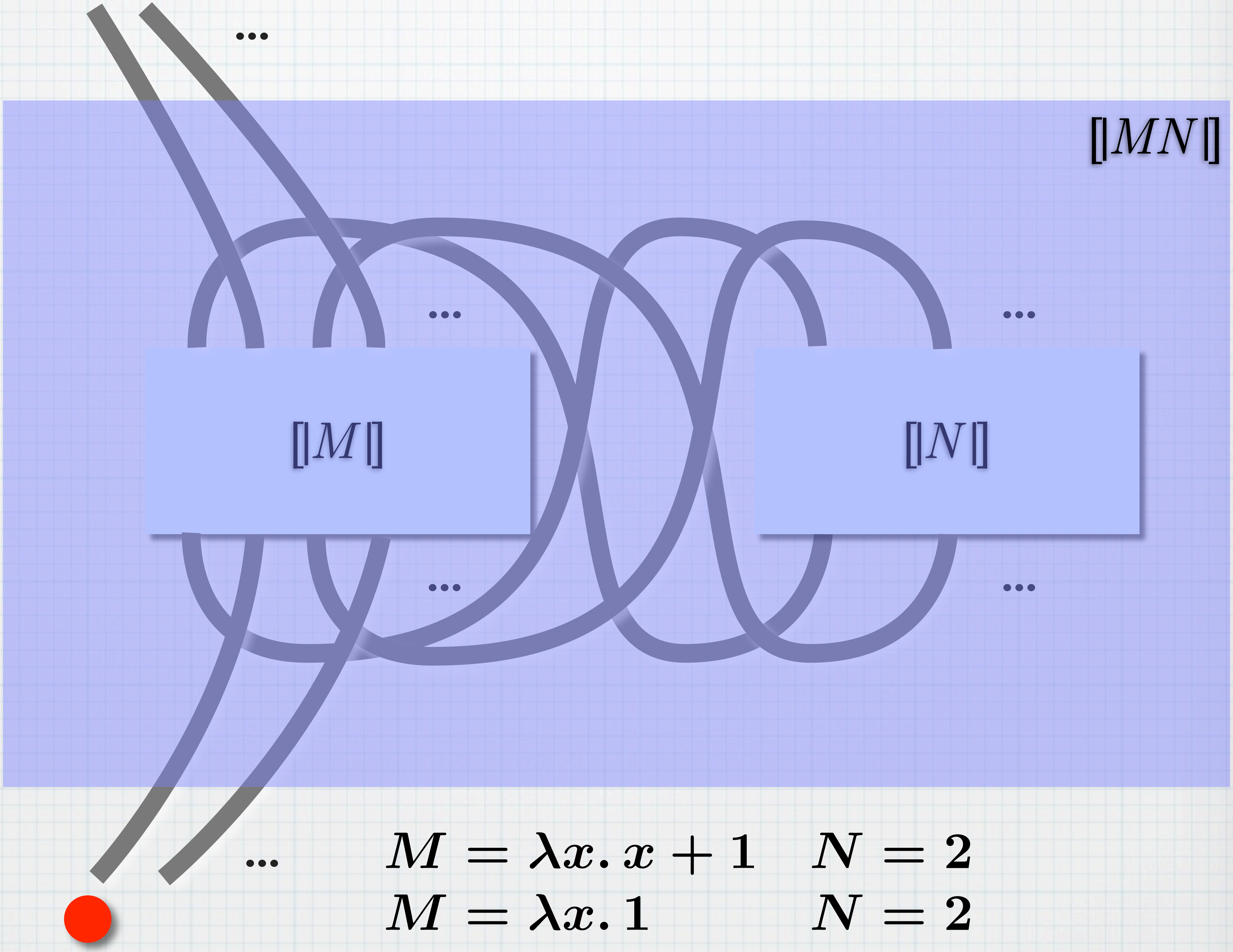
$$M = \lambda x.\, x + 1 \qquad N = 2$$

$$M = \lambda x.\, 1 \qquad N = 2$$

➜ $M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)$

$[\![MN]\!]$

$=$



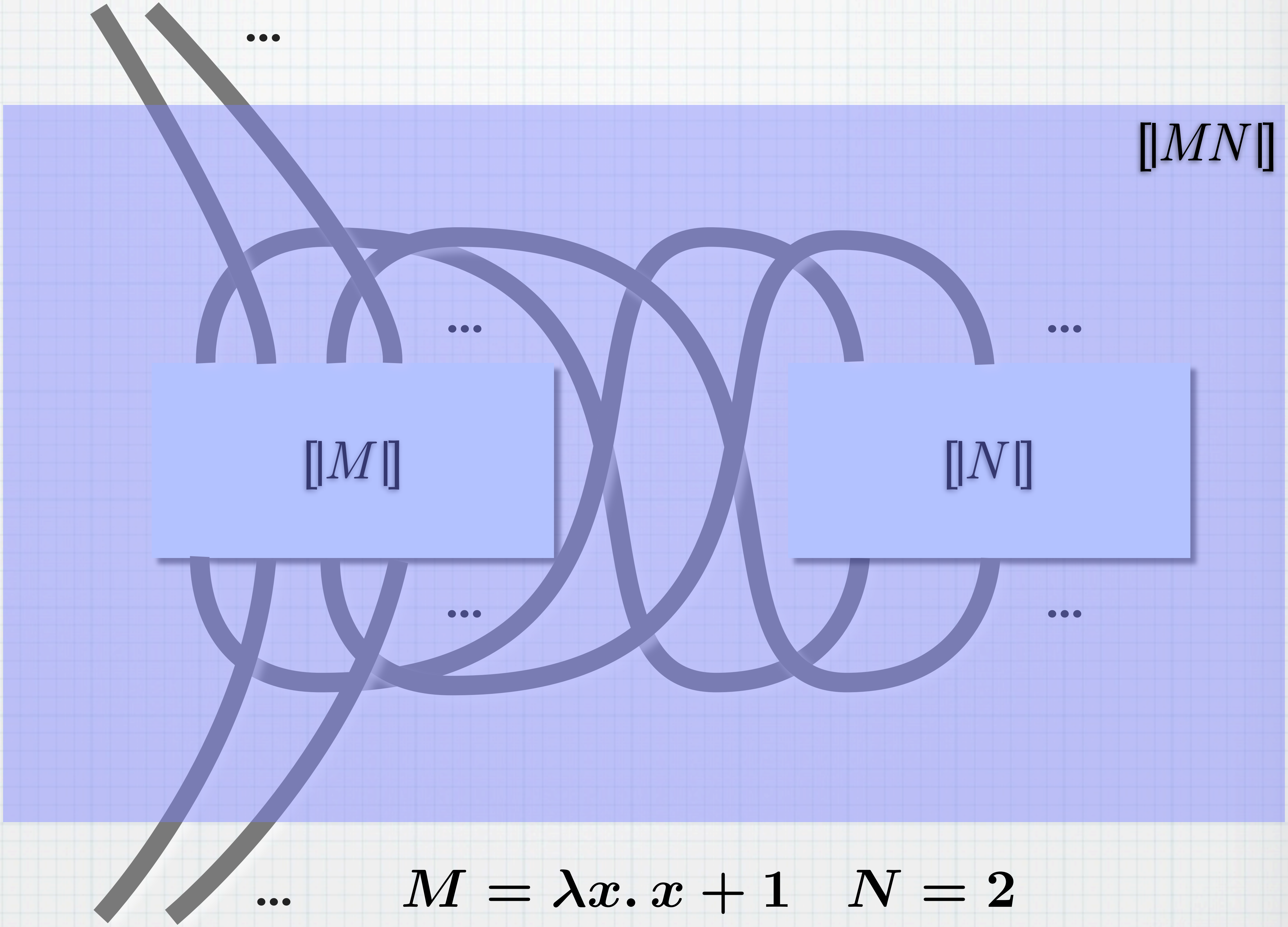
$M = \lambda x.\, x + 1 \quad N = 2$

$M = \lambda x.\, 1 \quad N = 2$

→ $M = \lambda f.\, f1 \quad N = \lambda x.\, (x + 1)$

$[\![MN]\!]$

$=$



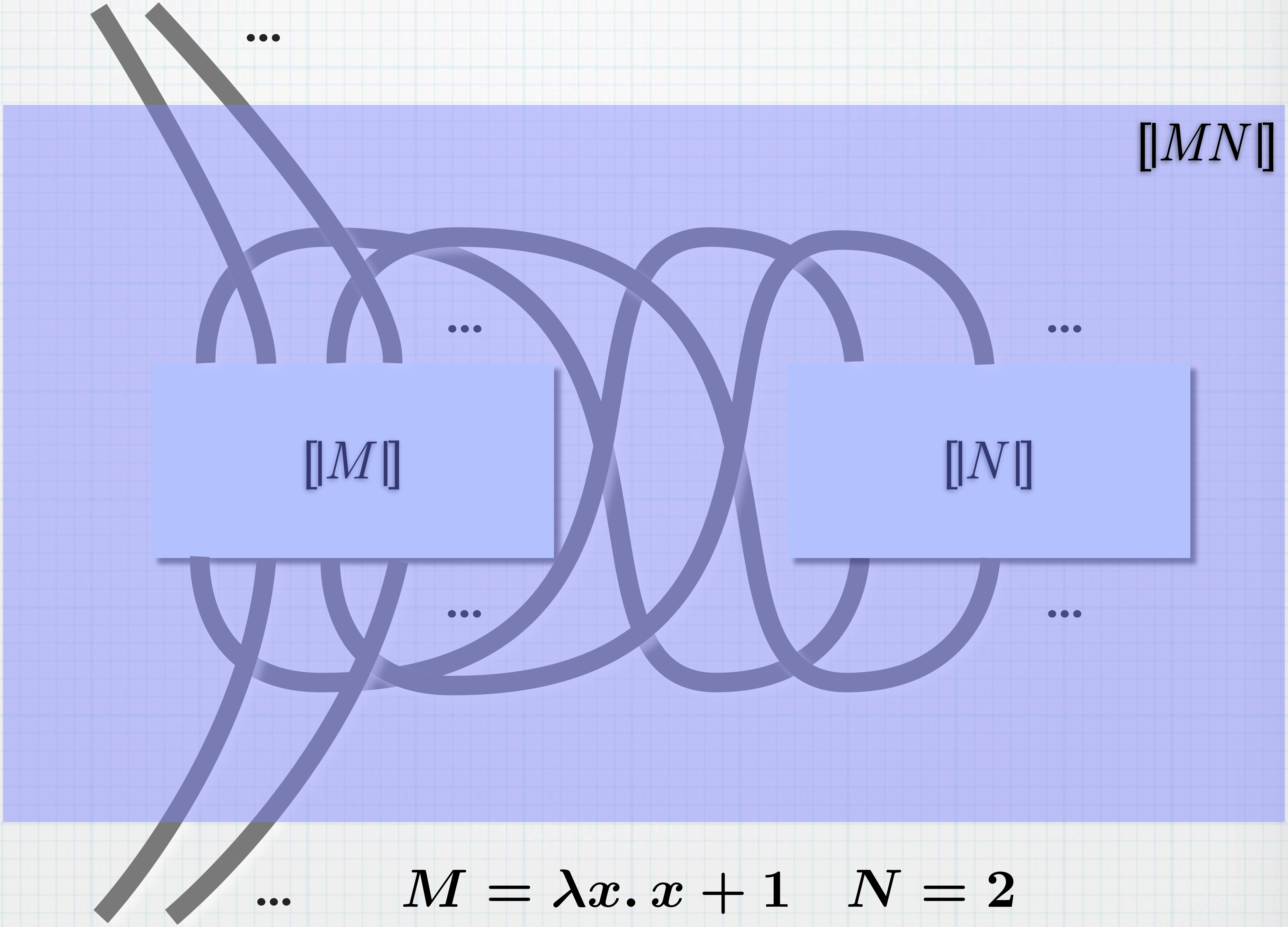
$M = \lambda x.\, x + 1 \qquad N = 2$

$M = \lambda x.\, 1 \qquad N = 2$

➜ $M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)$

$[\![\boldsymbol{MN}]\!]$

$=$



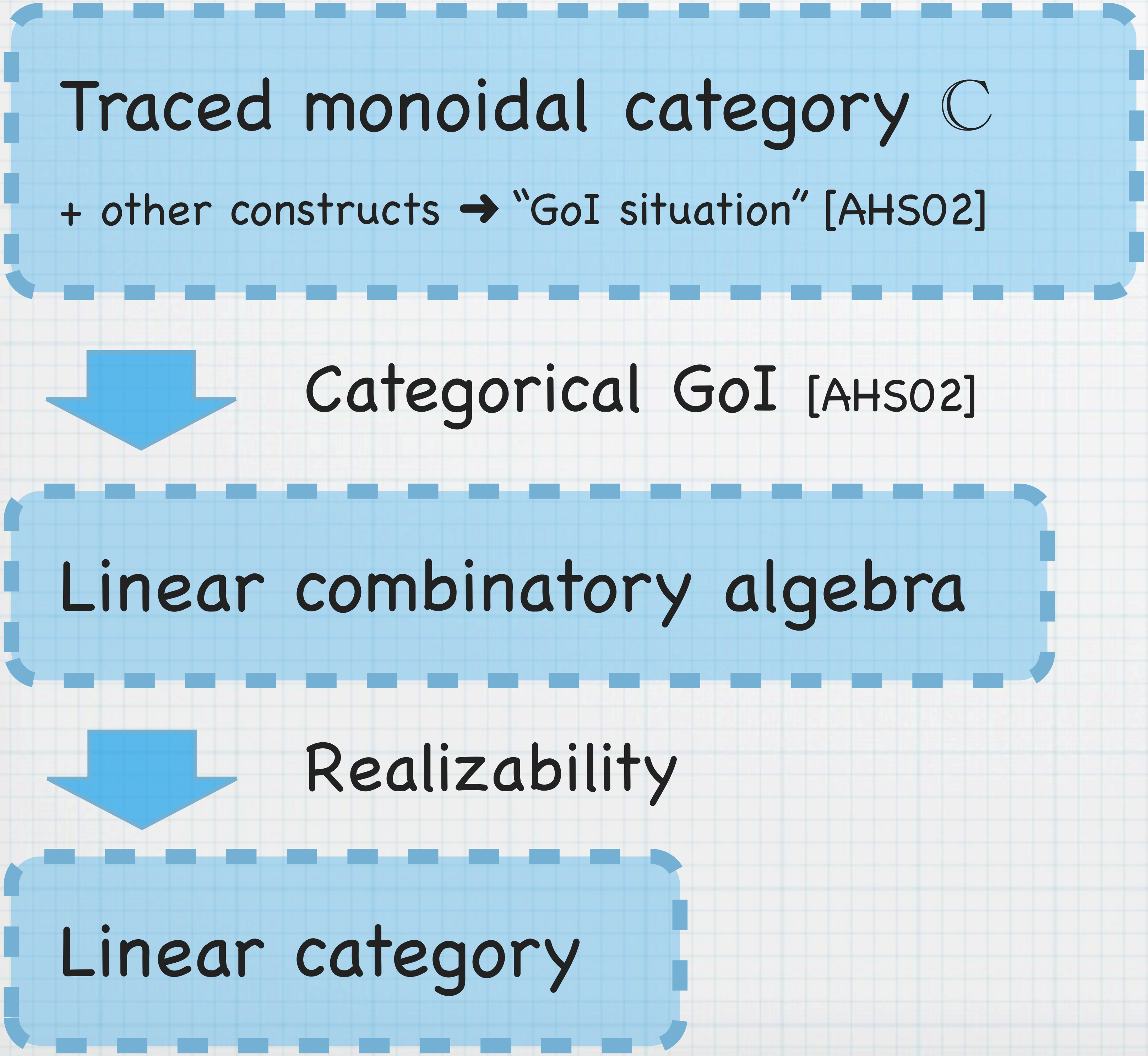
$$\begin{array}{ll} \boldsymbol{M = \lambda x.\, x + 1} & \boldsymbol{N = 2} \\ \boldsymbol{M = \lambda x.\, 1} & \boldsymbol{N = 2} \\ \boldsymbol{M = \lambda f.\, f 1} & \boldsymbol{N = \lambda x.\, (x + 1)} \end{array}$$

Categorical GoI

- Axiomatics of GoI in the categorical language

- Our main reference:
  - [AHS02] S. Abramsky, E. Haghverdi, and P. Scott, "Geometry of interaction and linear combinatory algebras," MSCS 2002
  - Especially its technical report version (Oxford CL), since it's a bit more detailed

The Categorical GoI Workflow



Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

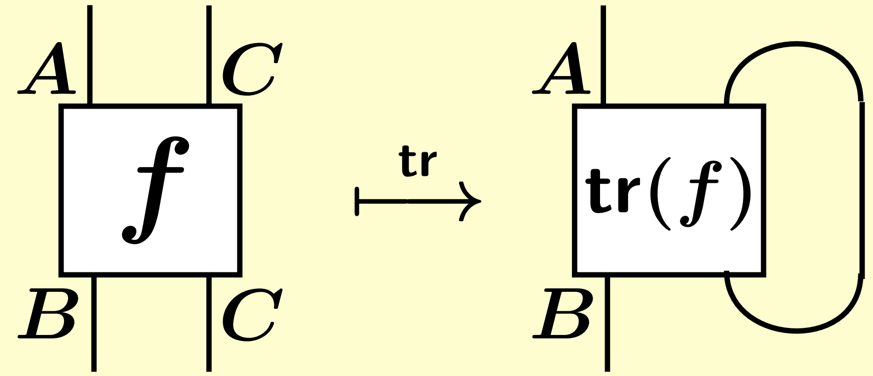
Realizability

Linear category

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ “GoI situation” [AHS02]



Categorical GoI [AHS02]

Linear combinatory algebra

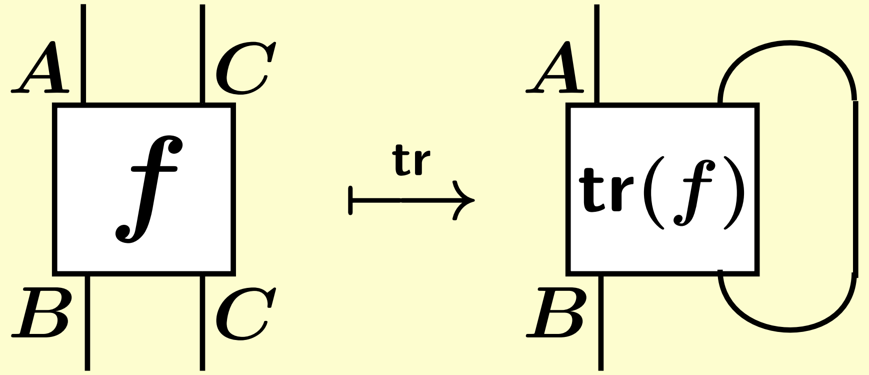
Realizability

Linear category

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]



Categorical GoI [AHS02]

- Applicative str. + combinators
- Model of untyped calculus

Linear combinatory algebra

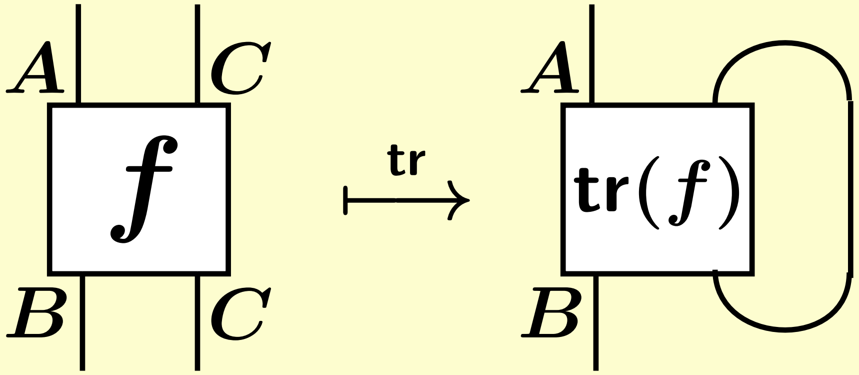
Realizability

Linear category

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]



Categorical GoI [AHS02]

Linear combinatory algebra

- Applicative str. + combinators
- Model of untyped calculus

Realizability

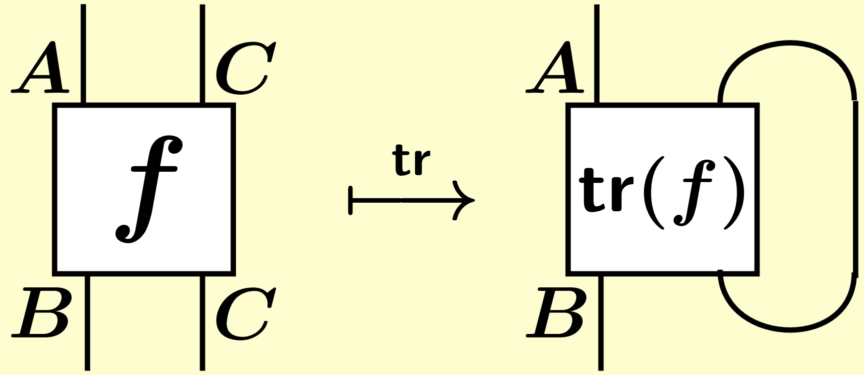
Linear category

Model of typed calculus

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]



Categorical GoI [AHS02]

- Applicative str. + combinators
- Model of untyped calculus

Linear combinatory algebra

Realizability

- PER, ω-set, assembly, ...
- "Programming in untyped λ"

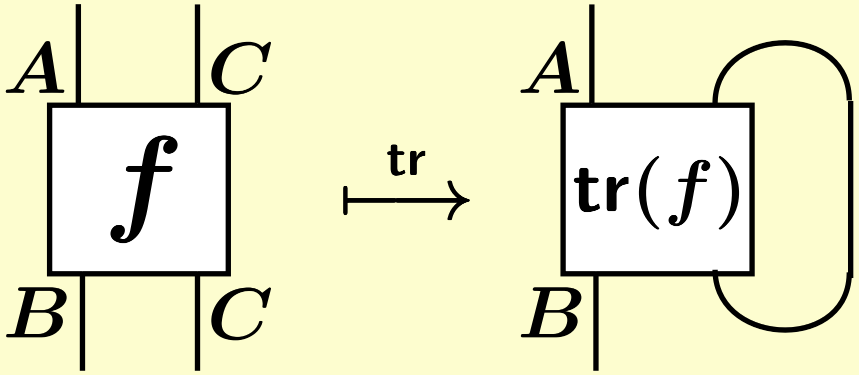
Linear category

Model of typed calculus

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ “GoI situation” [AHS02]



Categorical GoI [AHS02]

- Applicative str. + combinators
- Model of untyped calculus

Linear combinatory algebra

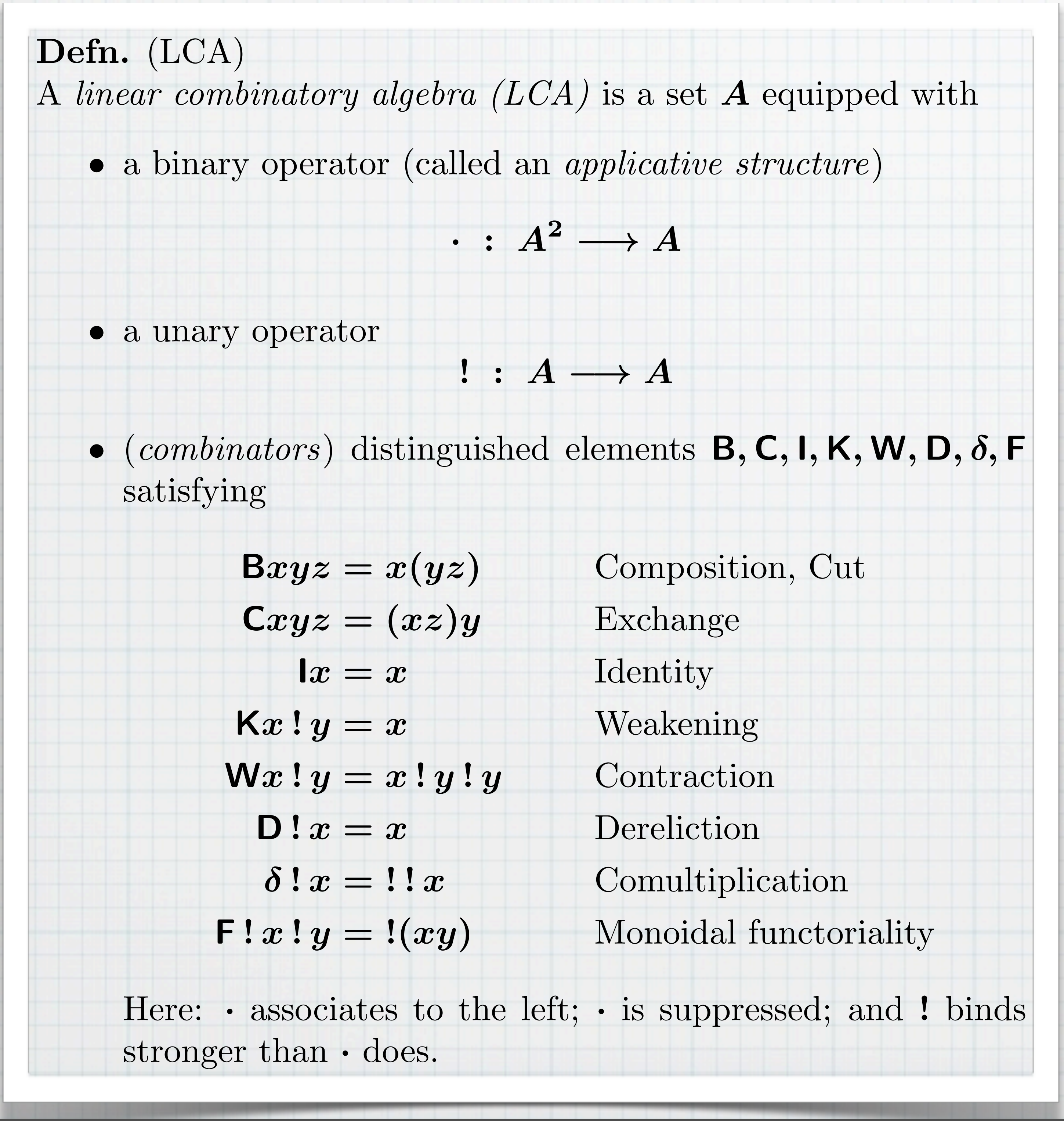
Realizability

- PER, ω-set, assembly, ...
- “Programming in untyped λ”

Linear category

Model of typed calculus

Linear Combinatory Algebra (LCA)



Defn. (LCA)
A *linear combinatory algebra (LCA)* is a set $\boldsymbol{A}$ equipped with

- a binary operator (called an *applicative structure*)

$$\cdot \; : \; \boldsymbol{A}^2 \longrightarrow \boldsymbol{A}$$

- a unary operator

$$! \; : \; \boldsymbol{A} \longrightarrow \boldsymbol{A}$$

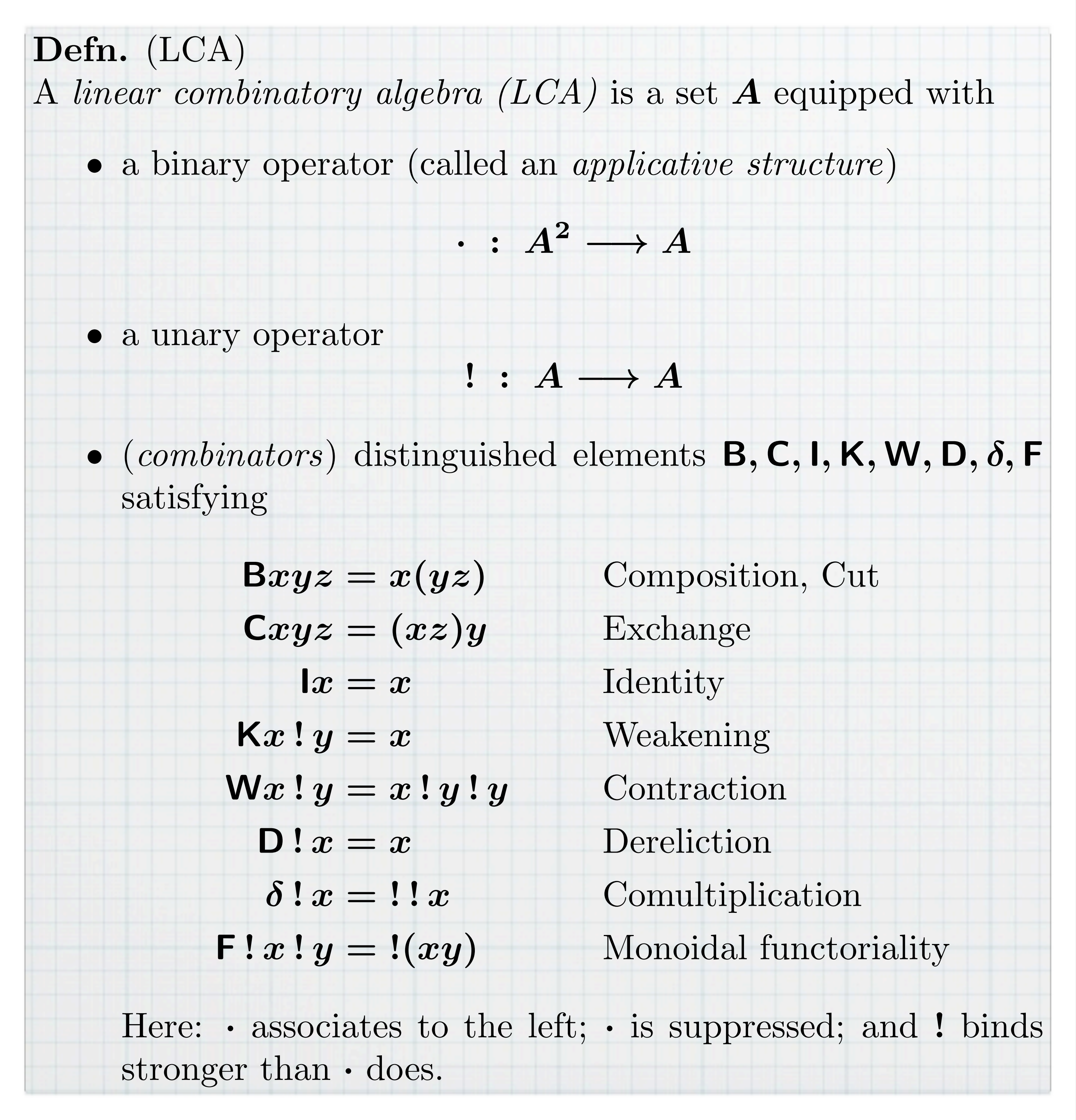
- (*combinators*) distinguished elements $\mathbf{B}, \mathbf{C}, \mathbf{I}, \mathbf{K}, \mathbf{W}, \mathbf{D}, \boldsymbol{\delta}, \mathbf{F}$ satisfying

$\mathbf{B}xyz = x(yz)$	Composition, Cut
$\mathbf{C}xyz = (xz)y$	Exchange
$\mathbf{I}x = x$	Identity
$\mathbf{K}x\,!\,y = x$	Weakening
$\mathbf{W}x\,!\,y = x\,!\,y\,!\,y$	Contraction
$\mathbf{D}\,!\,x = x$	Dereliction
$\boldsymbol{\delta}\,!\,x = !\,!\,x$	Comultiplication
$\mathbf{F}\,!\,x\,!\,y = !(xy)$	Monoidal functoriality

Here: $\cdot$ associates to the left; $\cdot$ is suppressed; and $!$ binds stronger than $\cdot$ does.

Linear Combinatory Algebra (LCA)

What
we want (outcome)



Defn. (LCA)
A *linear combinatory algebra (LCA)* is a set $\boldsymbol{A}$ equipped with

- a binary operator (called an *applicative structure*)

$$\cdot \;:\; \boldsymbol{A}^2 \longrightarrow \boldsymbol{A}$$

- a unary operator

$$! \;:\; \boldsymbol{A} \longrightarrow \boldsymbol{A}$$

- (*combinators*) distinguished elements $\mathsf{B}, \mathsf{C}, \mathsf{I}, \mathsf{K}, \mathsf{W}, \mathsf{D}, \delta, \mathsf{F}$ satisfying

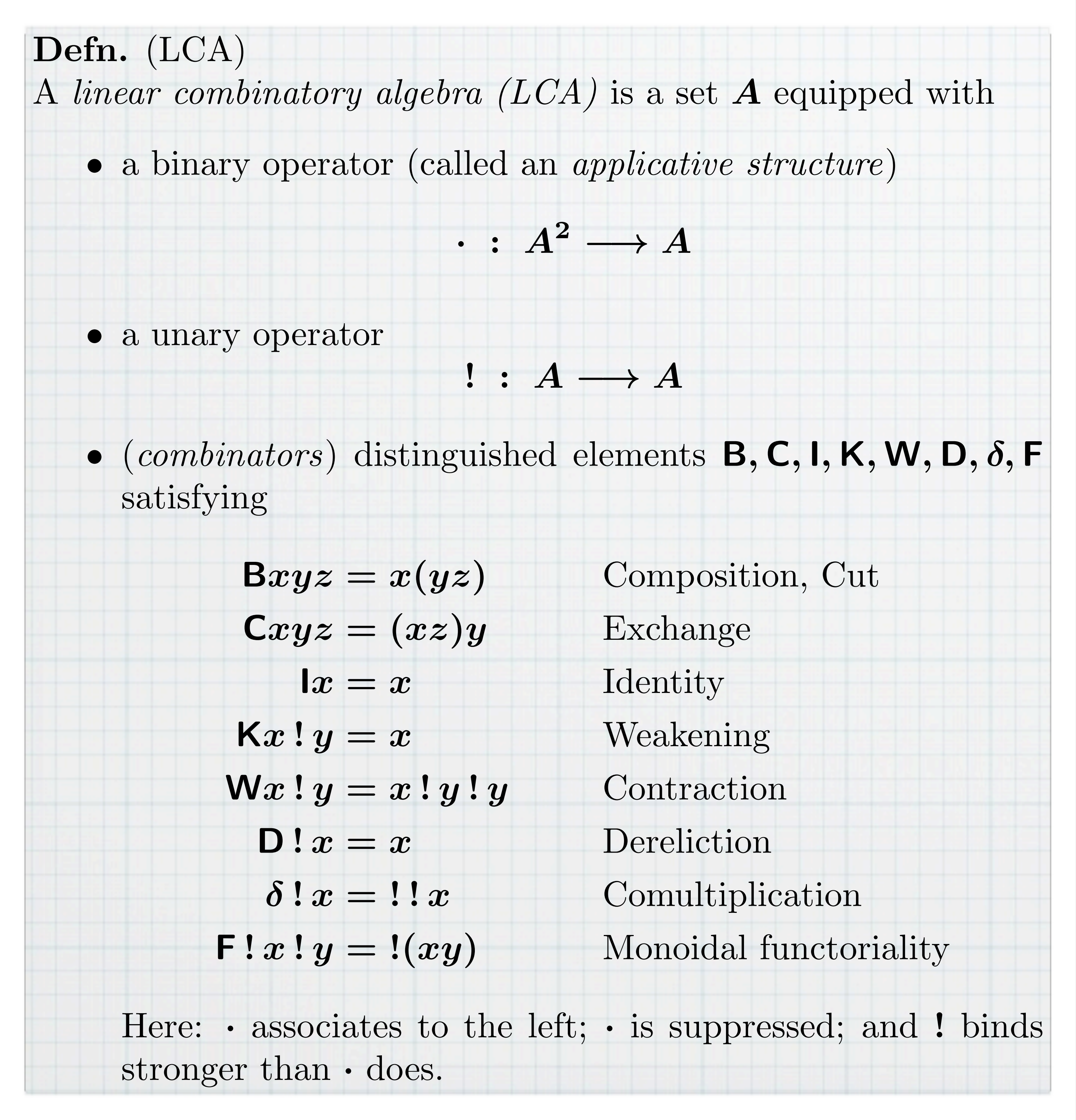
$\mathsf{B}xyz = x(yz)$	Composition, Cut
$\mathsf{C}xyz = (xz)y$	Exchange
$\mathsf{I}x = x$	Identity
$\mathsf{K}x\,!\,y = x$	Weakening
$\mathsf{W}x\,!\,y = x\,!\,y\,!\,y$	Contraction
$\mathsf{D}\,!\,x = x$	Dereliction
$\delta\,!\,x = !\,!\,x$	Comultiplication
$\mathsf{F}\,!\,x\,!\,y = !(xy)$	Monoidal functoriality

Here: $\cdot$ associates to the left; $\cdot$ is suppressed; and $!$ binds stronger than $\cdot$ does.

Linear Combinatory Algebra (LCA)

What
we want (outcome)

- Model of untyped linear λ



Defn. (LCA)
A *linear combinatory algebra (LCA)* is a set $\boldsymbol{A}$ equipped with

- a binary operator (called an *applicative structure*)

$$\cdot \;:\; \boldsymbol{A}^2 \longrightarrow \boldsymbol{A}$$

- a unary operator

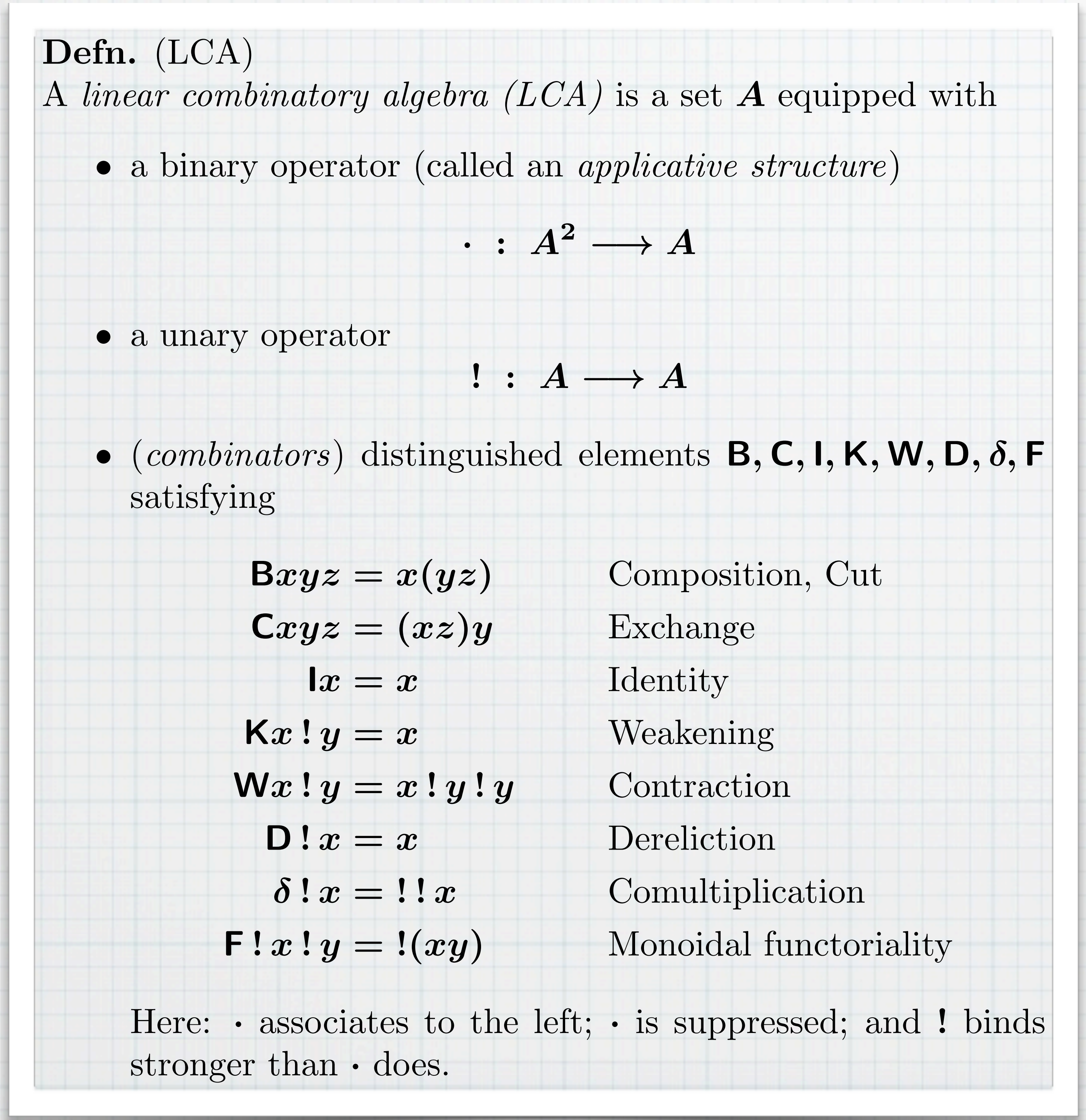
$$! \;:\; \boldsymbol{A} \longrightarrow \boldsymbol{A}$$

- (*combinators*) distinguished elements $\mathsf{B}, \mathsf{C}, \mathsf{I}, \mathsf{K}, \mathsf{W}, \mathsf{D}, \delta, \mathsf{F}$ satisfying

$\mathsf{B}xyz = x(yz)$	Composition, Cut
$\mathsf{C}xyz = (xz)y$	Exchange
$\mathsf{I}x = x$	Identity
$\mathsf{K}x\,!\,y = x$	Weakening
$\mathsf{W}x\,!\,y = x\,!\,y\,!\,y$	Contraction
$\mathsf{D}\,!\,x = x$	Dereliction
$\delta\,!\,x = !\,!\,x$	Comultiplication
$\mathsf{F}\,!\,x\,!\,y = !(xy)$	Monoidal functoriality

Here: $\cdot$ associates to the left; $\cdot$ is suppressed; and $!$ binds stronger than $\cdot$ does.

Linear Combinatory Algebra (LCA)



Defn. (LCA)
A *linear combinatory algebra (LCA)* is a set $\boldsymbol{A}$ equipped with

- a binary operator (called an *applicative structure*)

$$\cdot \;:\; \boldsymbol{A}^2 \longrightarrow \boldsymbol{A}$$

- a unary operator

$$! \;:\; \boldsymbol{A} \longrightarrow \boldsymbol{A}$$

- (*combinators*) distinguished elements $\mathsf{B}, \mathsf{C}, \mathsf{I}, \mathsf{K}, \mathsf{W}, \mathsf{D}, \delta, \mathsf{F}$ satisfying

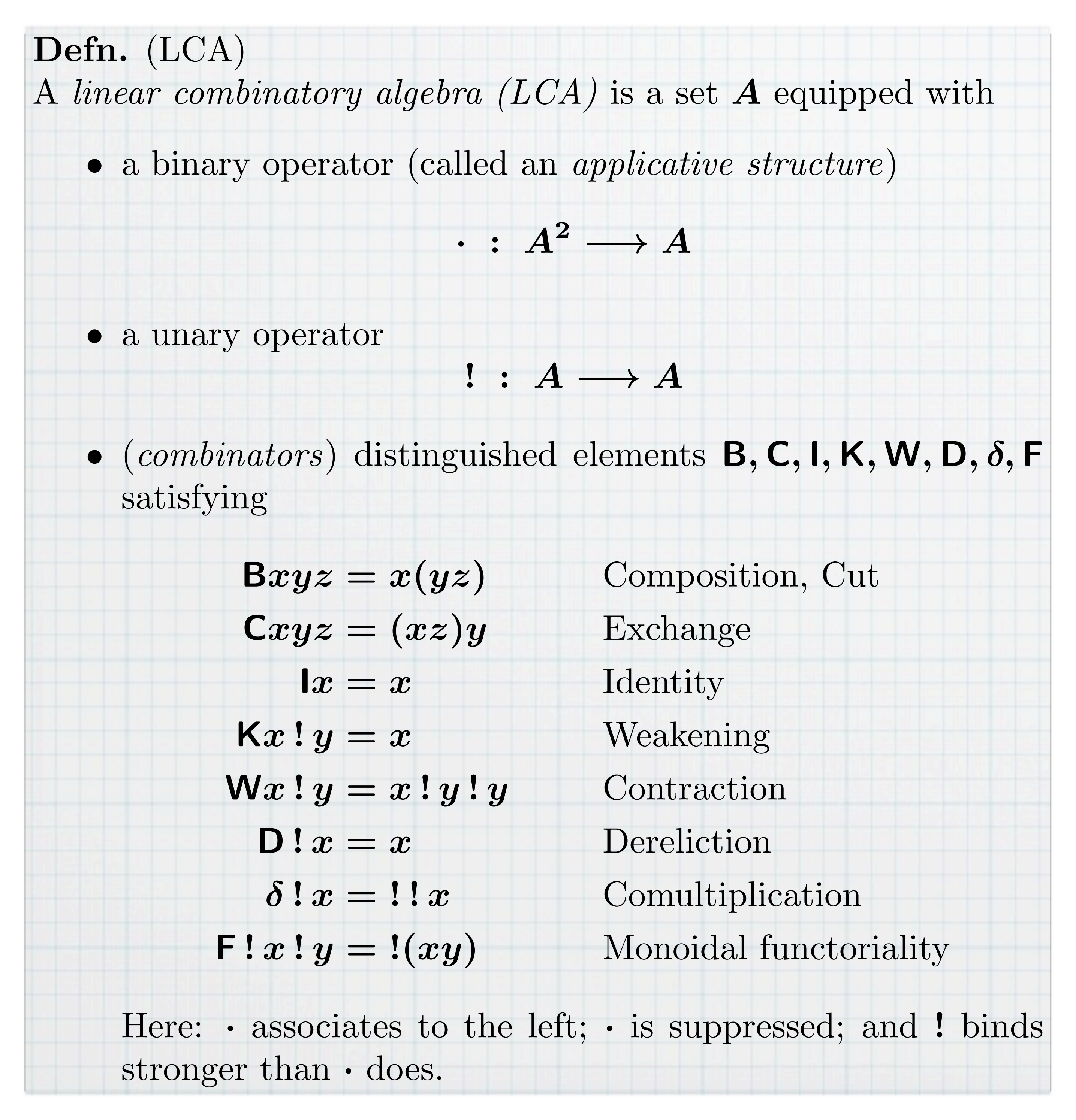
$\mathsf{B}xyz = x(yz)$	Composition, Cut
$\mathsf{C}xyz = (xz)y$	Exchange
$\mathsf{I}x = x$	Identity
$\mathsf{K}x\,!\,y = x$	Weakening
$\mathsf{W}x\,!\,y = x\,!\,y\,!\,y$	Contraction
$\mathsf{D}\,!\,x = x$	Dereliction
$\delta\,!\,x = !\,!\,x$	Comultiplication
$\mathsf{F}\,!\,x\,!\,y = !(xy)$	Monoidal functoriality

Here: $\cdot$ associates to the left; $\cdot$ is suppressed; and $!$ binds stronger than $\cdot$ does.

What
we want (outcome)

- ✱ Model of untyped linear λ
- ✱ a ∈ A ≈ closed linear λ-term

Linear Combinatory Algebra (LCA)



Defn. (LCA)
A *linear combinatory algebra (LCA)* is a set $\boldsymbol{A}$ equipped with

- a binary operator (called an *applicative structure*)

$$\cdot \ : \ \boldsymbol{A}^2 \longrightarrow \boldsymbol{A}$$

- a unary operator

$$! \ : \ \boldsymbol{A} \longrightarrow \boldsymbol{A}$$

- (*combinators*) distinguished elements $\mathsf{B}, \mathsf{C}, \mathsf{I}, \mathsf{K}, \mathsf{W}, \mathsf{D}, \delta, \mathsf{F}$ satisfying

$\mathsf{B}xyz = x(yz)$	Composition, Cut
$\mathsf{C}xyz = (xz)y$	Exchange
$\mathsf{I}x = x$	Identity
$\mathsf{K}x\,!\,y = x$	Weakening
$\mathsf{W}x\,!\,y = x\,!\,y\,!\,y$	Contraction
$\mathsf{D}\,!\,x = x$	Dereliction
$\delta\,!\,x = !\,!\,x$	Comultiplication
$\mathsf{F}\,!\,x\,!\,y = !(xy)$	Monoidal functoriality

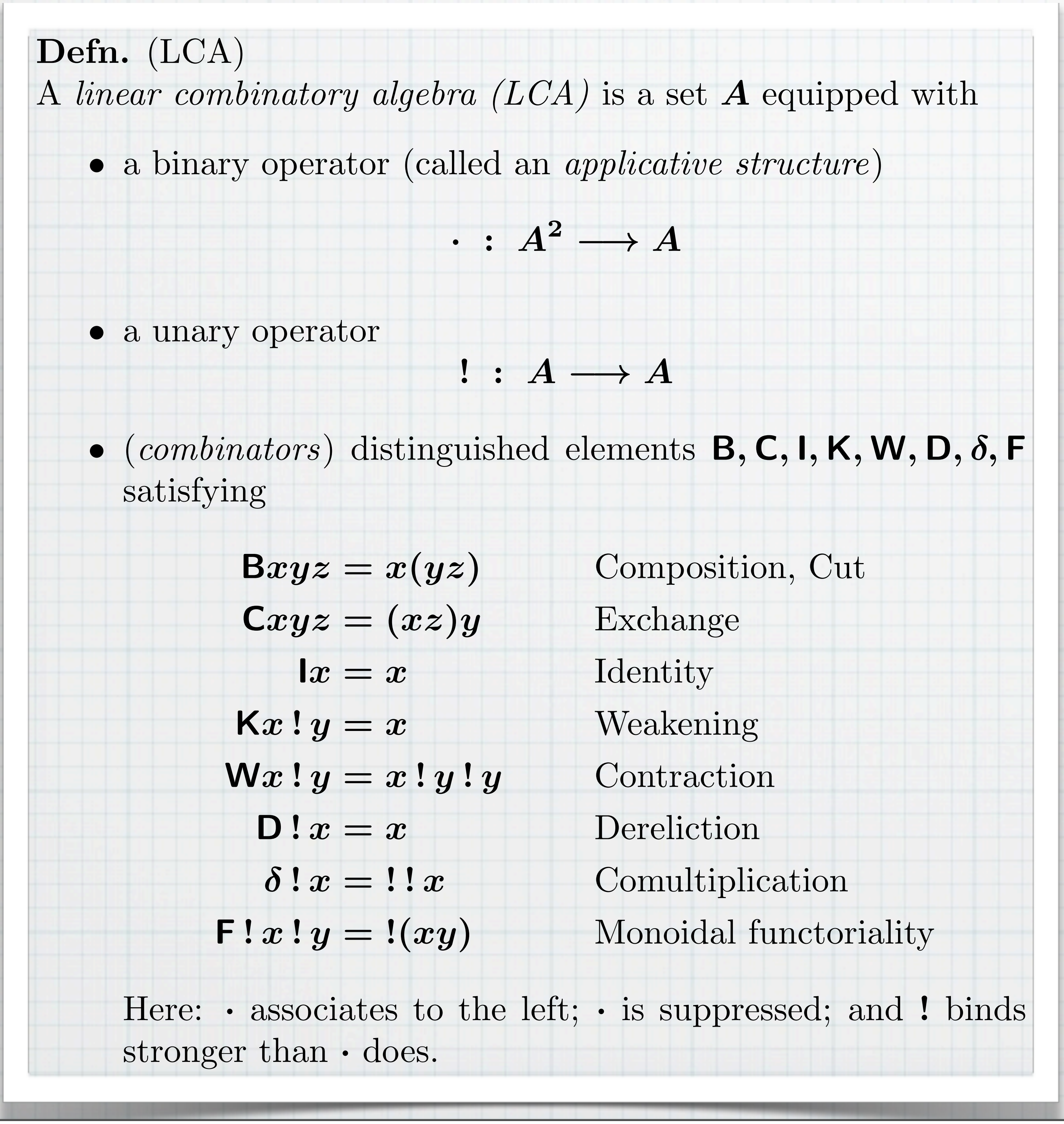
Here: $\cdot$ associates to the left; $\cdot$ is suppressed; and $!$ binds stronger than $\cdot$ does.

What
we want (outcome)

- Model of untyped linear λ
- a ∈ A ≈ closed linear λ-term
- No **S** or **K** (linear!)

Hasuo (Tokyo)

Linear Combinatory Algebra (LCA)



Defn. (LCA)
A *linear combinatory algebra (LCA)* is a set $\boldsymbol{A}$ equipped with

- a binary operator (called an *applicative structure*)

$$\cdot \;:\; \boldsymbol{A}^2 \longrightarrow \boldsymbol{A}$$

- a unary operator

$$! \;:\; \boldsymbol{A} \longrightarrow \boldsymbol{A}$$

- (*combinators*) distinguished elements $\mathsf{B}, \mathsf{C}, \mathsf{I}, \mathsf{K}, \mathsf{W}, \mathsf{D}, \delta, \mathsf{F}$ satisfying

$$\begin{array}{rll}
\mathsf{B}xyz = x(yz) & & \text{Composition, Cut} \\
\mathsf{C}xyz = (xz)y & & \text{Exchange} \\
\mathsf{I}x = x & & \text{Identity} \\
\mathsf{K}x\,!\,y = x & & \text{Weakening} \\
\mathsf{W}x\,!\,y = x\,!\,y\,!\,y & & \text{Contraction} \\
\mathsf{D}\,!\,x = x & & \text{Dereliction} \\
\delta\,!\,x = !\,!\,x & & \text{Comultiplication} \\
\mathsf{F}\,!\,x\,!\,y = !(xy) & & \text{Monoidal functoriality}
\end{array}$$

Here: $\cdot$ associates to the left; $\cdot$ is suppressed; and $!$ binds stronger than $\cdot$ does.

What
we want (outcome)

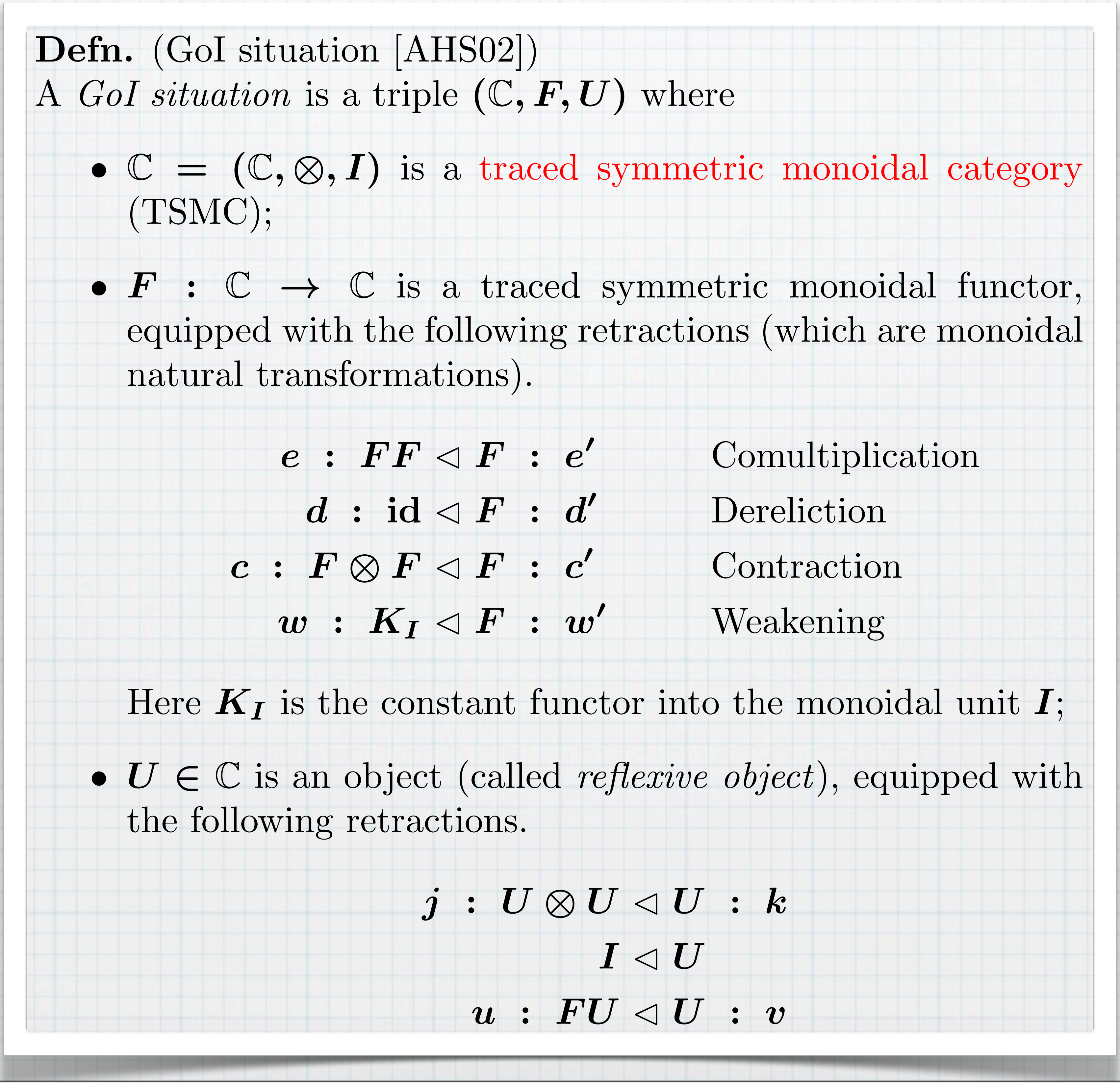
- Model of untyped linear λ
- a ∈ A ≈ closed linear λ-term
- No **S** or **K** (linear!)
- Combinatory completeness: e.g.

$$\lambda xyz.\, zxy$$

designates an elem. of A

What we use (ingredient)

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

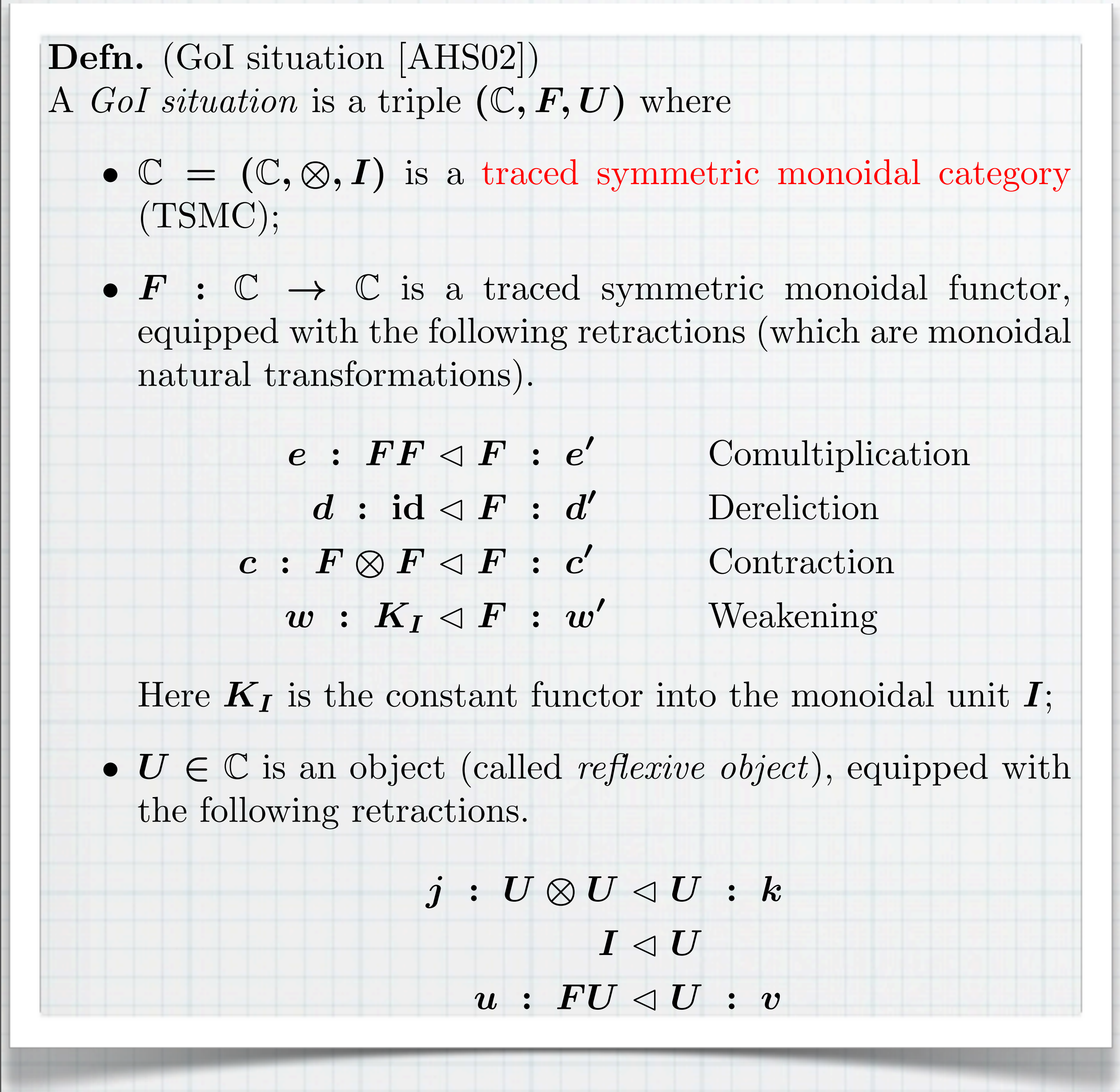
  $$\begin{aligned} e &: FF \lhd F : e' && \text{Comultiplication} \\ d &: \mathrm{id} \lhd F : d' && \text{Dereliction} \\ c &: F \otimes F \lhd F : c' && \text{Contraction} \\ w &: K_I \lhd F : w' && \text{Weakening} \end{aligned}$$

  Here K_I is the constant functor into the monoidal unit I;
- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

  $$\begin{aligned} j &: U \otimes U \lhd U : k \\ & I \lhd U \\ u &: FU \lhd U : v \end{aligned}$$

Hasuo (Tokyo)

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

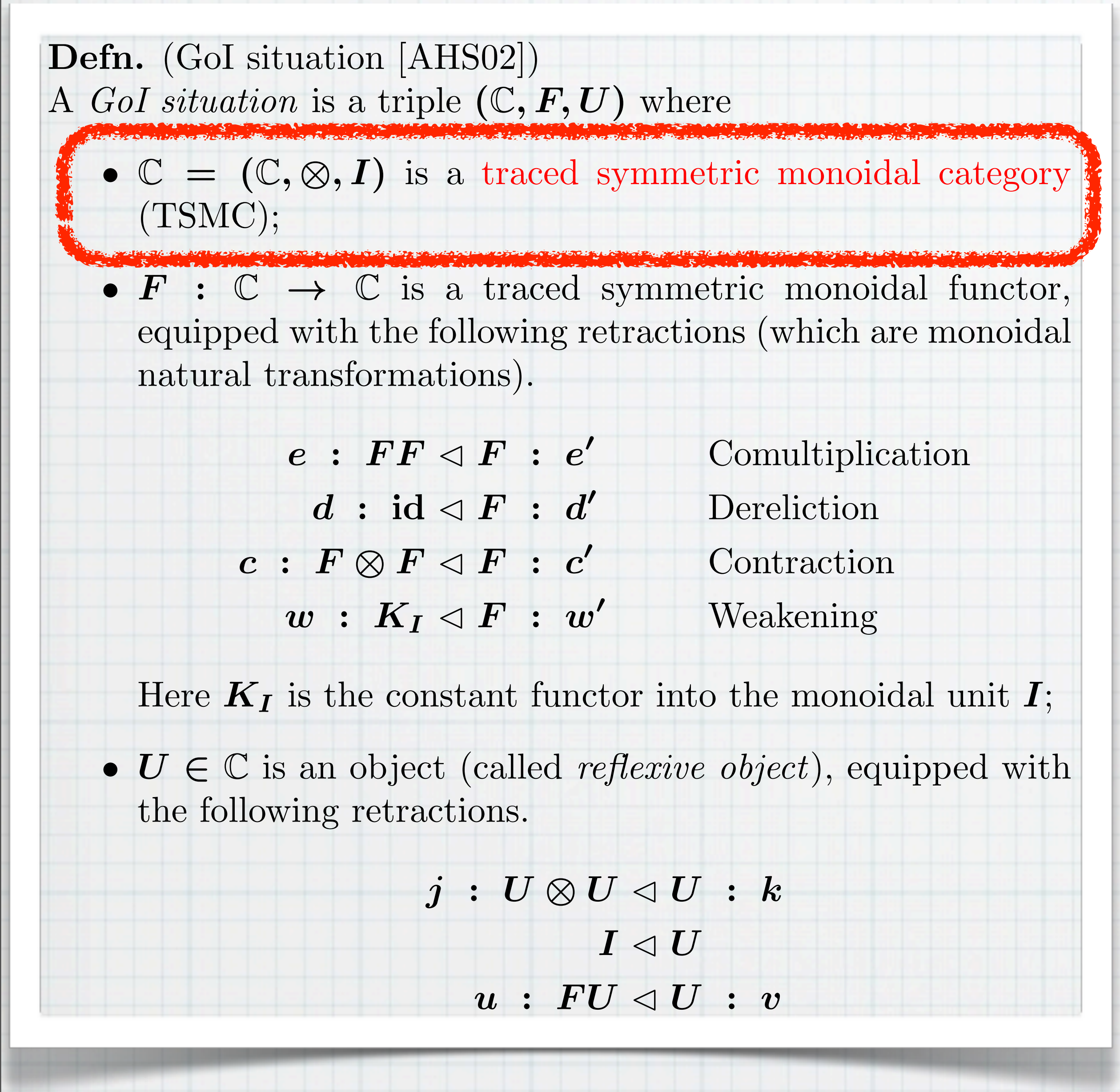
$$\begin{array}{rcll} \boldsymbol{e} : & \boldsymbol{F}\boldsymbol{F} \lhd \boldsymbol{F} & : \boldsymbol{e}' & \text{Comultiplication} \\ \boldsymbol{d} : & \mathbf{id} \lhd \boldsymbol{F} & : \boldsymbol{d}' & \text{Dereliction} \\ \boldsymbol{c} : & \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} & : \boldsymbol{c}' & \text{Contraction} \\ \boldsymbol{w} : & \boldsymbol{K_I} \lhd \boldsymbol{F} & : \boldsymbol{w}' & \text{Weakening} \end{array}$$

  Here $\boldsymbol{K_I}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;
- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{array}{rcl} \boldsymbol{j} : & \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} & : \boldsymbol{k} \\ & \boldsymbol{I} \lhd \boldsymbol{U} & \\ \boldsymbol{u} : & \boldsymbol{F}\boldsymbol{U} \lhd \boldsymbol{U} & : \boldsymbol{v} \end{array}$$

- Monoidal category $(\mathbb{C}, \otimes, \boldsymbol{I})$
  - String diagrams

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$$\begin{array}{rcll} e & : & FF \lhd F : e' & \text{Comultiplication} \\ d & : & \mathrm{id} \lhd F : d' & \text{Dereliction} \\ c & : & F \otimes F \lhd F : c' & \text{Contraction} \\ w & : & K_I \lhd F : w' & \text{Weakening} \end{array}$$

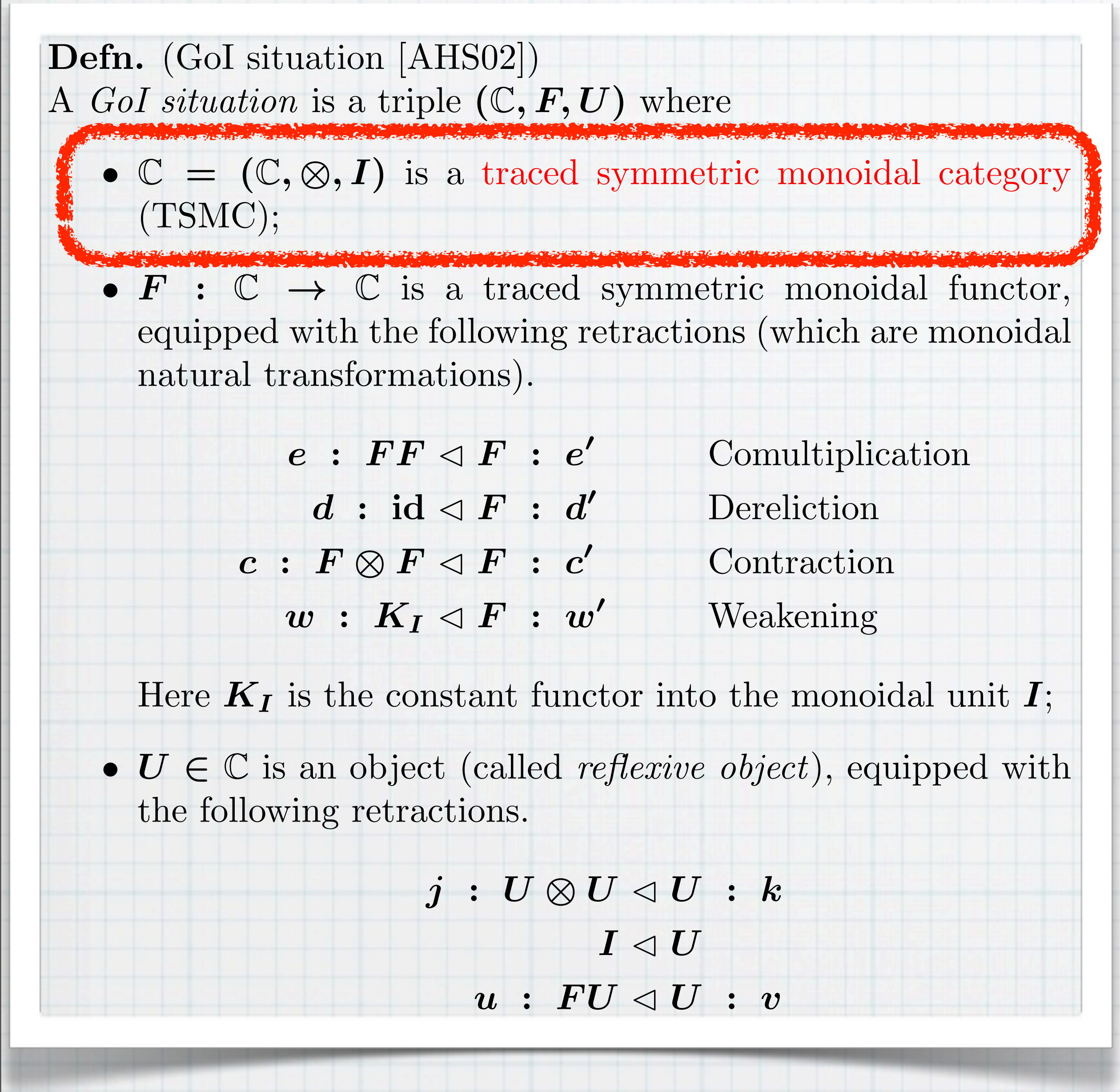
Here K_I is the constant functor into the monoidal unit I;

- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{array}{c} j : U \otimes U \lhd U : k \\ I \lhd U \\ u : FU \lhd U : v \end{array}$$

- Monoidal category $(\mathbb{C}, \otimes, I)$
  - String diagrams

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$$\begin{array}{rcl l}
e : \boldsymbol{F}\boldsymbol{F} \lhd \boldsymbol{F} : e' & & & \text{Comultiplication} \\
d : \mathrm{id} \lhd \boldsymbol{F} : d' & & & \text{Dereliction} \\
c : \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : c' & & & \text{Contraction} \\
w : \boldsymbol{K}_{\boldsymbol{I}} \lhd \boldsymbol{F} : w' & & & \text{Weakening}
\end{array}$$

  Here $\boldsymbol{K}_{\boldsymbol{I}}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;

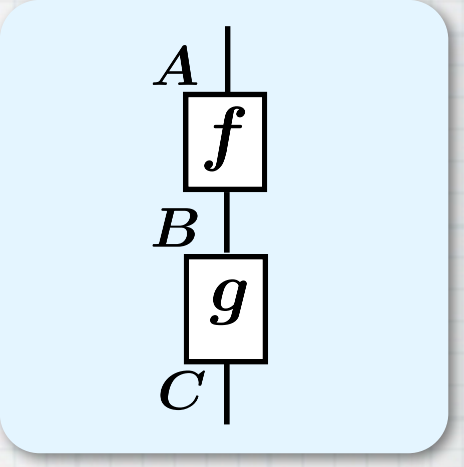
- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$j : \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} : k$$
$$\boldsymbol{I} \lhd \boldsymbol{U}$$
$$u : \boldsymbol{F}\boldsymbol{U} \lhd \boldsymbol{U} : v$$

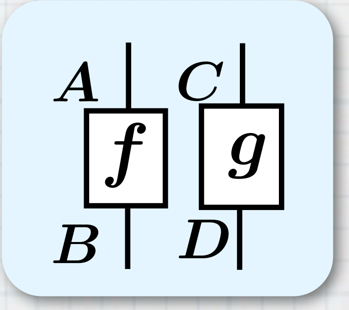
* Monoidal category $(\mathbb{C}, \otimes, \boldsymbol{I})$

* String diagrams

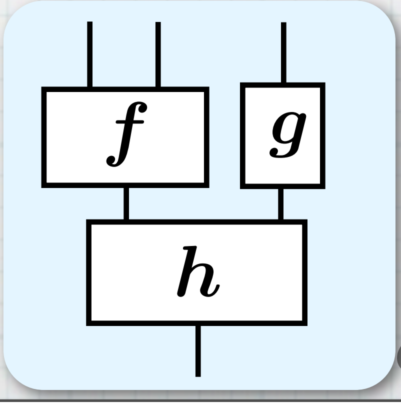
$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{g \circ f} C}$$



$$\frac{A \xrightarrow{f} B \quad C \xrightarrow{g} D}{A \otimes C \xrightarrow{f \otimes g} B \otimes D}$$

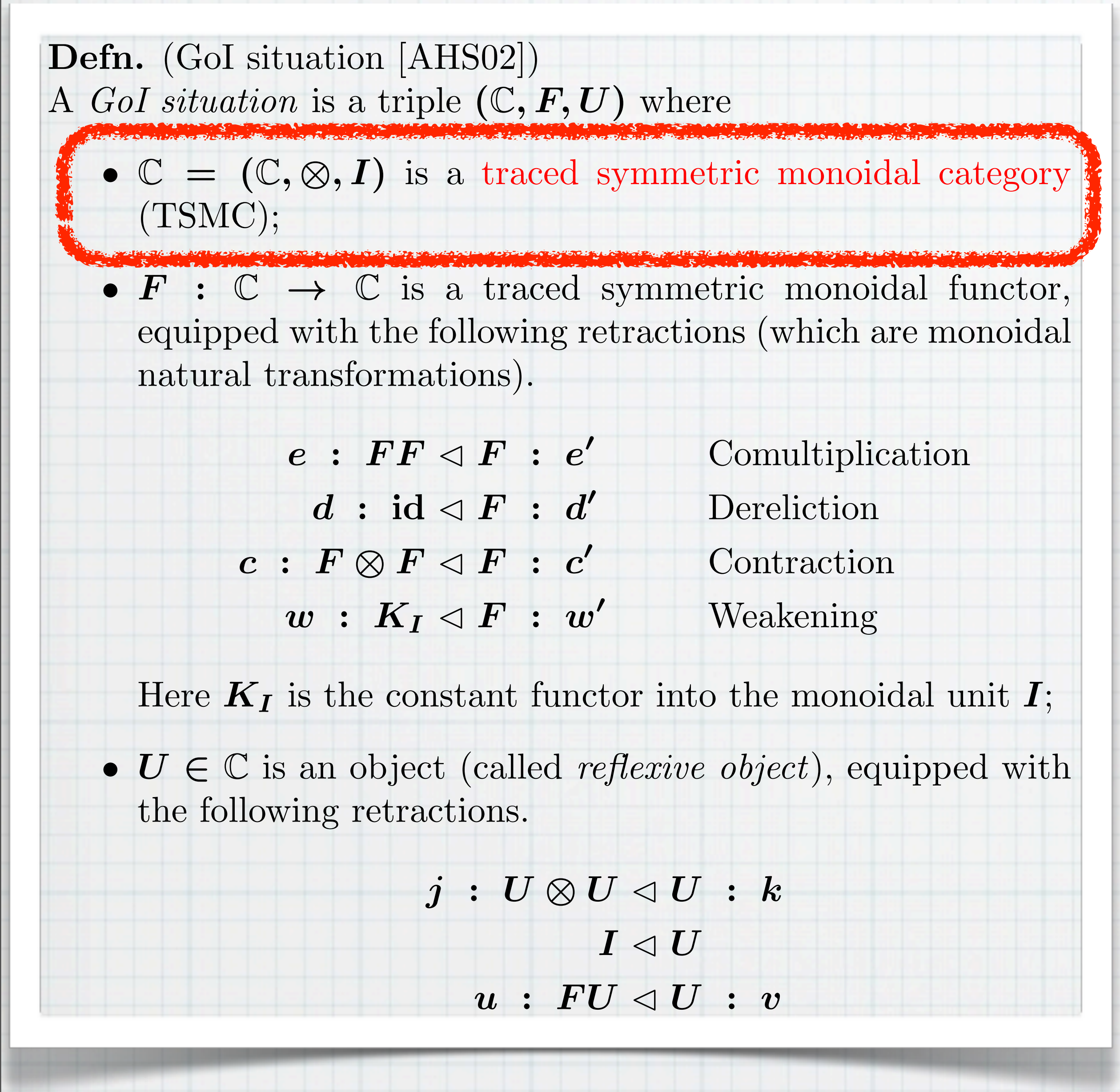


$$h \circ (f \otimes g)$$



o)

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$$\begin{array}{rcl}
e : FF \lhd F : e' & & \text{Comultiplication} \\
d : \mathrm{id} \lhd F : d' & & \text{Dereliction} \\
c : F \otimes F \lhd F : c' & & \text{Contraction} \\
w : K_I \lhd F : w' & & \text{Weakening}
\end{array}$$

Here $\boldsymbol{K_I}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;

- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

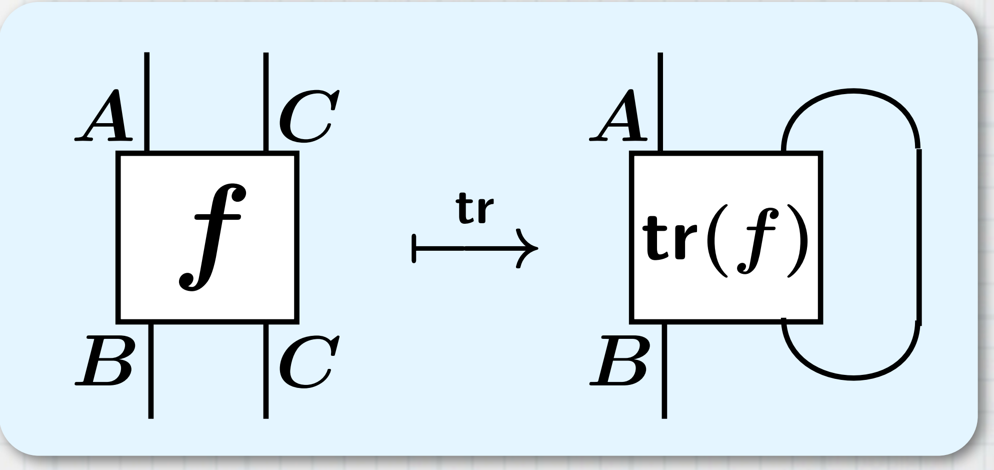
$$\begin{array}{c}
j : U \otimes U \lhd U : k \\
I \lhd U \\
u : FU \lhd U : v
\end{array}$$

- Traced monoidal category

  - “feedback”

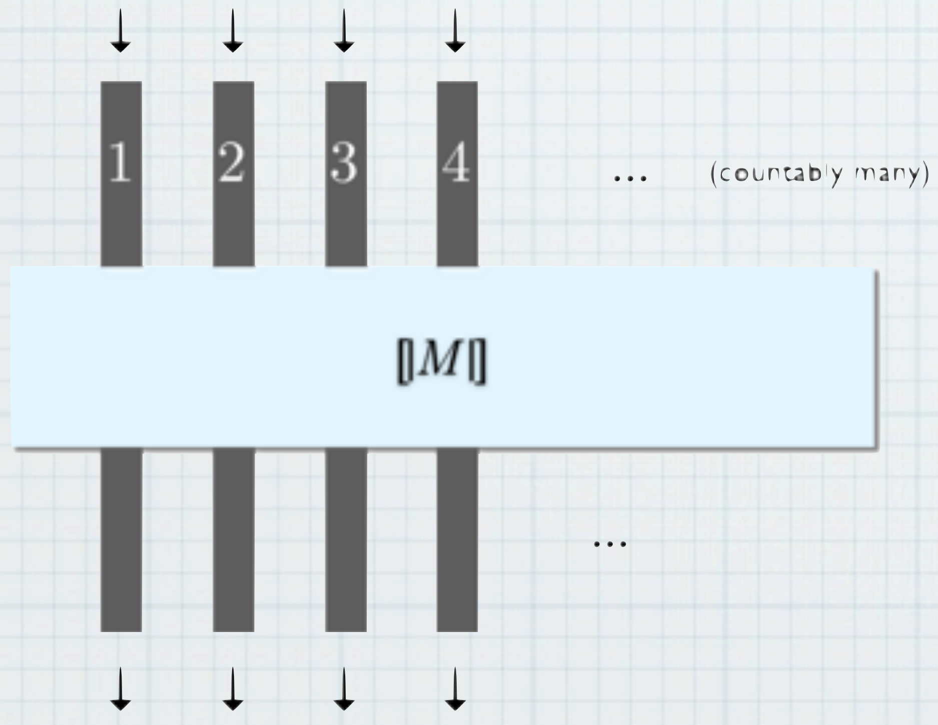
$$\frac{A \otimes C \xrightarrow{f} B \otimes C}{A \xrightarrow{\mathrm{tr}(f)} B}$$

that is

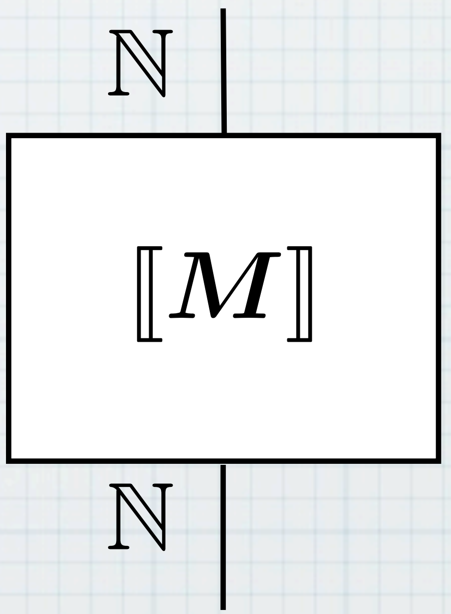


String Diagram vs. “Pipe Diagram”

- I use two ways of depicting partial functions $\mathbb{N} \rightharpoonup \mathbb{N}$



Pipe diagram

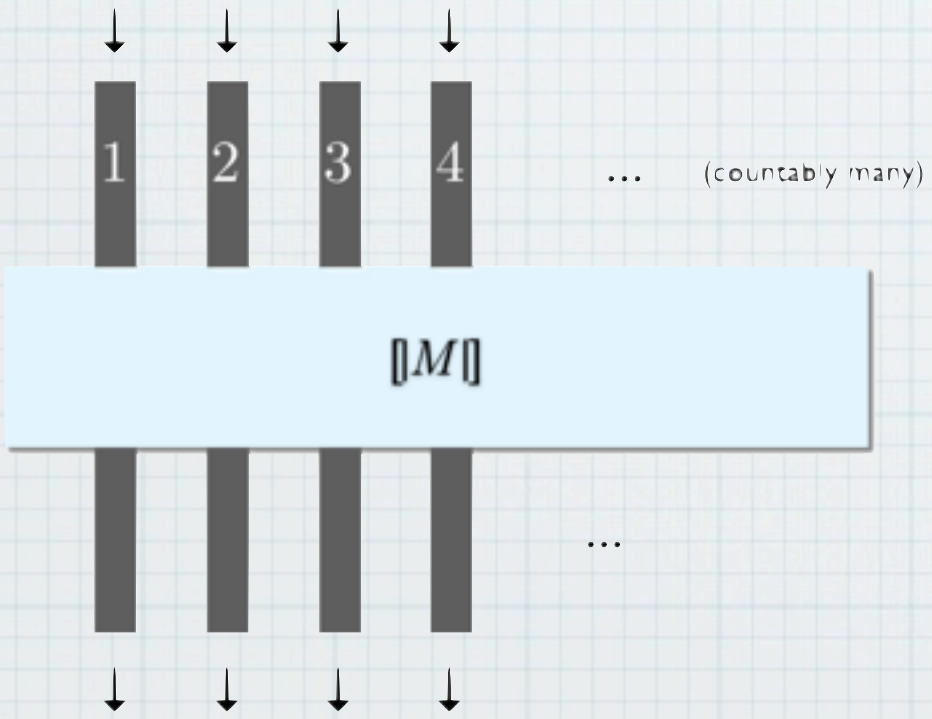


String diagram

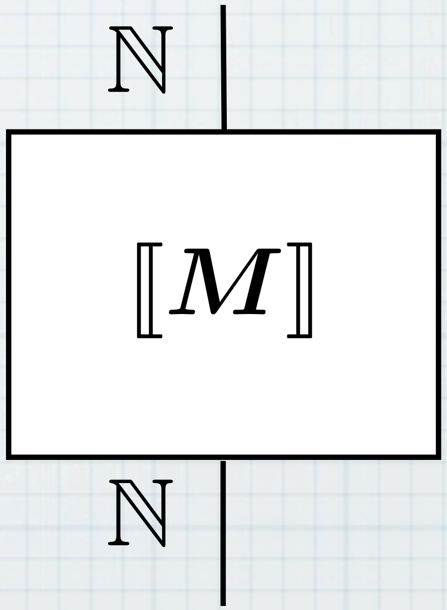
String Diagram vs. "Pipe Diagram"

- I use two ways of depicting partial functions $\mathbb{N} \rightharpoonup \mathbb{N}$

In the monoidal category $(\mathbf{Pfn}, +, 0)$



Pipe diagram

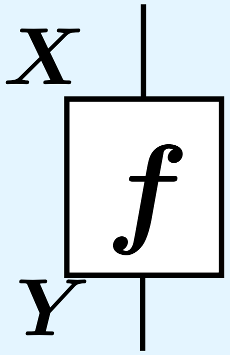


String diagram

Traced Sym. Monoidal Category $(\mathbf{Pfn}, +, 0)$

- Category **Pfn** of partial functions
  - Obj. A set X
  - Arr. A partial function

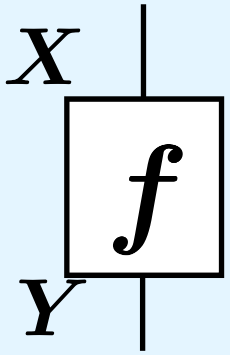
$$\begin{array}{c} X \to Y \ \text{ in } \mathbf{Pfn} \\ \hline\hline X \rightharpoonup Y, \text{ partial function} \end{array}$$



Traced Sym. Monoidal Category $(\mathbf{Pfn}, +, 0)$

- Category $\mathbf{Pfn}$ of partial functions
  - Obj. A set X
  - Arr. A partial function

$$\frac{\boldsymbol{X} \to \boldsymbol{Y} \ \text{ in } \mathbf{Pfn}}{\boldsymbol{X} \rightharpoonup \boldsymbol{Y}, \text{ partial function}}$$



- is traced symmetric monoidal

Traced Sym. Monoidal Category

$(\mathbf{Pfn}, +, 0)$

- $$\frac{X + Z \xrightarrow{f} Y + Z \qquad \text{in } \mathbf{Pfn}}{X \xrightarrow{\mathsf{tr}(f)} Y \qquad \text{in } \mathbf{Pfn}}$$

  How?

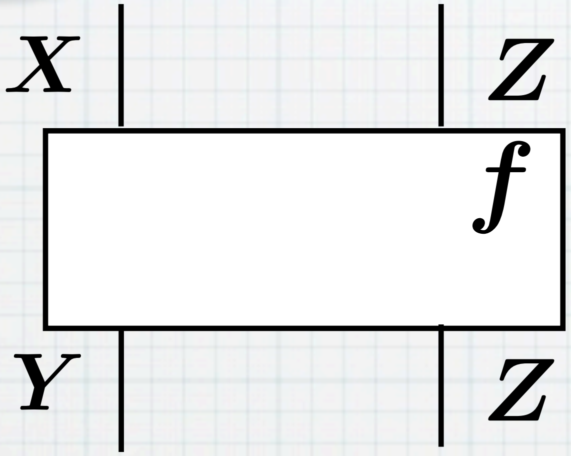
-

Traced Sym. Monoidal Category

$(\mathbf{Pfn}, +, 0)$

- $$\frac{X + Z \xrightarrow{f} Y + Z \quad \text{in } \mathbf{Pfn}}{X \xrightarrow{\mathsf{tr}(f)} Y \quad \text{in } \mathbf{Pfn}}$$

How?

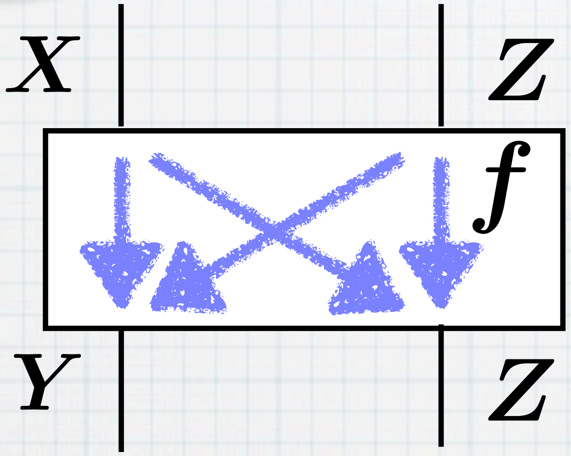
- 

Traced Sym. Monoidal Category

$$(\mathbf{Pfn}, +, 0)$$

- $$\frac{X + Z \xrightarrow{f} Y + Z \quad \text{in } \mathbf{Pfn}}{X \xrightarrow{\mathbf{tr}(f)} Y \quad \text{in } \mathbf{Pfn}}$$

How?

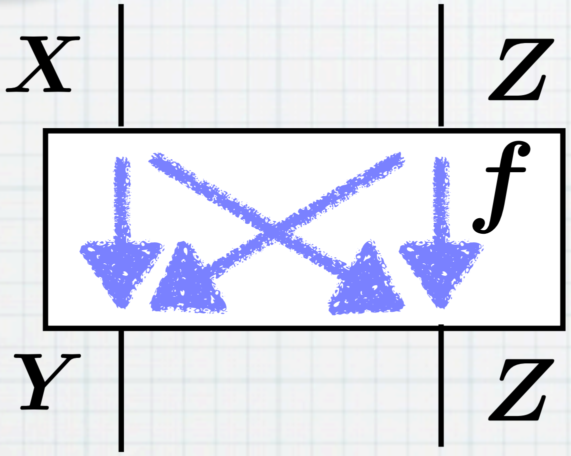
- 

Traced Sym. Monoidal Category

$(\mathbf{Pfn}, +, 0)$

- $$\frac{X + Z \xrightarrow{f} Y + Z \quad \text{in } \mathbf{Pfn}}{X \xrightarrow{\mathrm{tr}(f)} Y \quad \text{in } \mathbf{Pfn}}$$

How?

- 

$$f_{XY}(x) := \begin{cases} f(x) & \text{if } f(x) \in Y \\ \bot & \text{o.w.} \end{cases}$$

Similar for f_{XZ}, f_{ZY}, f_{ZZ}

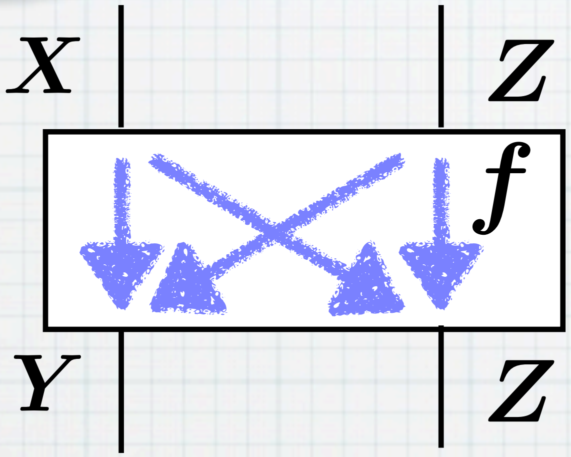
Hasuo (Tokyo)

Traced Sym. Monoidal Category

$(\mathbf{Pfn}, +, 0)$

- $$\frac{X+Z \xrightarrow{f} Y+Z \qquad \text{in } \mathbf{Pfn}}{X \xrightarrow{\mathbf{tr}(f)} Y \qquad \text{in } \mathbf{Pfn}}$$

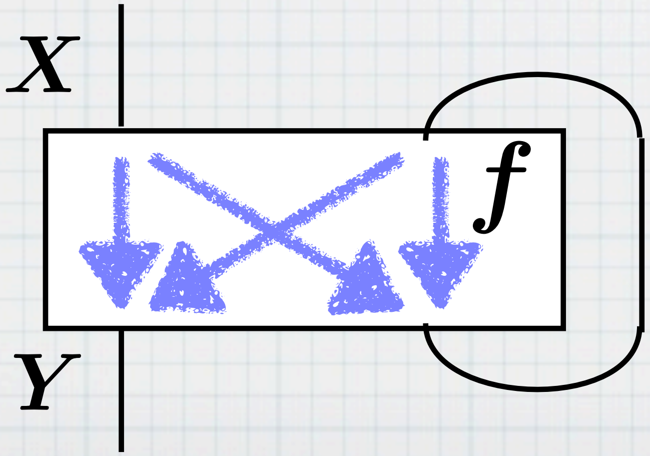
How?

- 

$$f_{XY}(x) := \begin{cases} f(x) & \text{if } f(x) \in Y \\ \bot & \text{o.w.} \end{cases}$$

Similar for f_{XZ}, f_{ZY}, f_{ZZ}

- Trace operator:

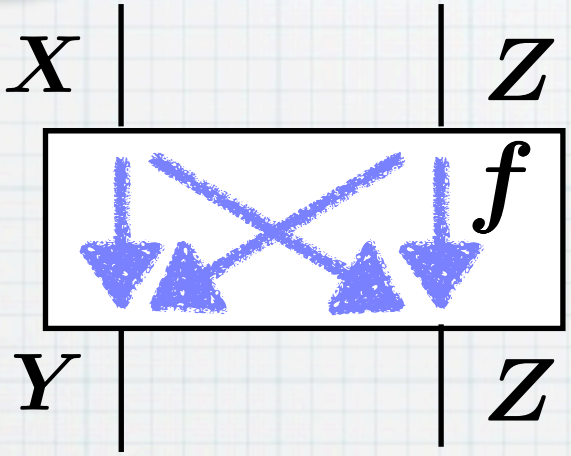


Traced Sym. Monoidal Category

$(\mathbf{Pfn}, +, 0)$

- $$\frac{X + Z \xrightarrow{f} Y + Z \quad \text{in } \mathbf{Pfn}}{X \xrightarrow{\mathsf{tr}(f)} Y \quad \text{in } \mathbf{Pfn}}$$

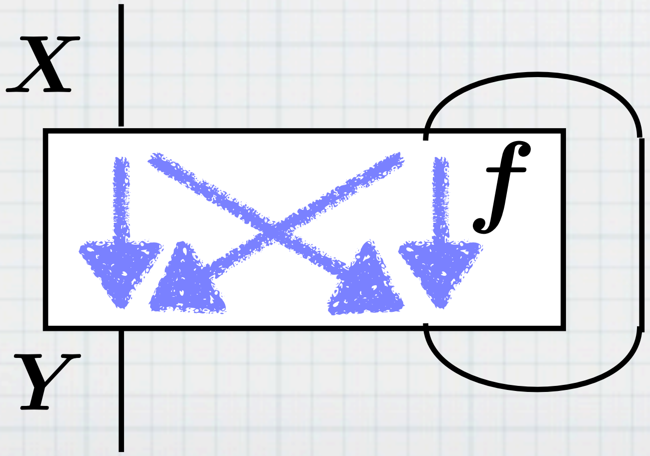
How?

- 

$$f_{XY}(x) := \begin{cases} f(x) & \text{if } f(x) \in Y \\ \perp & \text{o.w.} \end{cases}$$

Similar for f_{XZ}, f_{ZY}, f_{ZZ}

- Trace operator:



$$\mathsf{tr}(f) =$$

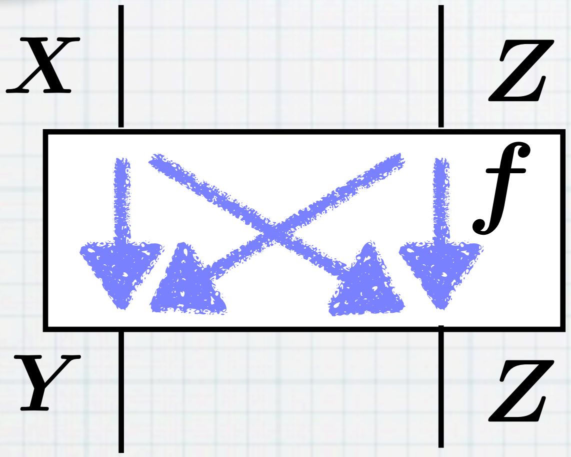
$$f_{XY} \sqcup \left(\coprod_{n \in \mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \right)$$

Traced Sym. Monoidal Category

$(\mathbf{Pfn}, +, 0)$

- $$\frac{X + Z \xrightarrow{f} Y + Z \qquad \text{in } \mathbf{Pfn}}{X \xrightarrow{\mathsf{tr}(f)} Y \qquad \text{in } \mathbf{Pfn}}$$

How?

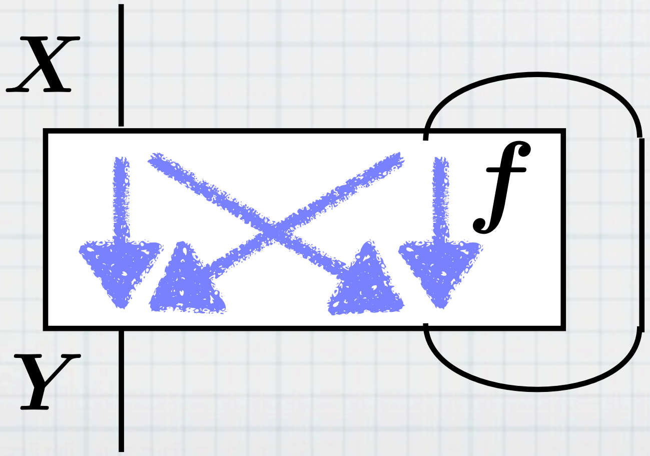
- 

$$f_{XY}(x) := \begin{cases} f(x) & \text{if } f(x) \in Y \\ \perp & \text{o.w.} \end{cases}$$

Similar for f_{XZ}, f_{ZY}, f_{ZZ}

- **Execution formula** (Girard)
- Partiality is essential (infinite loop)

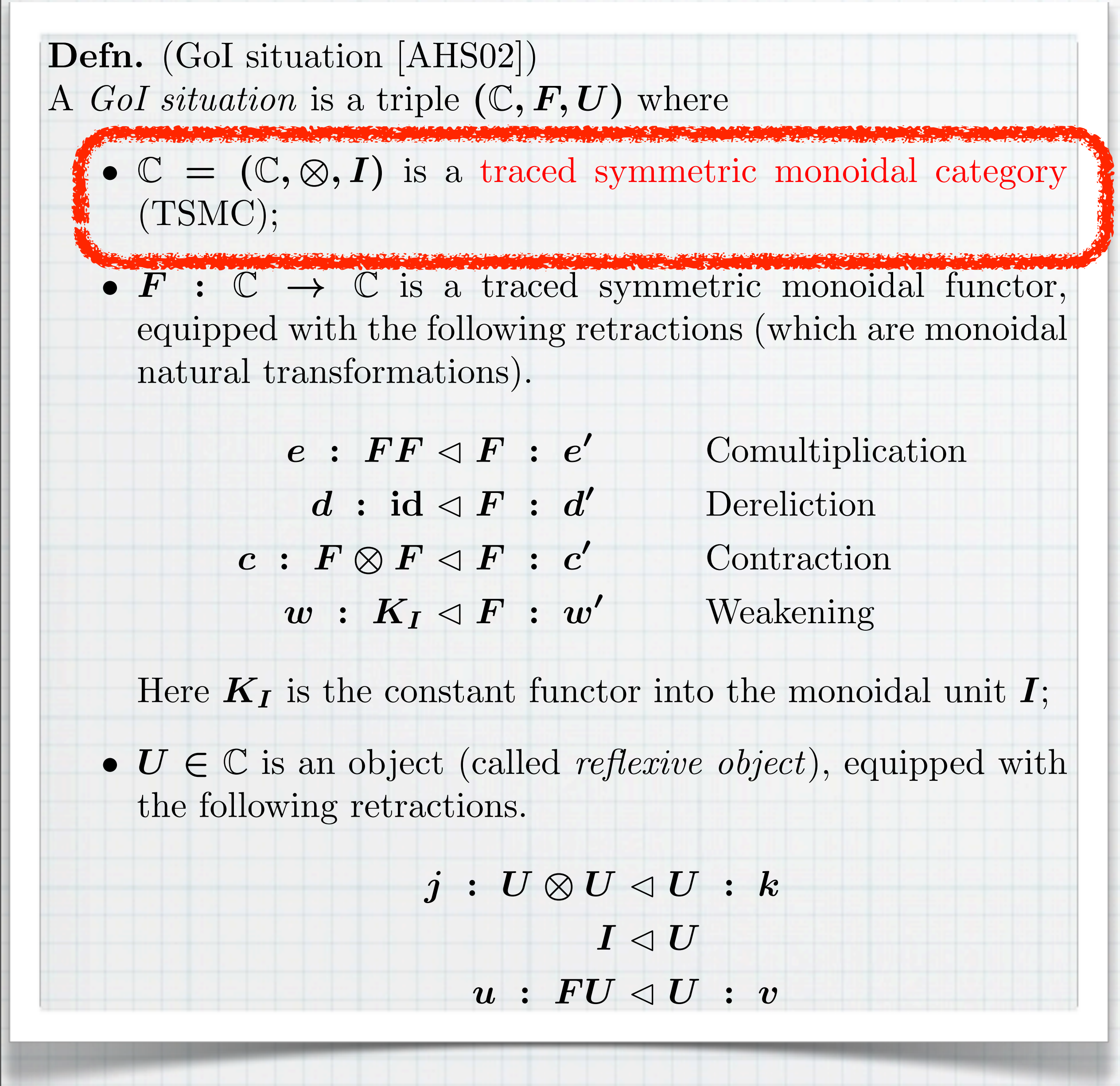
- Trace operator:



$$\mathsf{tr}(f) =$$

$$f_{XY} \sqcup \left(\coprod_{n \in \mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \right)$$

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ where

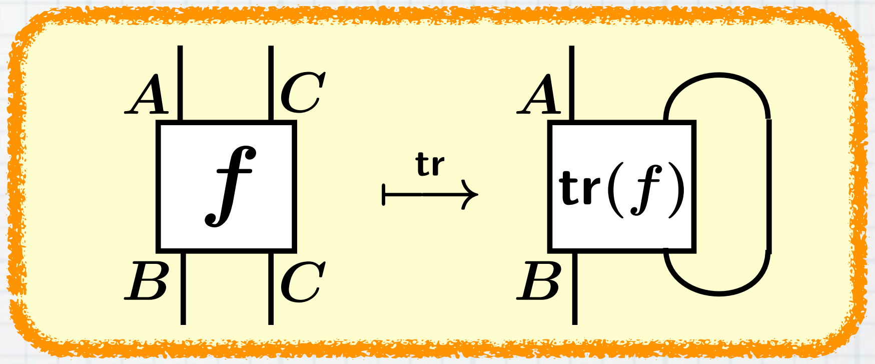
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$$\begin{array}{rl}
e : FF \lhd F : e' & \text{Comultiplication} \\
d : \mathrm{id} \lhd F : d' & \text{Dereliction} \\
c : F \otimes F \lhd F : c' & \text{Contraction} \\
w : K_I \lhd F : w' & \text{Weakening}
\end{array}$$

  Here K_I is the constant functor into the monoidal unit I;
- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{array}{c}
j : U \otimes U \lhd U : k \\
I \lhd U \\
u : FU \lhd U : v
\end{array}$$

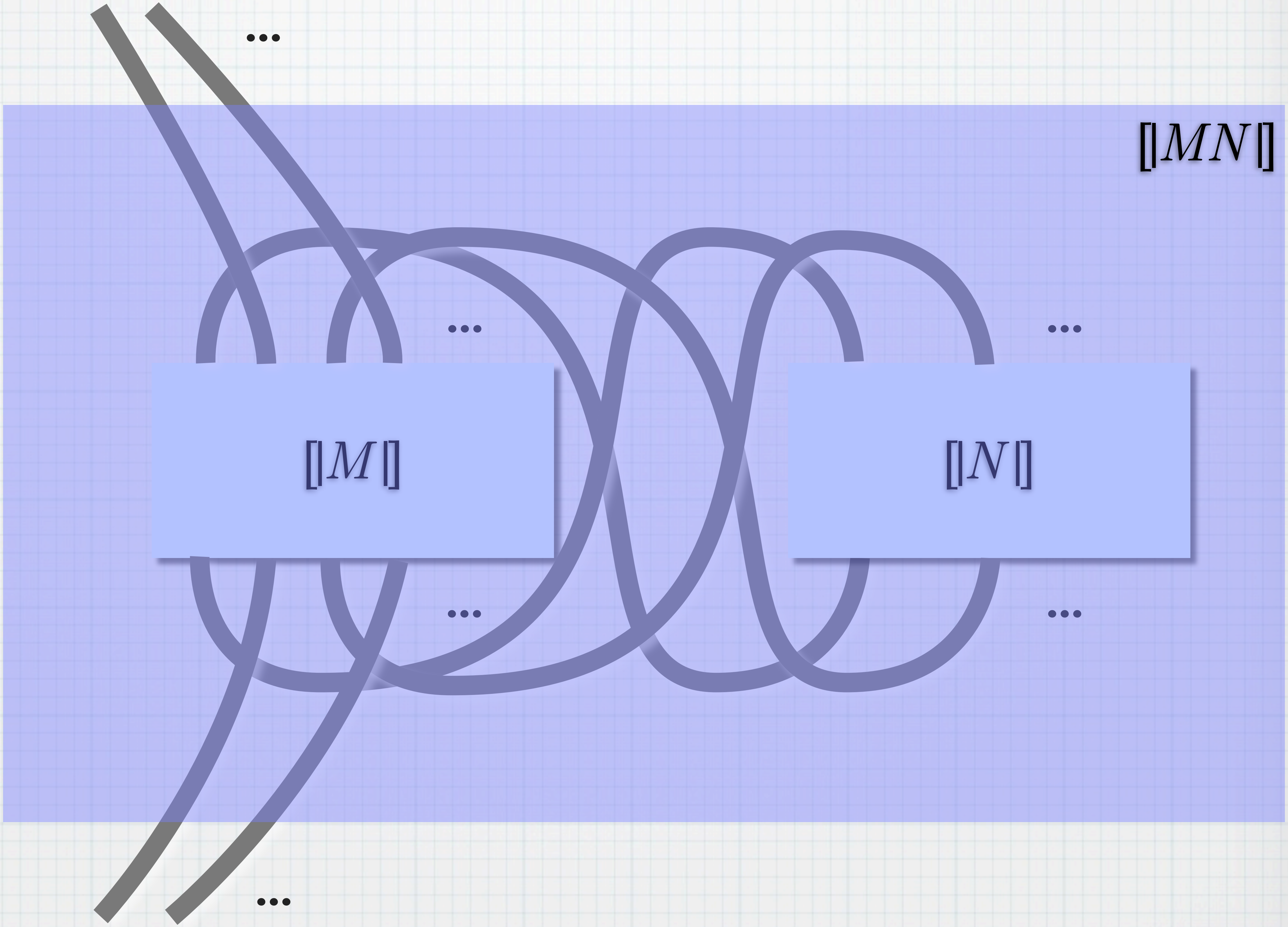
- Traced sym. monoidal cat.
  - Where one can "feedback"



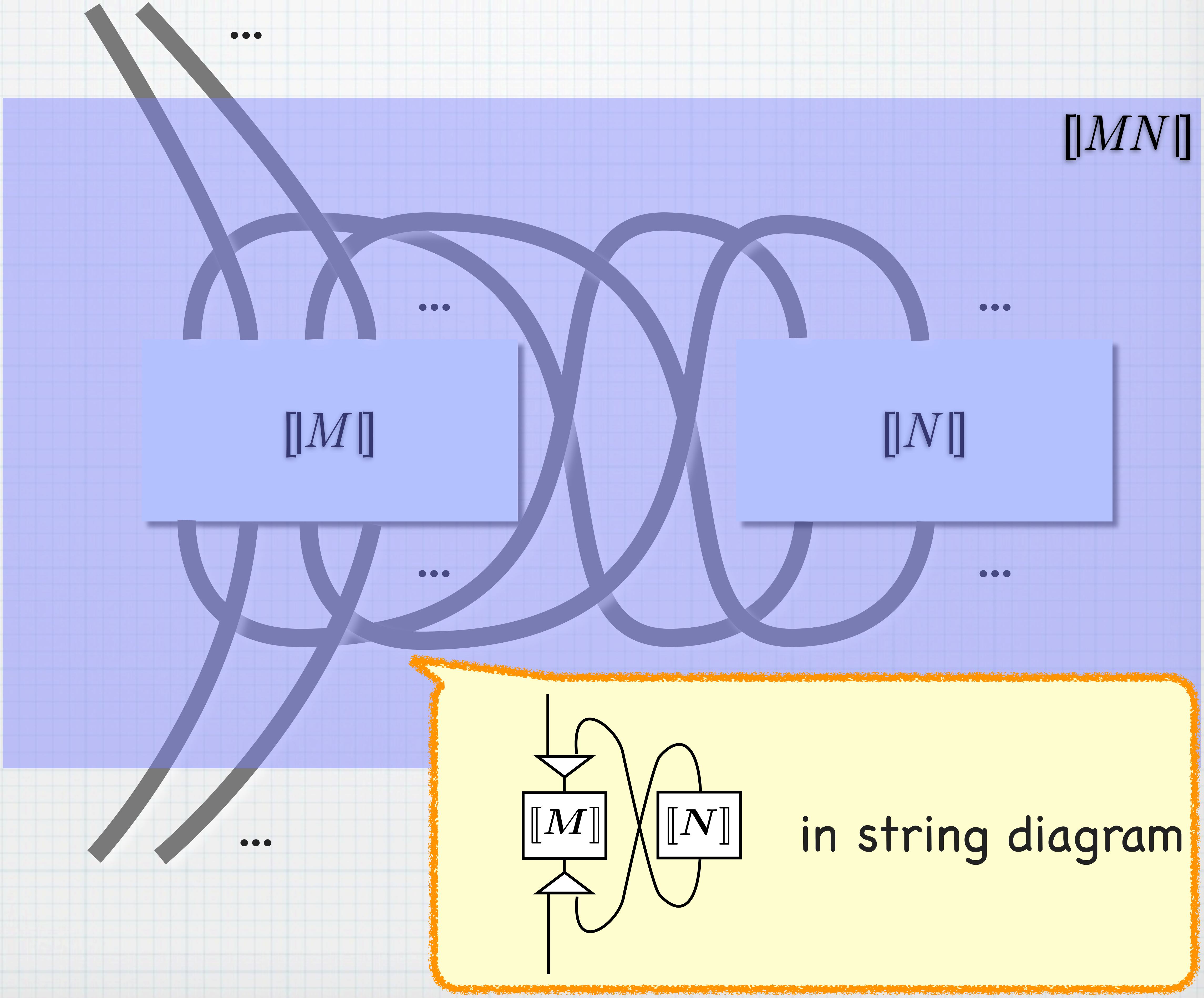
$$f \overset{\mathrm{tr}}{\longmapsto} \mathrm{tr}(f)$$

  - Why for GoI?

$$[\![\boldsymbol{MN}]\!] =$$

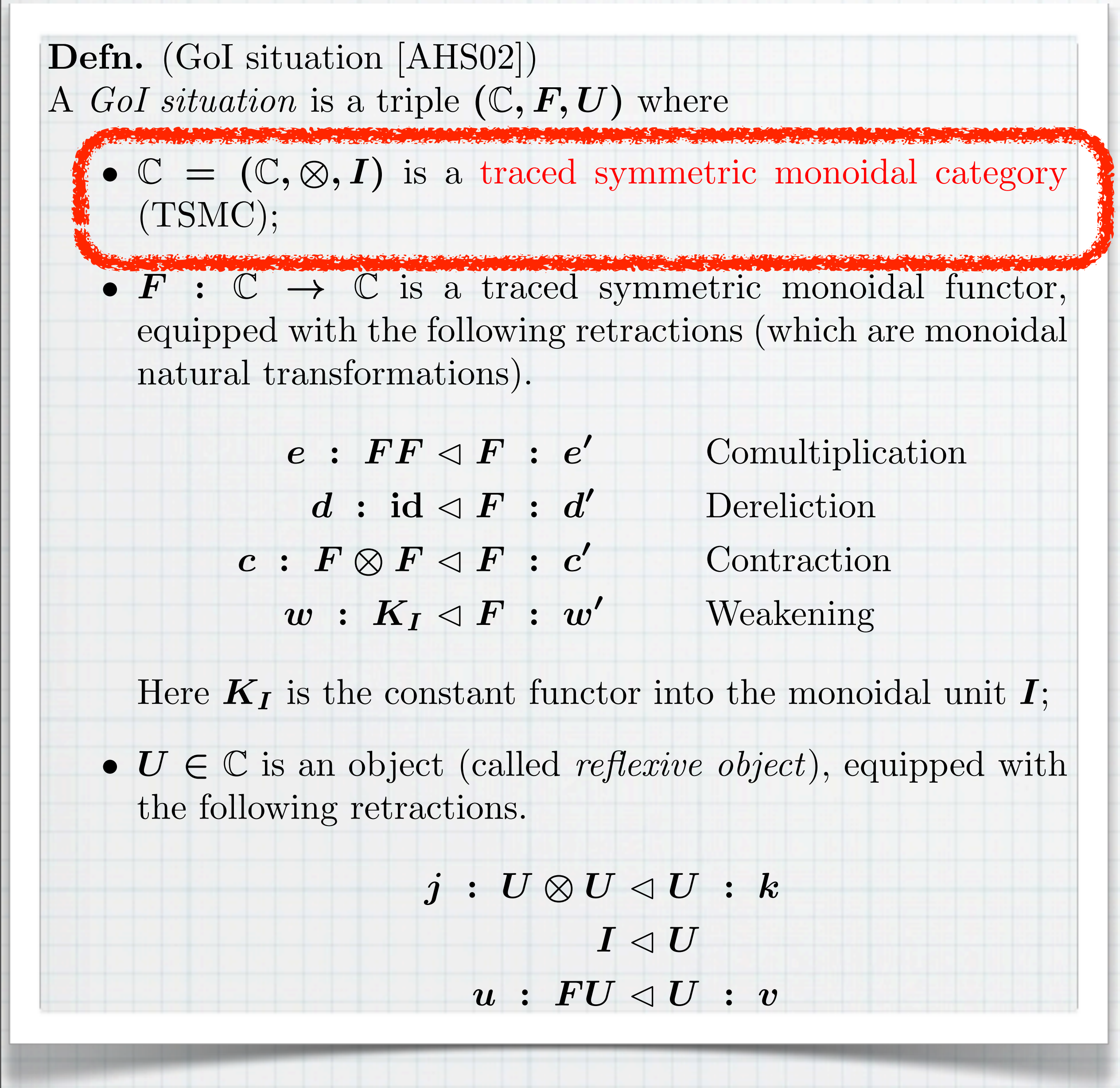


$$[\![MN]\!] =$$



in string diagram

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

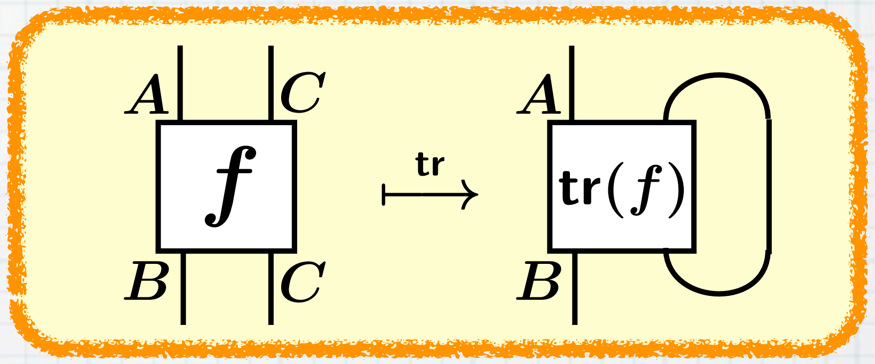
$$\begin{aligned} \boldsymbol{e} &: \boldsymbol{FF} \lhd \boldsymbol{F} : \boldsymbol{e'} && \text{Comultiplication} \\ \boldsymbol{d} &: \mathbf{id} \lhd \boldsymbol{F} : \boldsymbol{d'} && \text{Dereliction} \\ \boldsymbol{c} &: \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{c'} && \text{Contraction} \\ \boldsymbol{w} &: \boldsymbol{K_I} \lhd \boldsymbol{F} : \boldsymbol{w'} && \text{Weakening} \end{aligned}$$

Here $\boldsymbol{K_I}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;

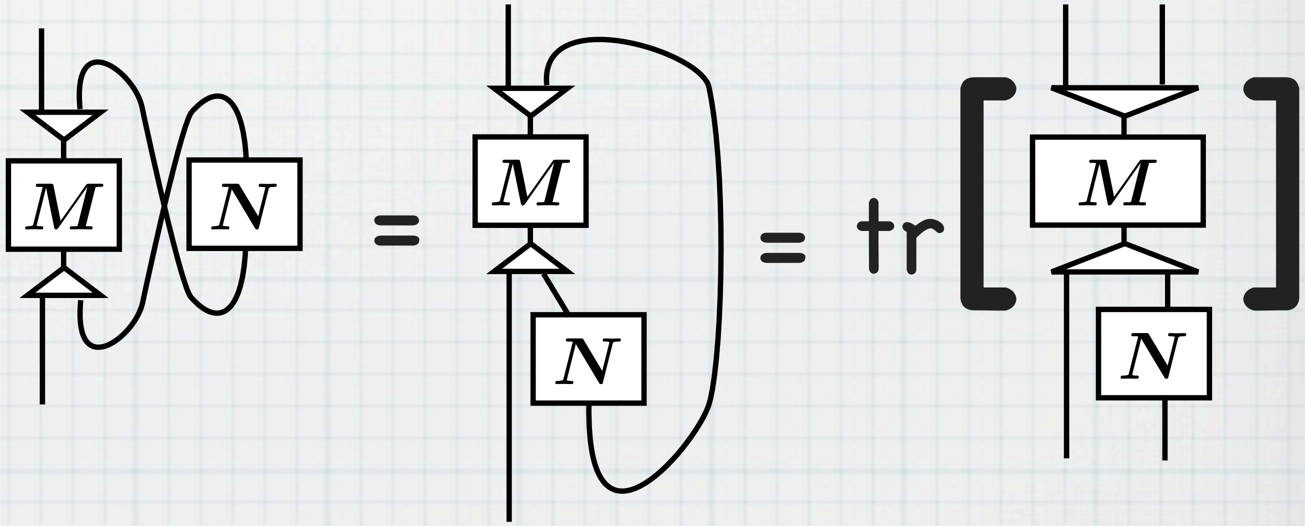
- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{aligned} \boldsymbol{j} : \boldsymbol{U} \otimes \boldsymbol{U} &\lhd \boldsymbol{U} : \boldsymbol{k} \\ \boldsymbol{I} &\lhd \boldsymbol{U} \\ \boldsymbol{u} : \boldsymbol{FU} &\lhd \boldsymbol{U} : \boldsymbol{v} \end{aligned}$$

- Traced sym. monoidal cat.
  - Where one can "feedback"



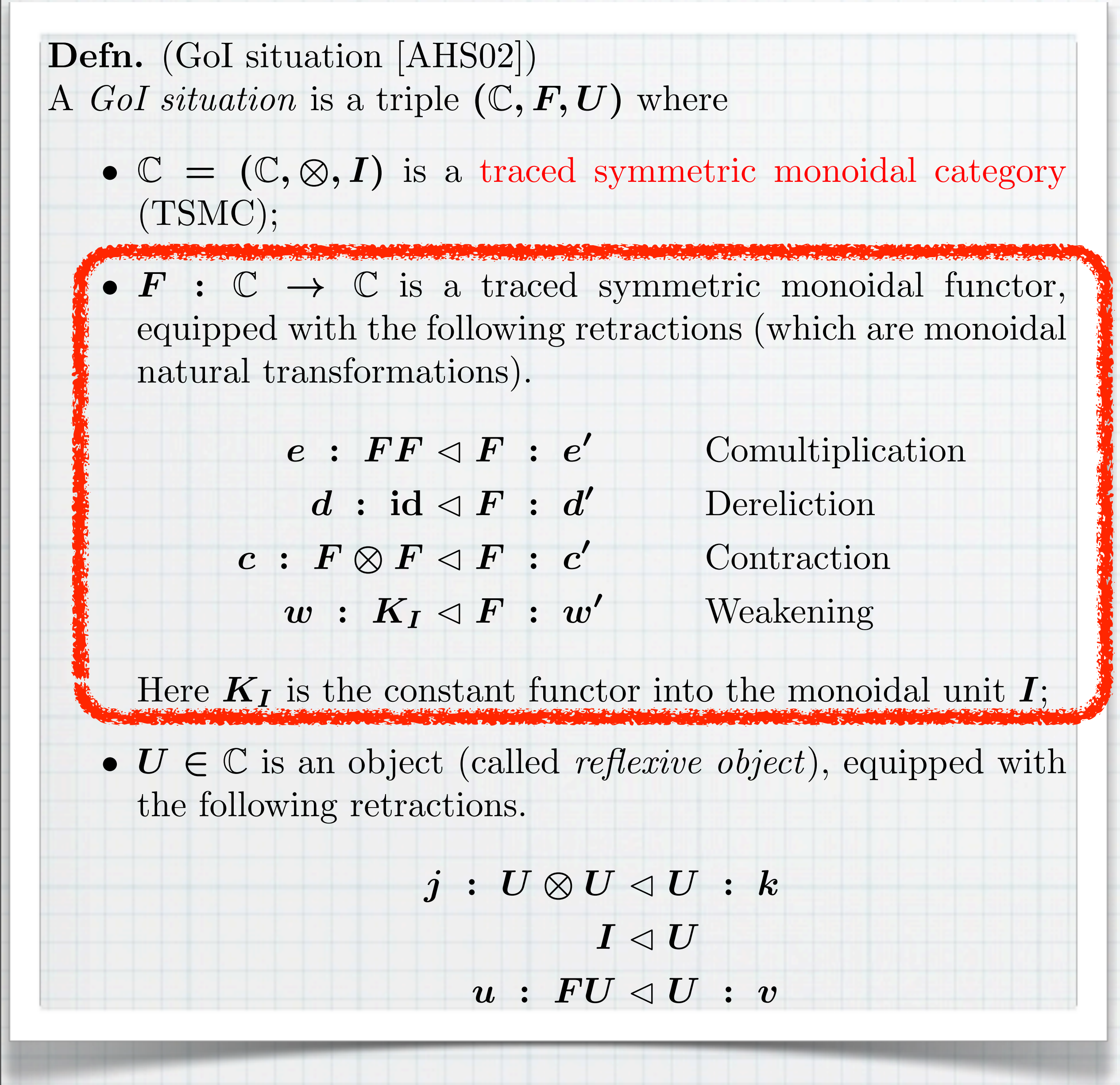
  - Why for GoI?



- Leading example: **Pfn**

Hasuo (Tokyo)

GoI situation



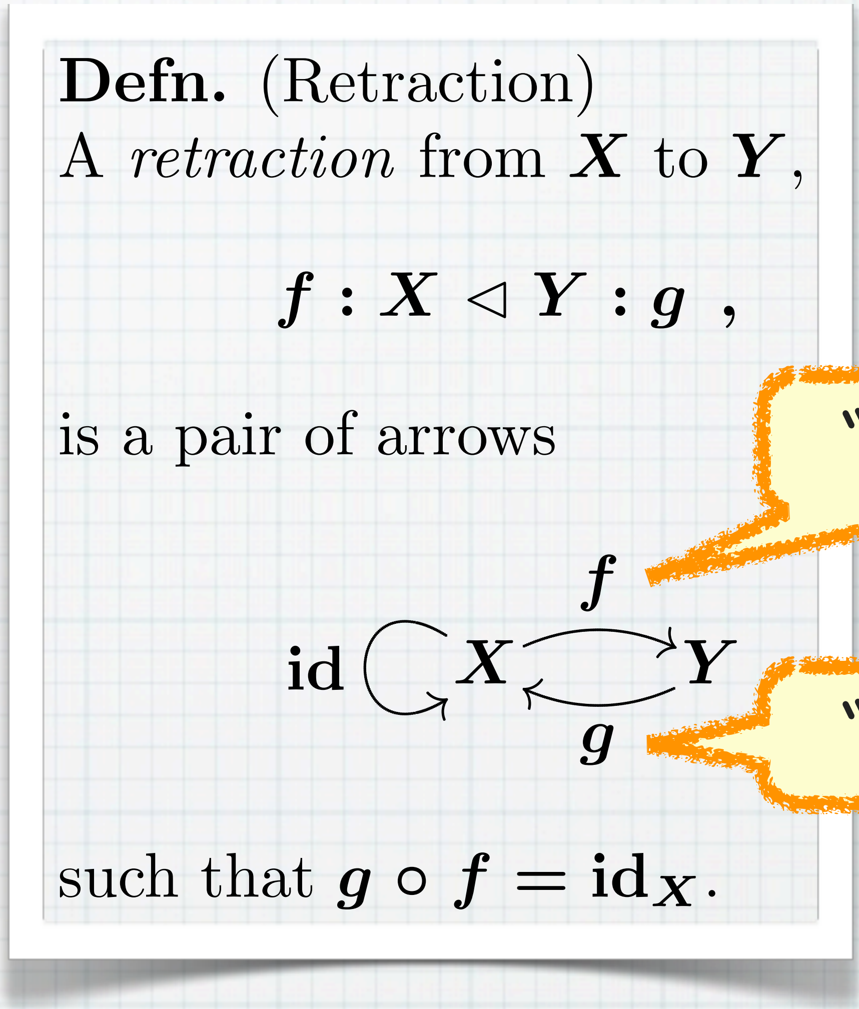
Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$$\begin{array}{rcll} \boldsymbol{e} : \boldsymbol{FF} \lhd \boldsymbol{F} : \boldsymbol{e'} & & \text{Comultiplication} \\ \boldsymbol{d} : \mathbf{id} \lhd \boldsymbol{F} : \boldsymbol{d'} & & \text{Dereliction} \\ \boldsymbol{c} : \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{c'} & & \text{Contraction} \\ \boldsymbol{w} : \boldsymbol{K_I} \lhd \boldsymbol{F} : \boldsymbol{w'} & & \text{Weakening} \end{array}$$

  Here $\boldsymbol{K_I}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;
- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{array}{c} \boldsymbol{j} : \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} : \boldsymbol{k} \\ \boldsymbol{I} \lhd \boldsymbol{U} \\ \boldsymbol{u} : \boldsymbol{FU} \lhd \boldsymbol{U} : \boldsymbol{v} \end{array}$$



Defn. (Retraction)
A *retraction* from $\boldsymbol{X}$ to $\boldsymbol{Y}$,

$$\boldsymbol{f} : \boldsymbol{X} \lhd \boldsymbol{Y} : \boldsymbol{g} \ ,$$

is a pair of arrows

$$\begin{tikzcd} \boldsymbol{X} \arrow[loop left, "\mathbf{id}"] \arrow[r, bend left, "\boldsymbol{f}"] & \boldsymbol{Y} \arrow[l, bend left, "\boldsymbol{g}"] \end{tikzcd}$$

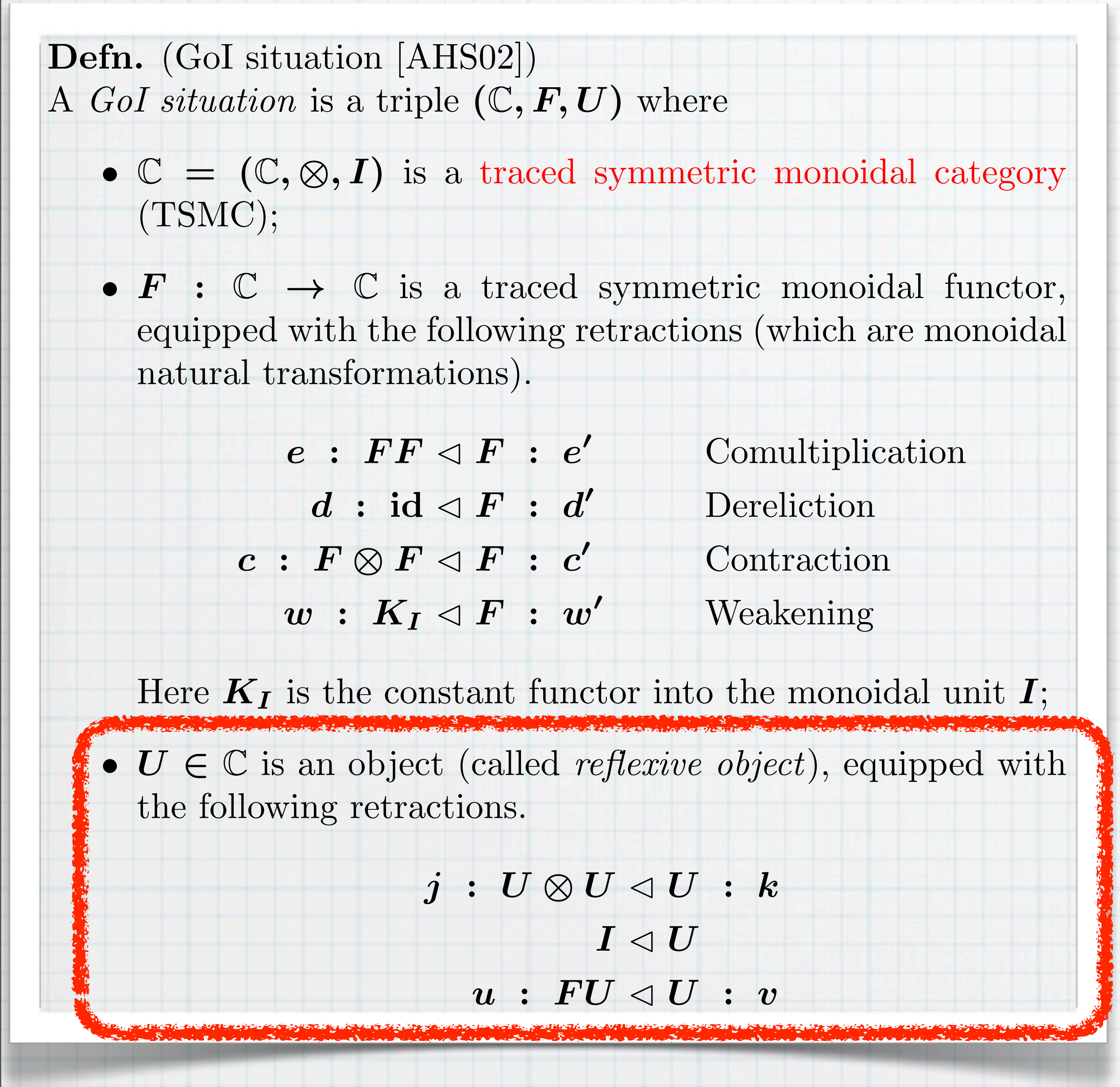
such that $\boldsymbol{g} \circ \boldsymbol{f} = \mathbf{id}_{\boldsymbol{X}}$.

"embedding"

"projection"

- ✱ Functor F
  - ✱ For obtaining $! : A \to A$

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$\boldsymbol{e} : \boldsymbol{F}\boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{e}'$	Comultiplication
$\boldsymbol{d} : \mathbf{id} \lhd \boldsymbol{F} : \boldsymbol{d}'$	Dereliction
$\boldsymbol{c} : \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{c}'$	Contraction
$\boldsymbol{w} : \boldsymbol{K}_{\boldsymbol{I}} \lhd \boldsymbol{F} : \boldsymbol{w}'$	Weakening

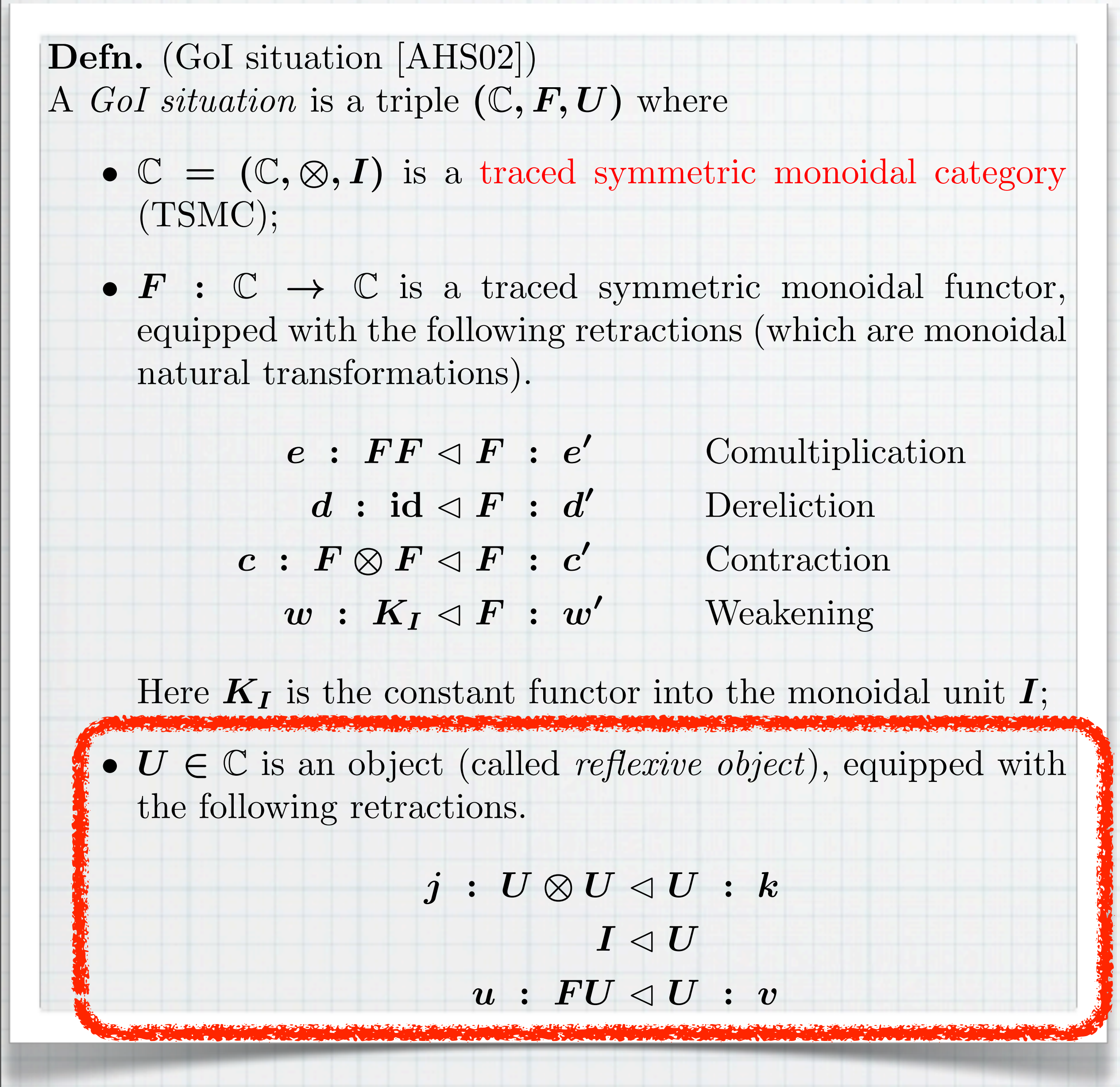
Here $\boldsymbol{K}_{\boldsymbol{I}}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;

- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{aligned} \boldsymbol{j} : \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} &: \boldsymbol{k} \\ \boldsymbol{I} \lhd \boldsymbol{U} & \\ \boldsymbol{u} : \boldsymbol{F}\boldsymbol{U} \lhd \boldsymbol{U} &: \boldsymbol{v} \end{aligned}$$

- The reflexive object U
  - Retr. $$\begin{tikzcd} \boldsymbol{U} \otimes \boldsymbol{U} \arrow[r, bend left, "\boldsymbol{j}"] & \boldsymbol{U} \arrow[l, bend left, "\boldsymbol{k}"] \end{tikzcd}$$

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

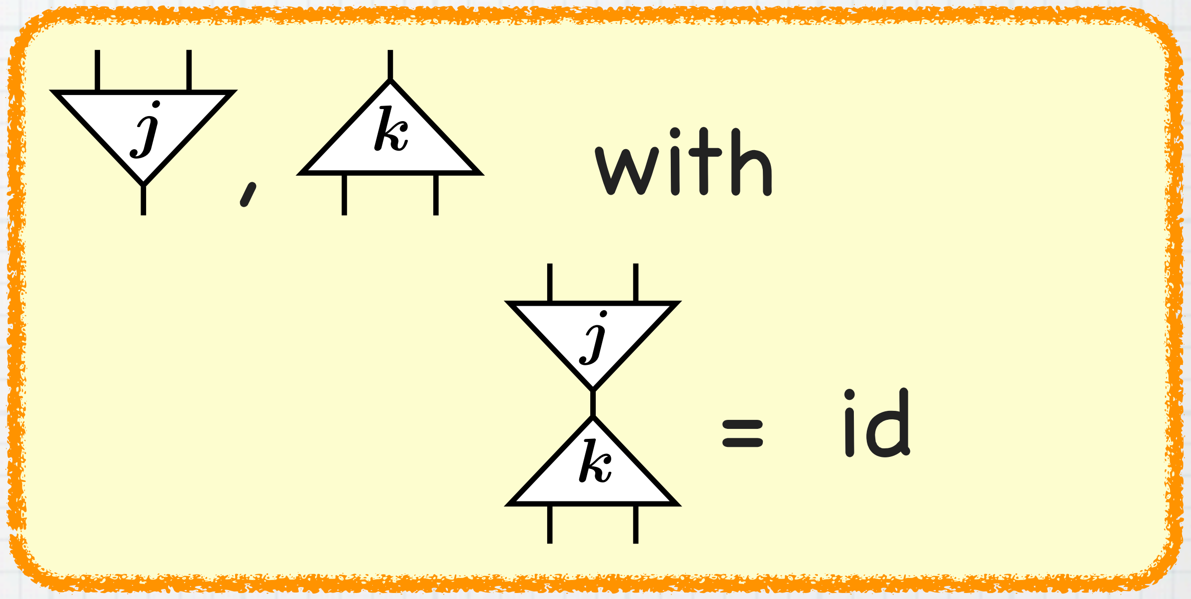
$$\begin{array}{rcll} e : FF \lhd F : e' & & \text{Comultiplication} \\ d : \mathrm{id} \lhd F : d' & & \text{Dereliction} \\ c : F \otimes F \lhd F : c' & & \text{Contraction} \\ w : K_I \lhd F : w' & & \text{Weakening} \end{array}$$

Here K_I is the constant functor into the monoidal unit I;

- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$j : U \otimes U \lhd U : k$$
$$I \lhd U$$
$$u : FU \lhd U : v$$

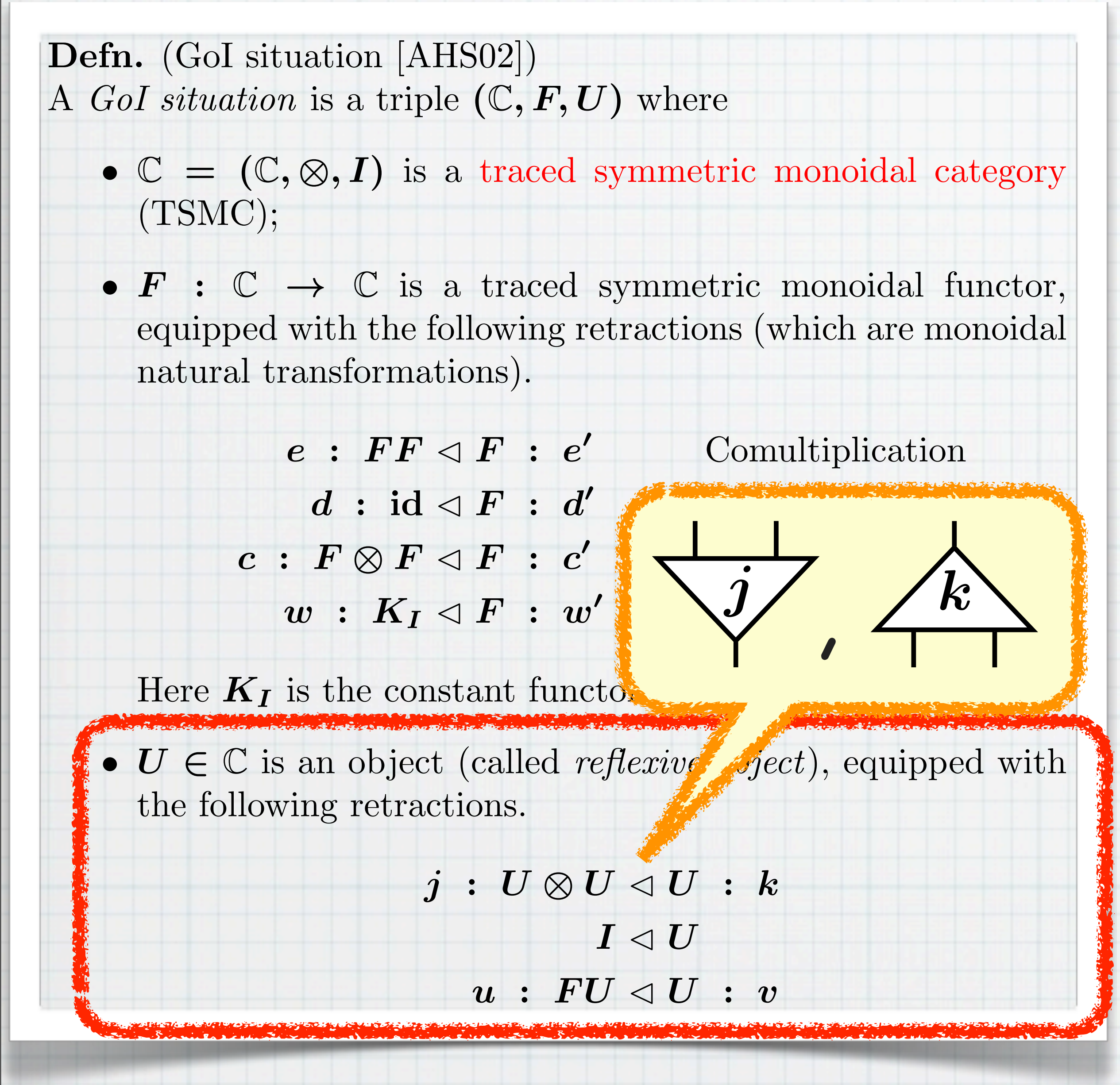
- The reflexive object U
- Retr. $$\begin{tikzcd} U \otimes U \arrow[r, bend left, "j"] & U \arrow[l, bend left, "k"] \end{tikzcd}$$



with

= id

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

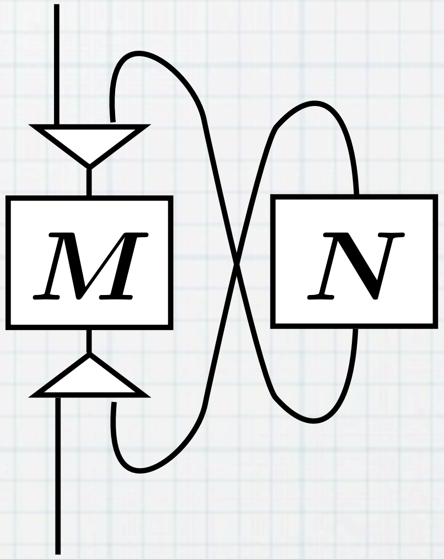
$$\begin{array}{rcl} \boldsymbol{e} : \boldsymbol{FF} \lhd \boldsymbol{F} : \boldsymbol{e'} & & \text{Comultiplication} \\ \boldsymbol{d} : \mathbf{id} \lhd \boldsymbol{F} : \boldsymbol{d'} & & \\ \boldsymbol{c} : \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{c'} & & \\ \boldsymbol{w} : \boldsymbol{K_I} \lhd \boldsymbol{F} : \boldsymbol{w'} & & \end{array}$$

Here $\boldsymbol{K_I}$ is the constant functo

- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

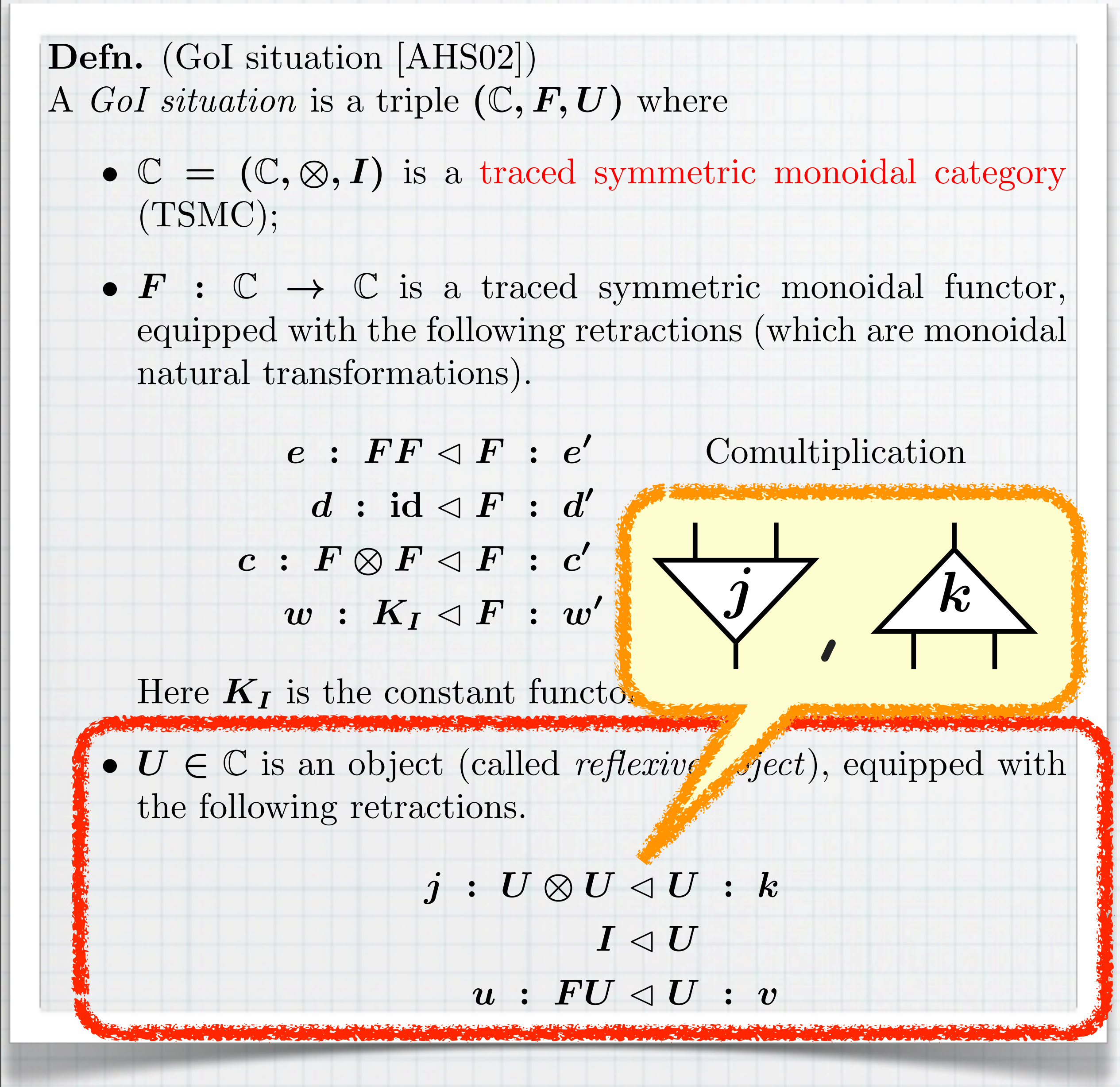
$$\begin{array}{c} \boldsymbol{j} : \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} : \boldsymbol{k} \\ \boldsymbol{I} \lhd \boldsymbol{U} \\ \boldsymbol{u} : \boldsymbol{FU} \lhd \boldsymbol{U} : \boldsymbol{v} \end{array}$$

- The reflexive object U
  - Why for GoI?



  - Example in **Pfn**:

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

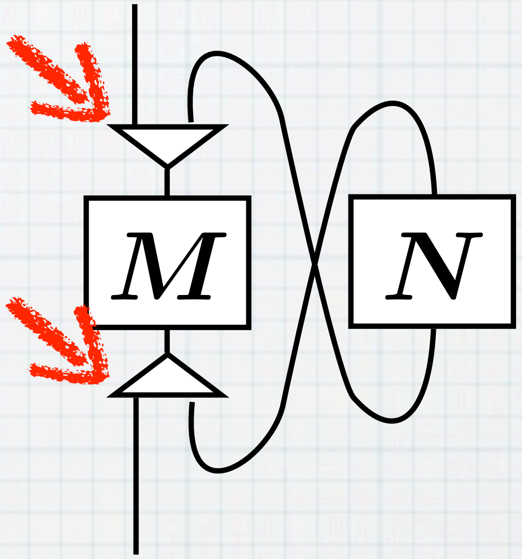
$$\begin{array}{rcl} \boldsymbol{e} : \boldsymbol{FF} \lhd \boldsymbol{F} : \boldsymbol{e'} & & \text{Comultiplication} \\ \boldsymbol{d} : \mathbf{id} \lhd \boldsymbol{F} : \boldsymbol{d'} & & \\ \boldsymbol{c} : \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{c'} & & \\ \boldsymbol{w} : \boldsymbol{K_I} \lhd \boldsymbol{F} : \boldsymbol{w'} & & \end{array}$$

Here $\boldsymbol{K_I}$ is the constant functo

- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexiv* *ject*), equipped with the following retractions.

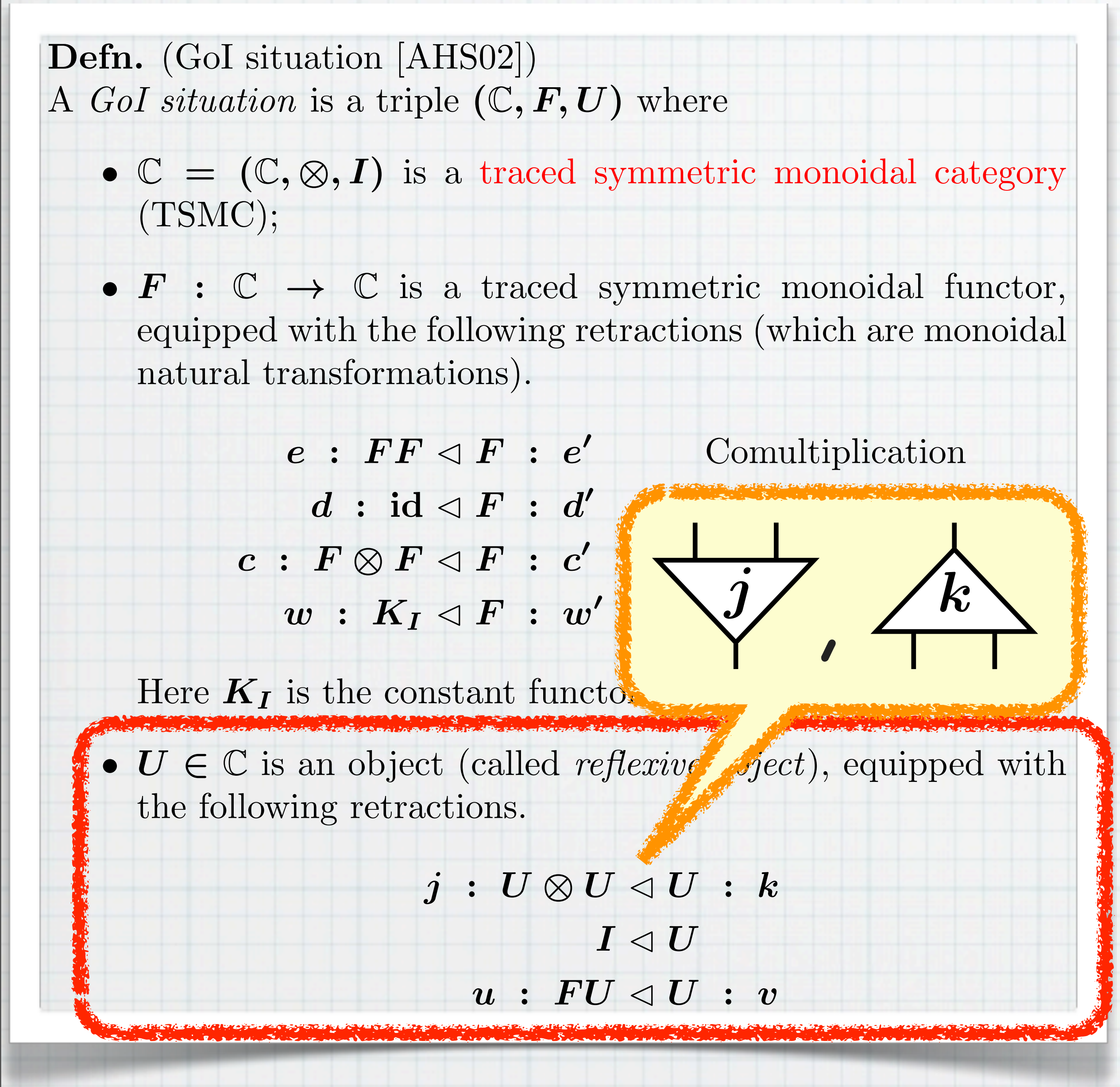
$$\begin{array}{c} \boldsymbol{j} : \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} : \boldsymbol{k} \\ \boldsymbol{I} \lhd \boldsymbol{U} \\ \boldsymbol{u} : \boldsymbol{FU} \lhd \boldsymbol{U} : \boldsymbol{v} \end{array}$$

- The reflexive object U
  - Why for GoI?



  - Example in **Pfn**:

GoI situation



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

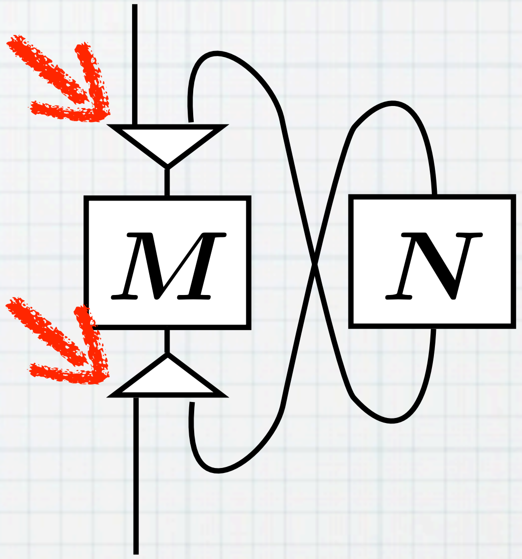
$$\begin{aligned} e &: FF \lhd F : e' && \text{Comultiplication} \\ d &: \mathrm{id} \lhd F : d' \\ c &: F \otimes F \lhd F : c' \\ w &: K_I \lhd F : w' \end{aligned}$$

  Here $\boldsymbol{K_I}$ is the constant functo

- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexiv* *ject*), equipped with the following retractions.

$$\begin{aligned} j : U \otimes U &\lhd U : k \\ I &\lhd U \\ u : FU &\lhd U : v \end{aligned}$$

- The reflexive object U
  - Why for GoI?

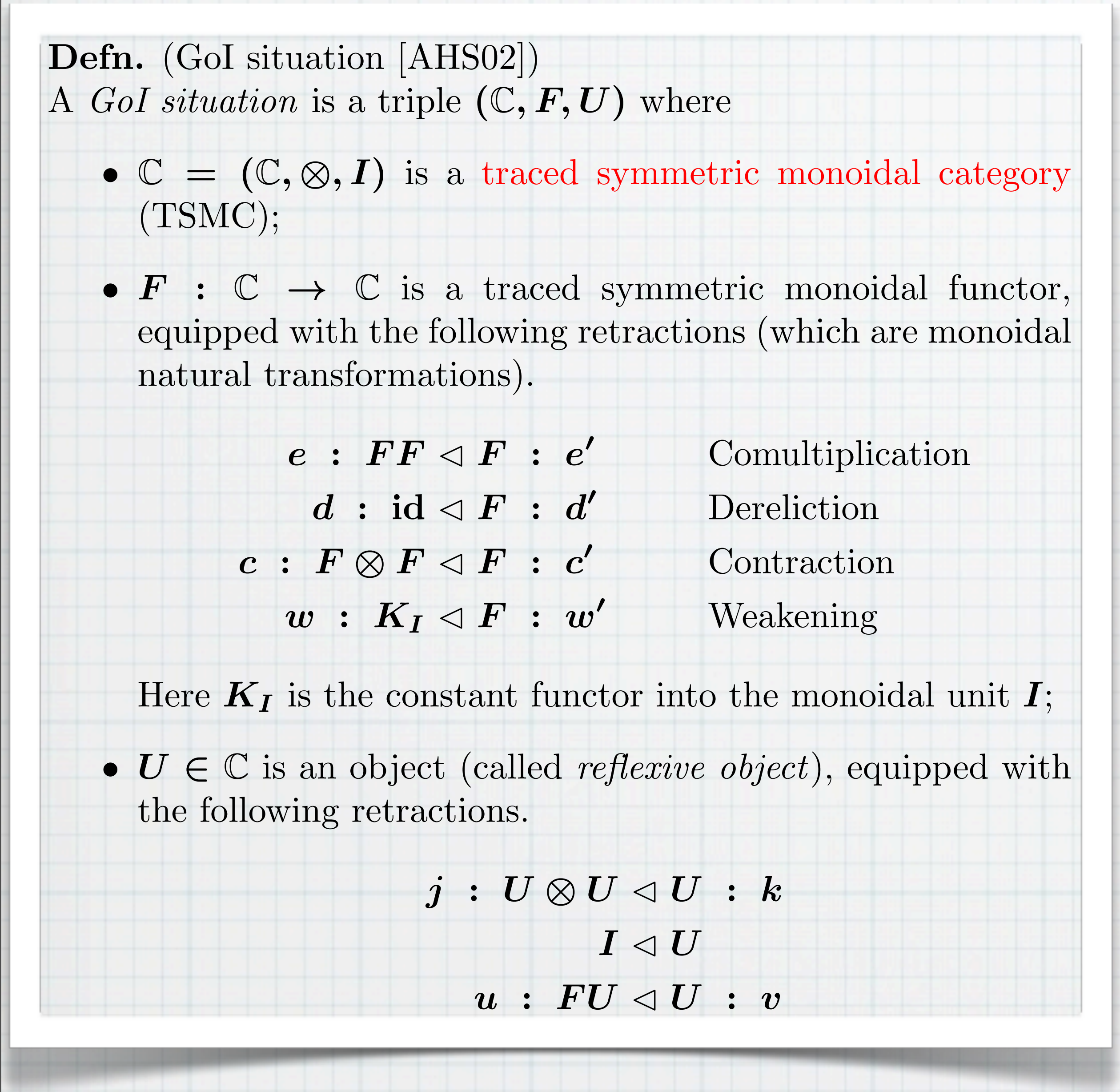


- Example in **Pfn**:

$\mathbb{N} \in \mathbf{Pfn}$, with

$$\mathbb{N} + \mathbb{N} \cong \mathbb{N},$$
$$\mathbb{N} \cdot \mathbb{N} \cong \mathbb{N}$$

GoI Situation: Summary



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

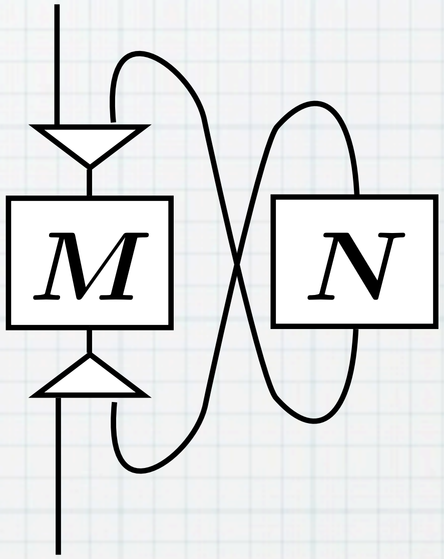
- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$\boldsymbol{e} : \boldsymbol{F}\boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{e}'$	Comultiplication
$\boldsymbol{d} : \mathbf{id} \lhd \boldsymbol{F} : \boldsymbol{d}'$	Dereliction
$\boldsymbol{c} : \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : \boldsymbol{c}'$	Contraction
$\boldsymbol{w} : \boldsymbol{K_I} \lhd \boldsymbol{F} : \boldsymbol{w}'$	Weakening

  Here $\boldsymbol{K_I}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;
- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

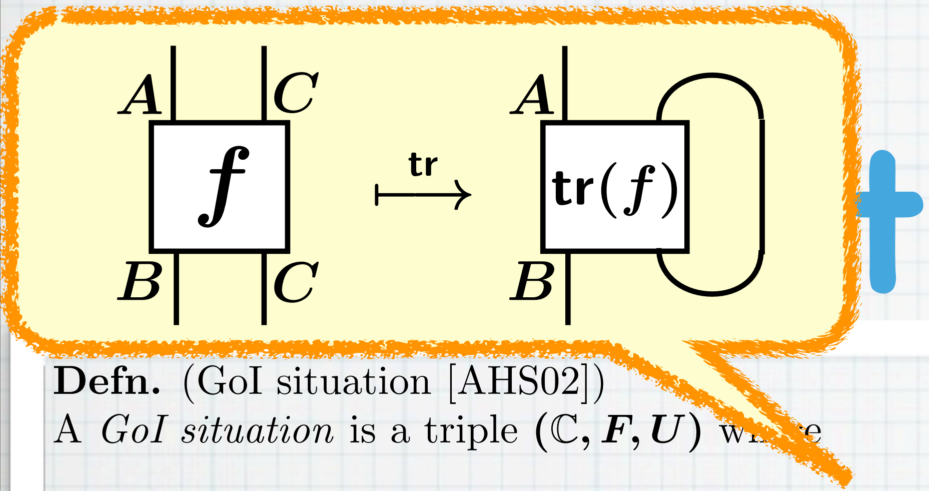
$$\boldsymbol{j} : \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} : \boldsymbol{k}$$
$$\boldsymbol{I} \lhd \boldsymbol{U}$$
$$\boldsymbol{u} : \boldsymbol{F}\boldsymbol{U} \lhd \boldsymbol{U} : \boldsymbol{v}$$

- Categorical axiomatics of the "GoI animation"



- Example:

$$(\mathbf{Pfn}, \mathbb{N} \cdot \_\,, \mathbb{N})$$



tuation: Summary

Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ where

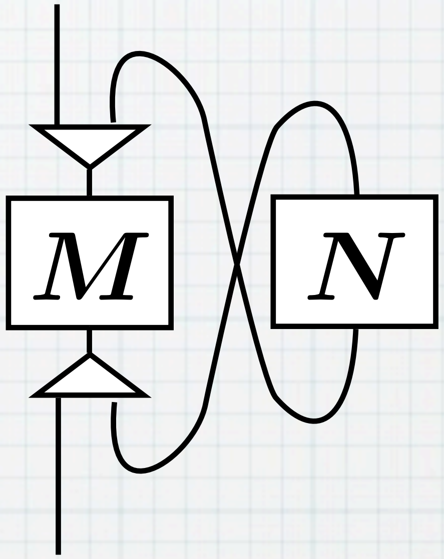
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$$\begin{array}{rcl}
e : FF \lhd F : e' & & \text{Comultiplication} \\
d : \mathrm{id} \lhd F : d' & & \text{Dereliction} \\
c : F \otimes F \lhd F : c' & & \text{Contraction} \\
w : K_I \lhd F : w' & & \text{Weakening}
\end{array}$$

  Here K_I is the constant functor into the monoidal unit I;
- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{array}{c}
j : U \otimes U \lhd U : k \\
I \lhd U \\
u : FU \lhd U : v
\end{array}$$

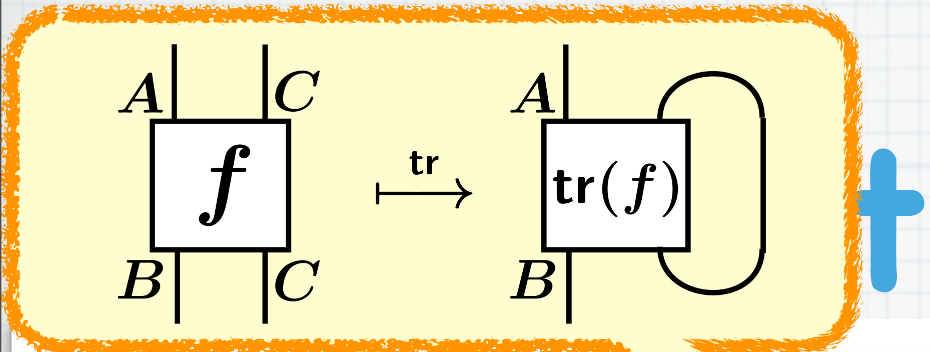
- Categorical axiomatics of the "GoI animation"

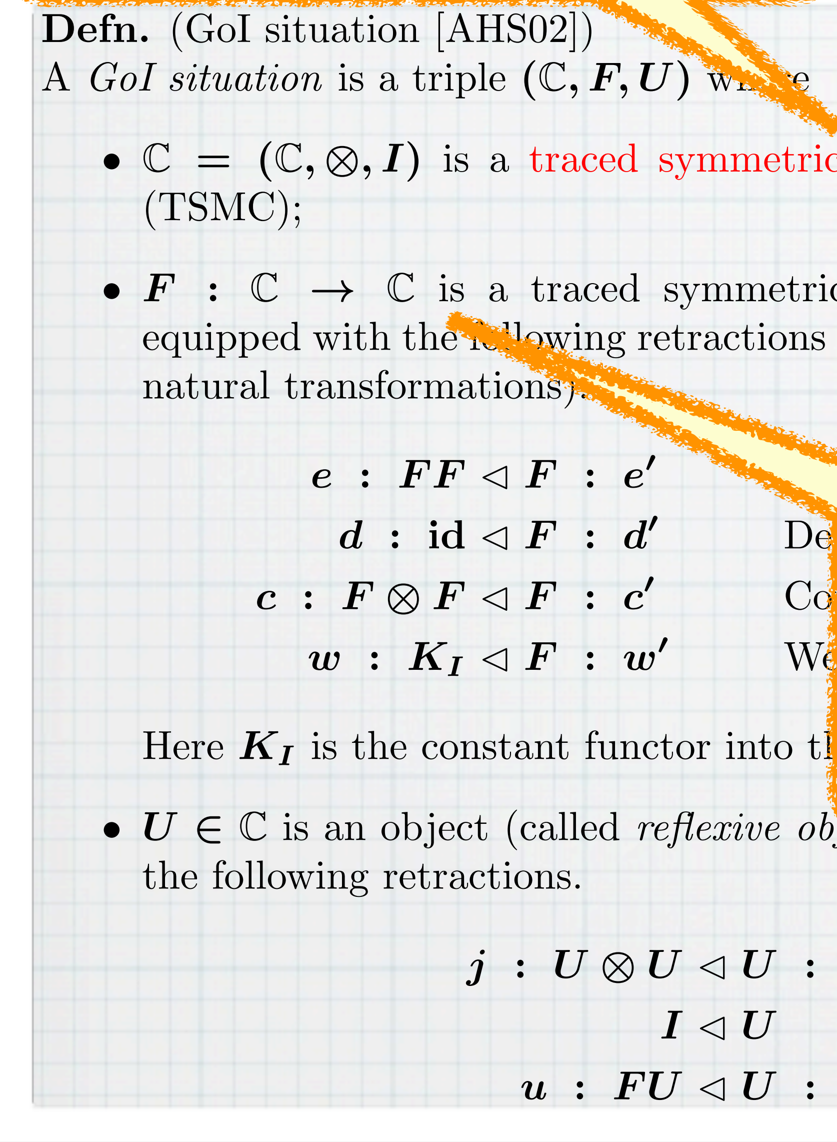


- Example:

$$(\mathbf{Pfn}, \mathbb{N} \cdot \_\,, \mathbb{N})$$

tuation: Summary




Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations):

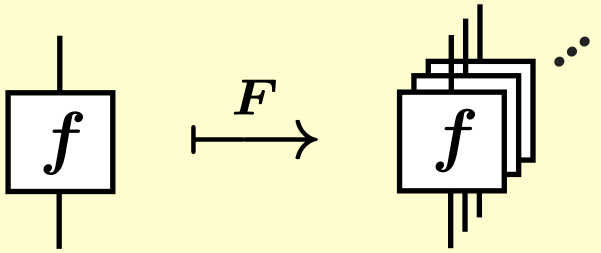
$$e : FF \lhd F : e'$$
$$d : \mathrm{id} \lhd F : d' \qquad \text{De}$$
$$c : F \otimes F \lhd F : c' \qquad \text{Co}$$
$$w : K_I \lhd F : w' \qquad \text{We}$$

Here K_I is the constant functor into th

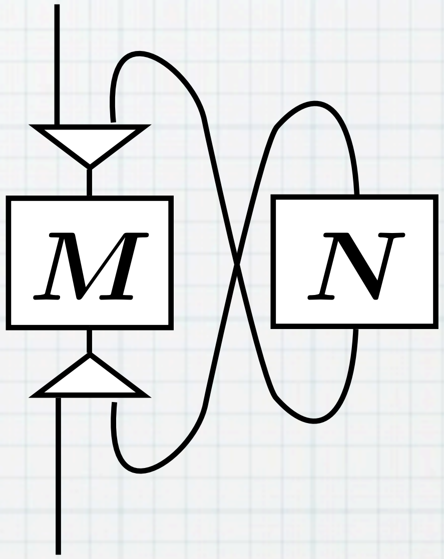
- $U \in \mathbb{C}$ is an object (called *reflexive obj* the following retractions.

$$j : U \otimes U \lhd U : k$$
$$I \lhd U$$
$$u : FU \lhd U : v$$

For ! , via



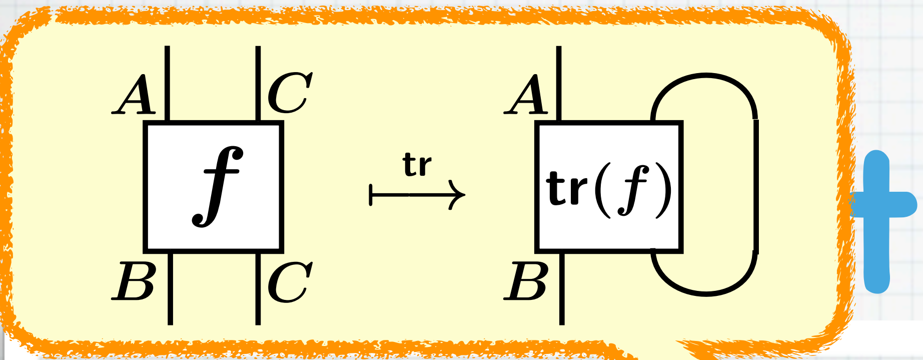
- Categorical axiomatics of the "GoI animation"



- Example:

$$(\mathbf{Pfn}, \mathbb{N} \cdot \_, \mathbb{N})$$

…tuation: Summary



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ wh e

- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations):

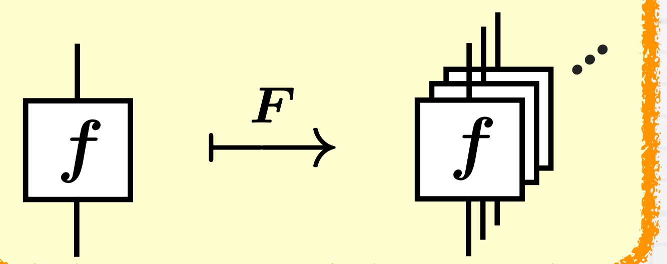
$$e : FF \lhd F : e'$$
$$d : \mathrm{id} \lhd F : d' \qquad \text{De}$$
$$c : F \otimes F \lhd F : c' \qquad \text{Co}$$
$$w : K_I \lhd F : w' \qquad \text{We}$$

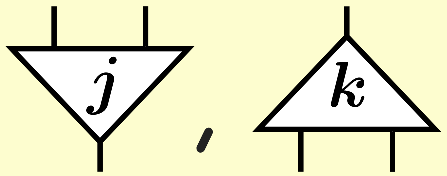
Here K_I is the constant functor into th

- $U \in \mathbb{C}$ is an object (called *reflexive obj* the following retractions.

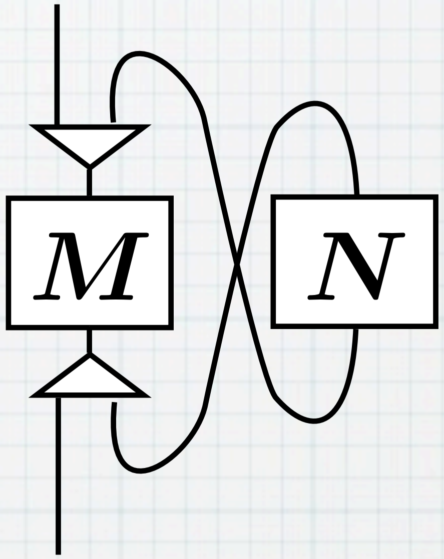
$$j : U \otimes U \lhd U : k$$
$$I \lhd U$$
$$u : FU \lhd U : v$$

For ! , via



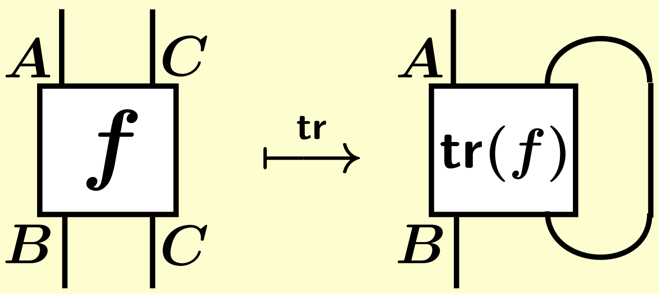


- Categorical axiomatics of the "GoI animation"

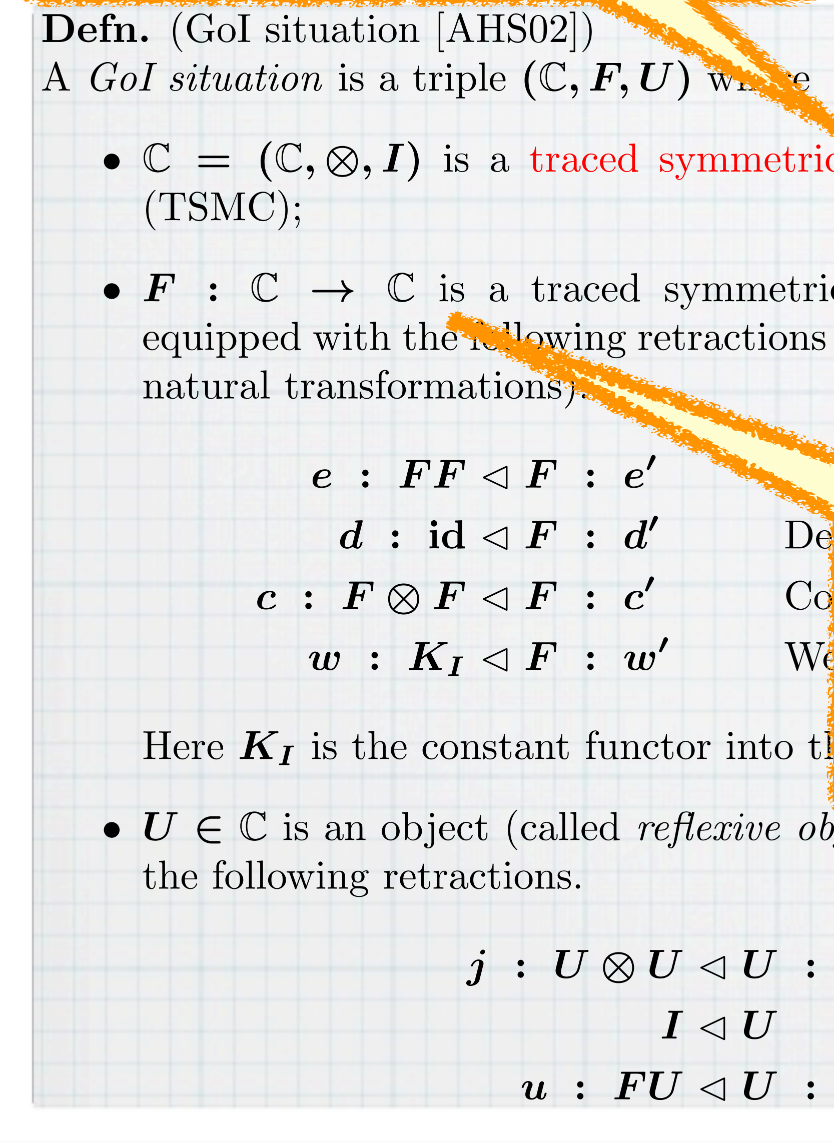


- Example:

$$(\mathbf{Pfn}, \mathbb{N} \cdot \_\, , \mathbb{N})$$



tuation: Summary



Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, F, U)$ w e

- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations):

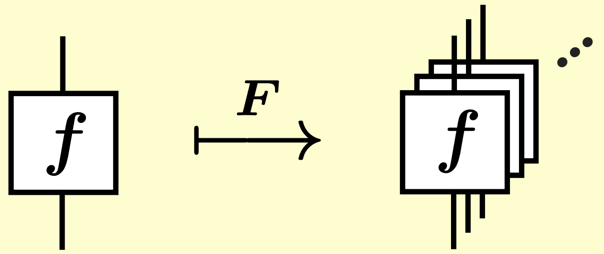
$$\begin{array}{rcl} e : FF \lhd F : e' & \\ d : \mathrm{id} \lhd F : d' & \text{De} \\ c : F \otimes F \lhd F : c' & \text{Co} \\ w : K_I \lhd F : w' & \text{We} \end{array}$$

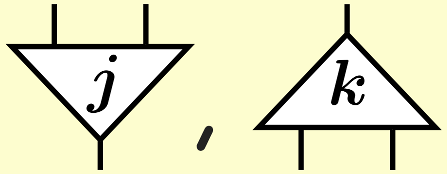
Here K_I is the constant functor into th

- $U \in \mathbb{C}$ is an object (called *reflexive obj* the following retractions.

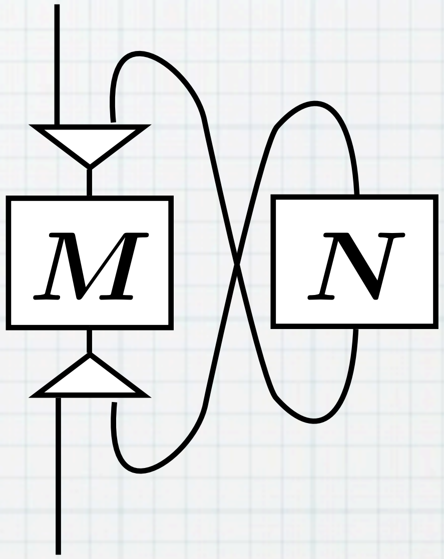
$$j : U \otimes U \lhd U : k$$
$$I \lhd U$$
$$u : FU \lhd U : v$$

For ! , via



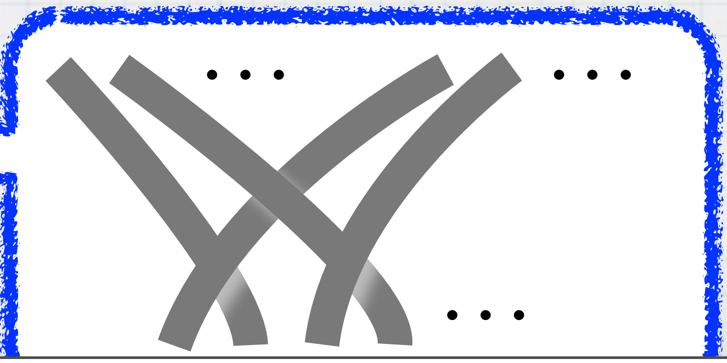


- Categorical axiomatics of the "GoI animation"



- Example:

$$(\mathbf{Pfn}, \mathbb{N} \cdot \_\,, \mathbb{N})$$



Categorical GoI: Constr. of an LCA

Thm. ([AHS02])
Given a GoI situation $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$, the homset

$$\mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$$

carries a canonical LCA structure.

Categorical GoI: Constr. of an LCA

Thm. ([AHS02])
Given a GoI situation $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$, the homset

$$\mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$$

carries a canonical LCA structure.

- Applicative str. $\cdot$
- ! operator
- Combinators B, C, I, ...

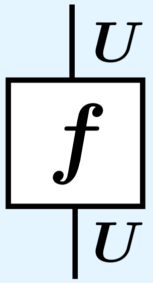
Categorical GoI: Constr. of an LCA

Thm. ([AHS02])
Given a GoI situation $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$, the homset

$$\mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$$

carries a canonical LCA structure.

- Applicative str. $\cdot$
- ! operator
- Combinators B, C, I, ...



$\in \mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$

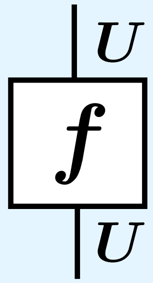
Categorical GoI: Constr. of an LCA

Thm. ([AHS02])
Given a GoI situation $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$, the homset

$$\mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$$

carries a canonical LCA structure.

- Applicative str. $\cdot$
- ! operator
- Combinators B, C, I, ...

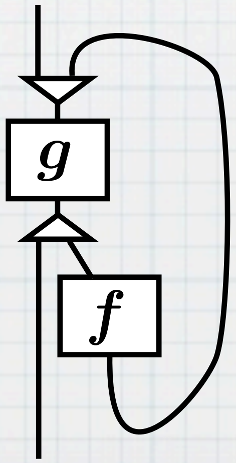


$\in \mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$

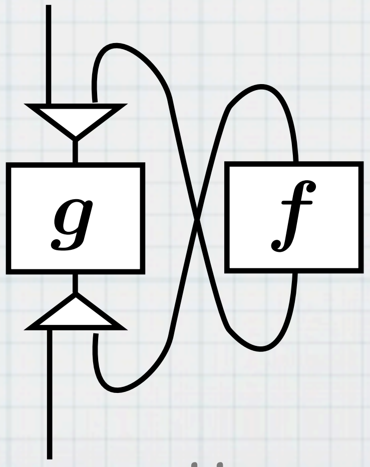
- $\boldsymbol{g} \cdot \boldsymbol{f}$

$$:= \mathbf{tr}\Big((\boldsymbol{U} \otimes \boldsymbol{f}) \circ \boldsymbol{k} \circ \boldsymbol{g} \circ \boldsymbol{j} \Big)$$

=



=

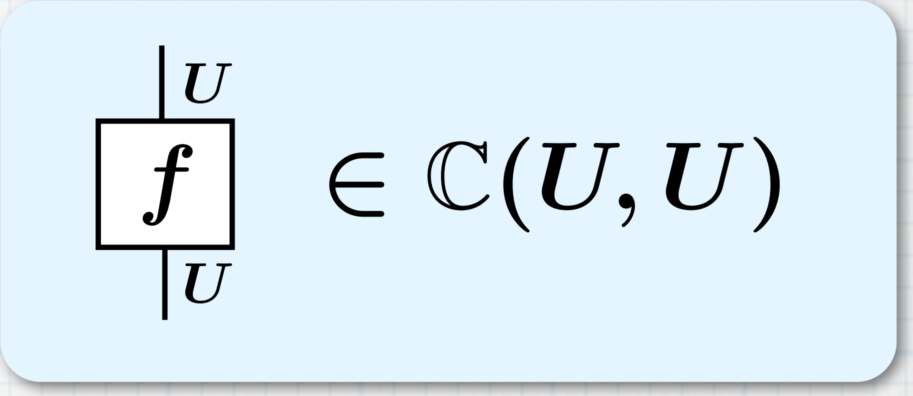


Categorical GoI: Constr. of an LCA

Thm. ([AHS02])
Given a GoI situation $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$, the homset

$$\mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$$

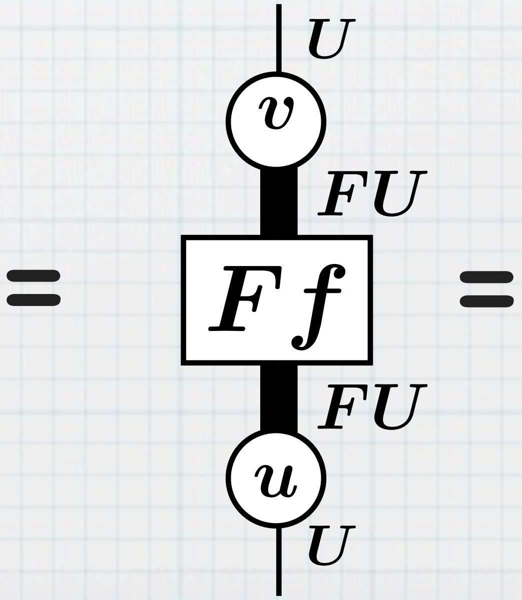
carries a canonical LCA structure.

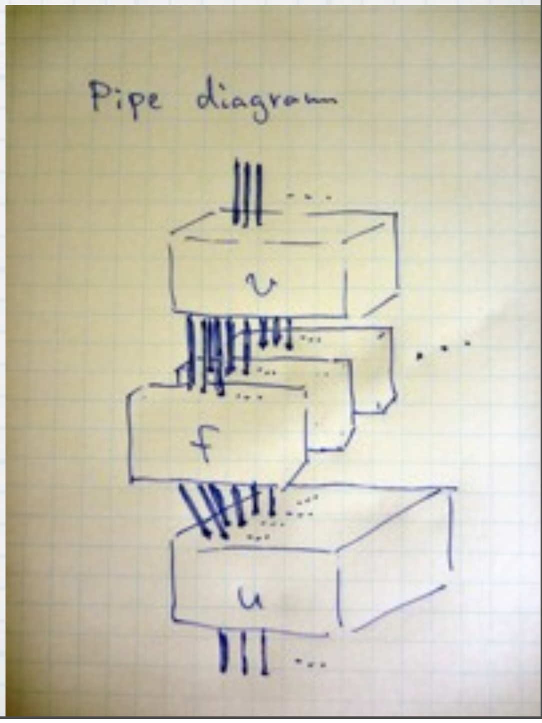


$\in \mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$

- Applicative str. ·
- ! operator
- Combinators B, C, I, ...

- $!\boldsymbol{f} \;:=\; \boldsymbol{u} \circ \boldsymbol{F}\boldsymbol{f} \circ \boldsymbol{v}$





Categorical GoI: Constr. of an LCA

- Combinator $Bxyz = x(yz)$

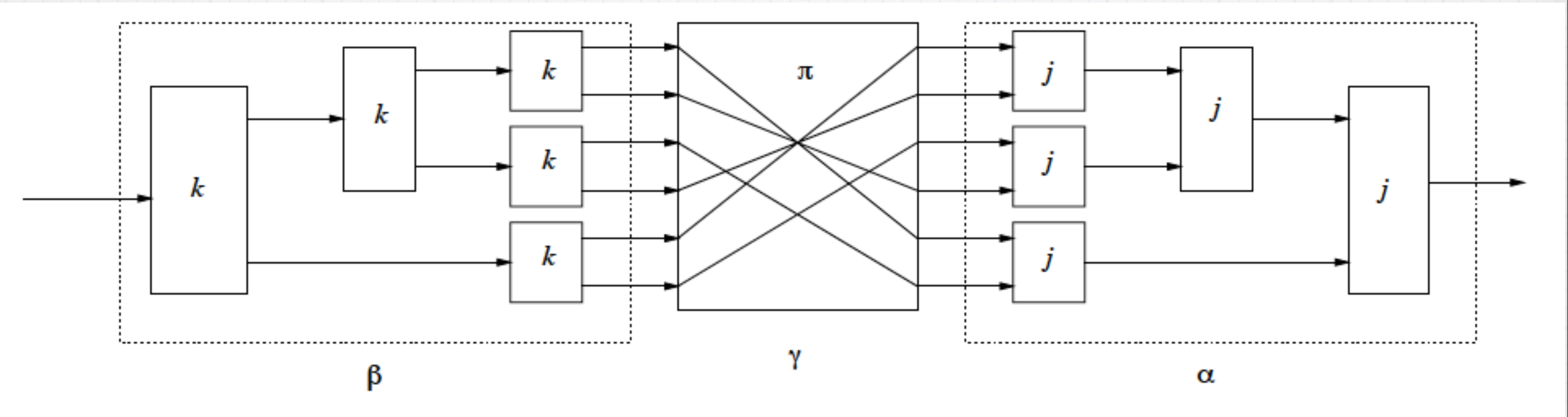
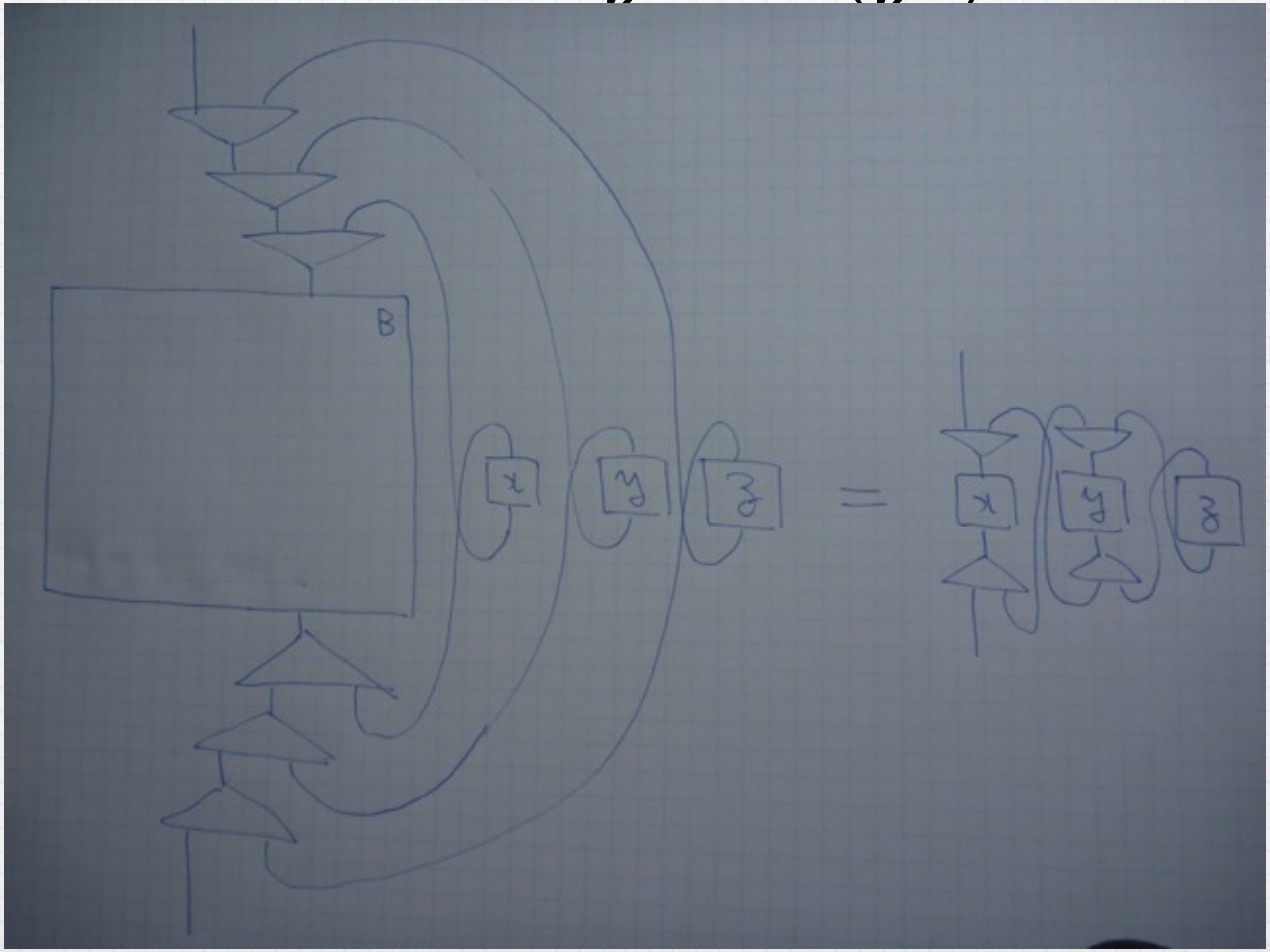


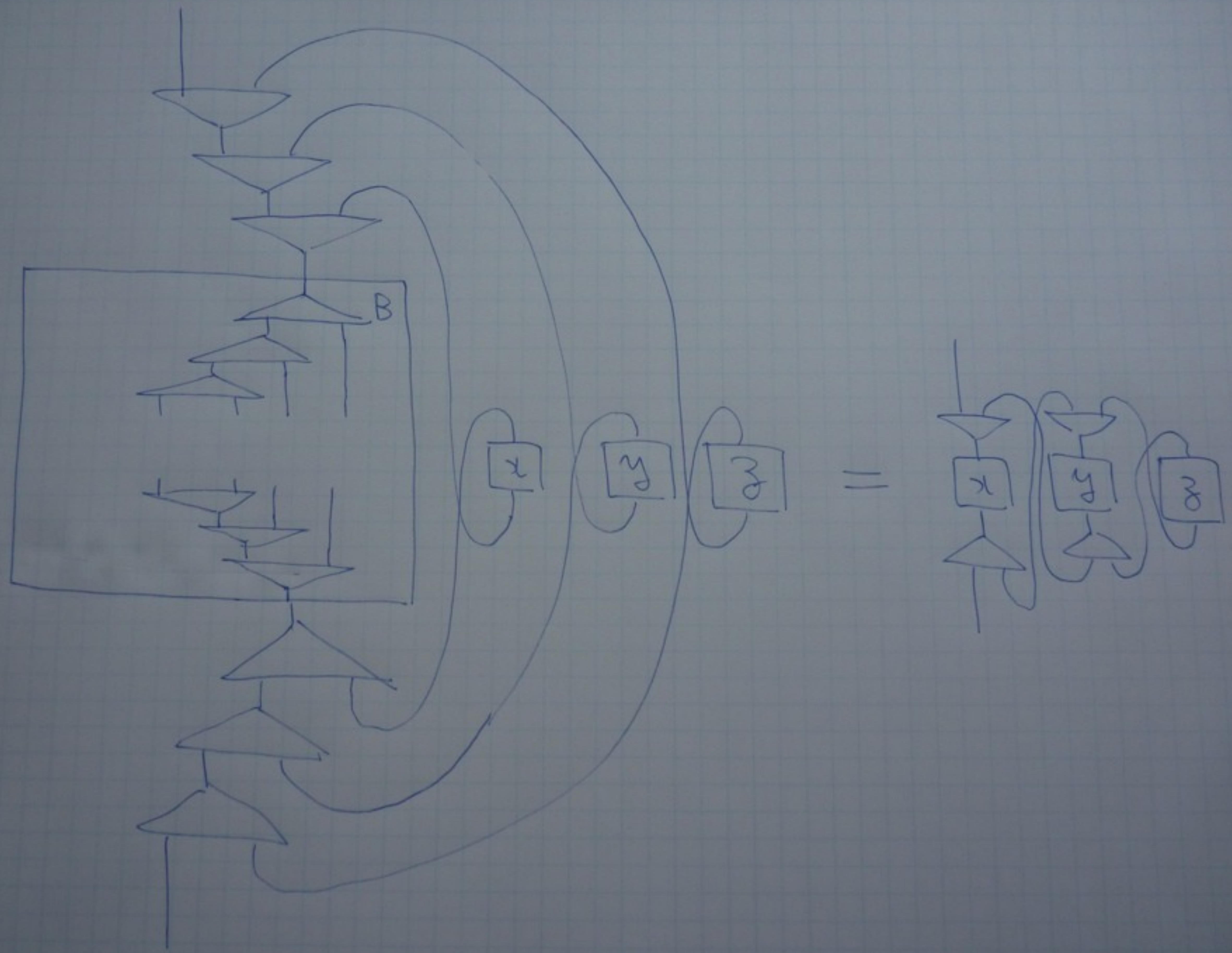
Figure 7: Composition Combinator B

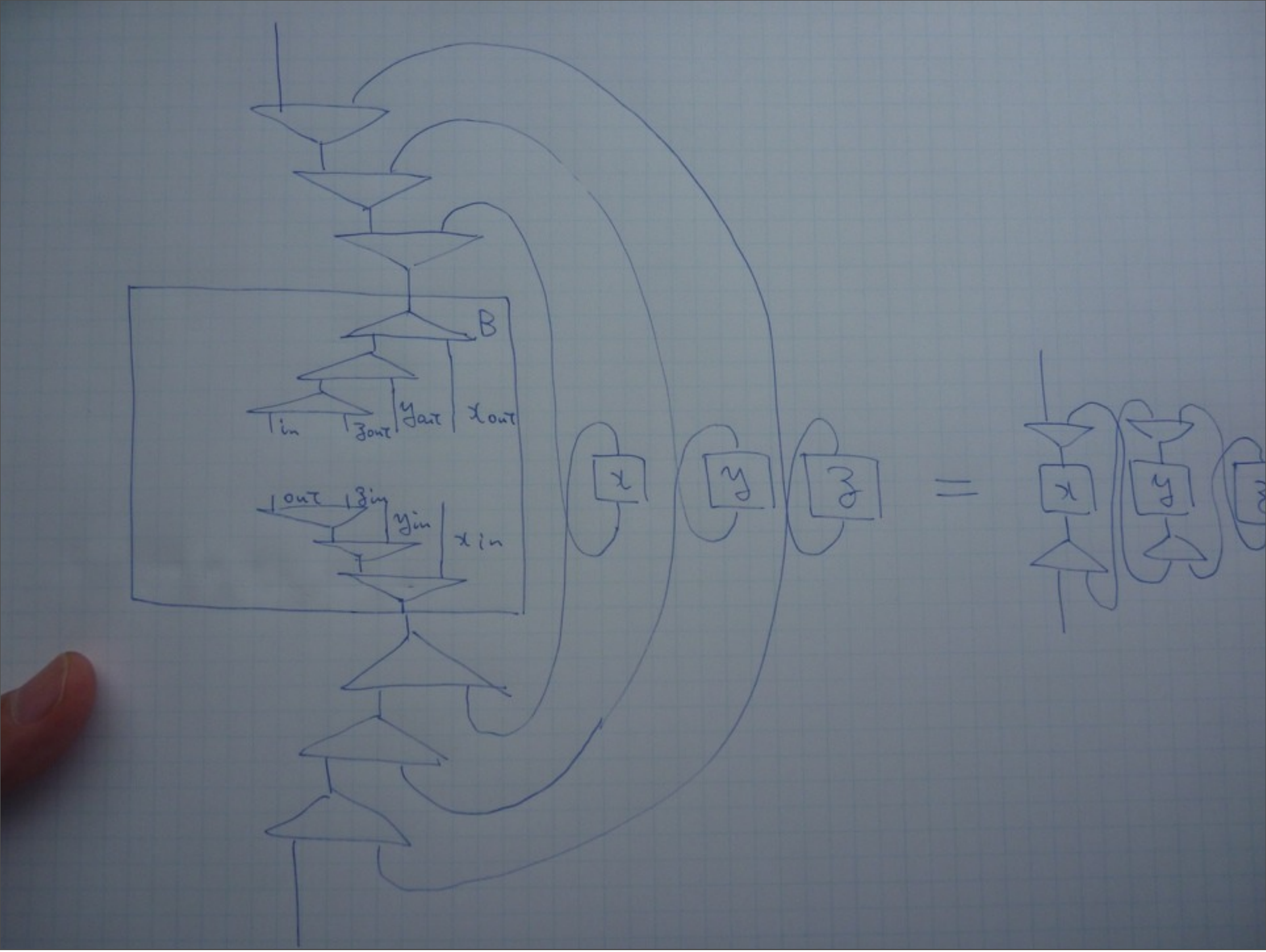
from [AHS02]

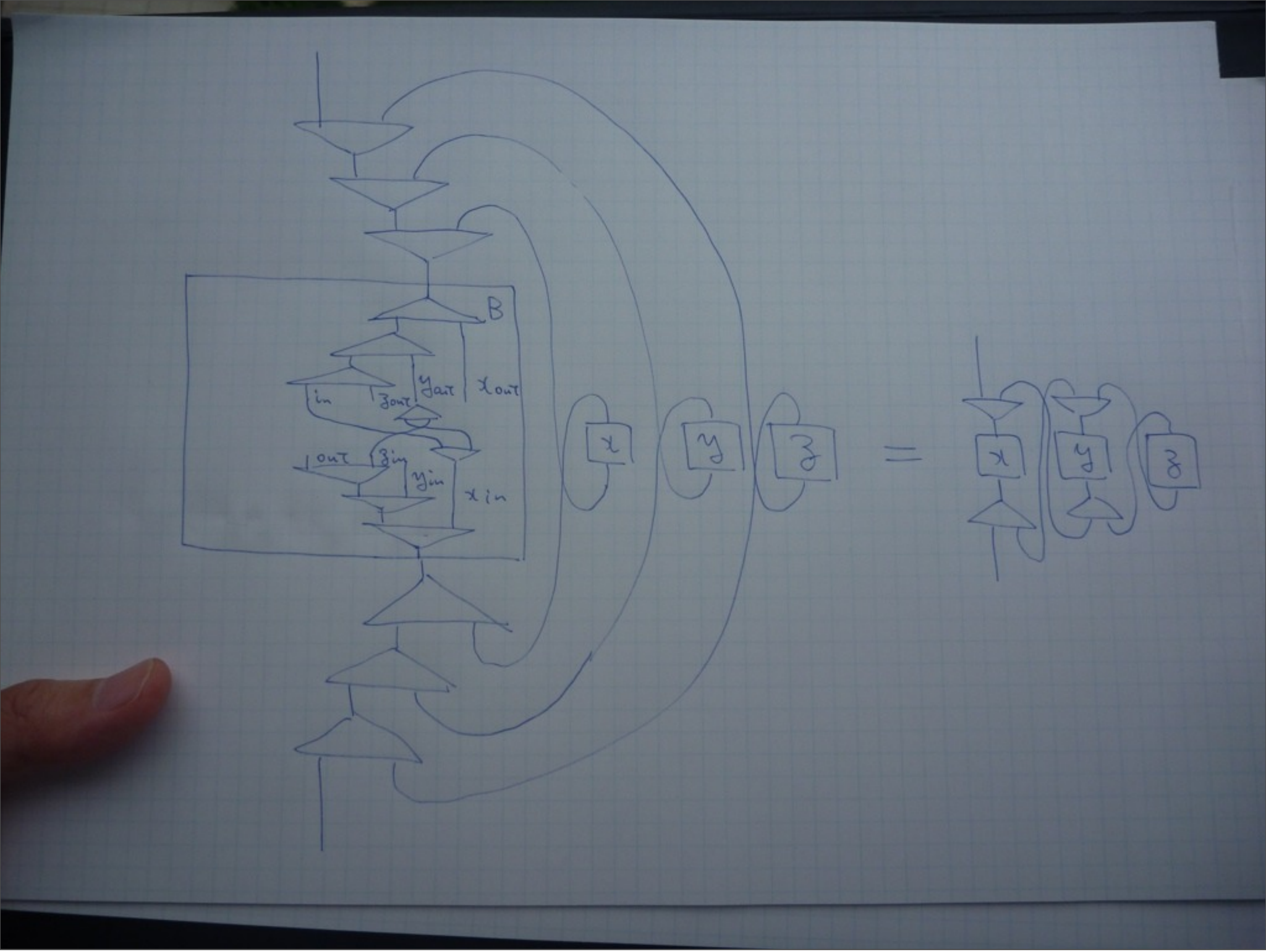
Categorical GoI: Constr. of an LCA

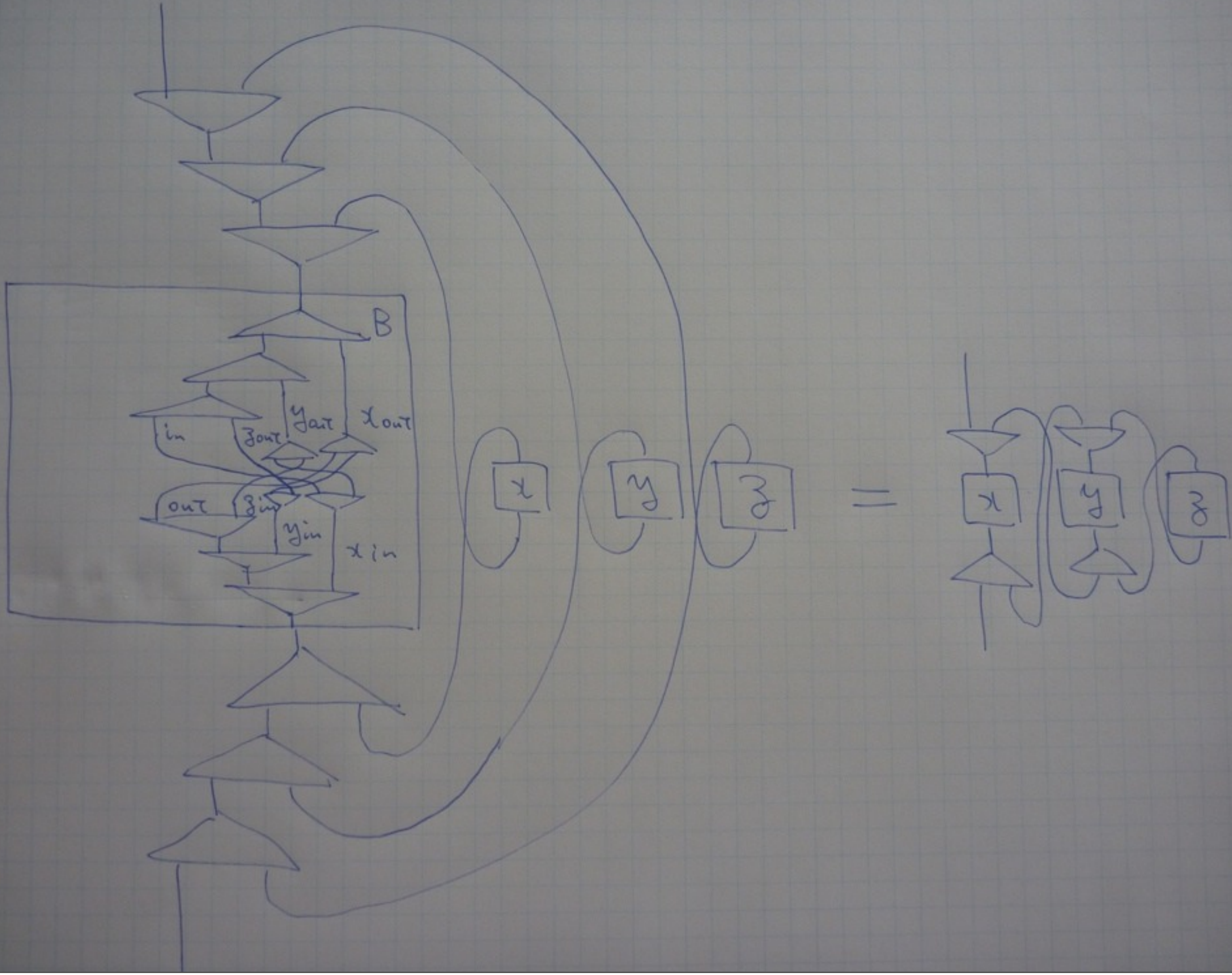
- Combinator $Bxyz = x(yz)$




B
x
y
z
=
x
y
z


B
in
zout
yout
xout
out
zin
yin
xin
x
y
z
=
x
y
z


B
in
z out
y out
x out
out
z in
y in
x in
x
y
z
=
x
y
z


B
in
zout
yout
xout
out
zin
yin
xin
x
y
z
=
x
y
z

Categorical GoI: Constr. of an LCA

- Combinator $Bxyz = x(yz)$

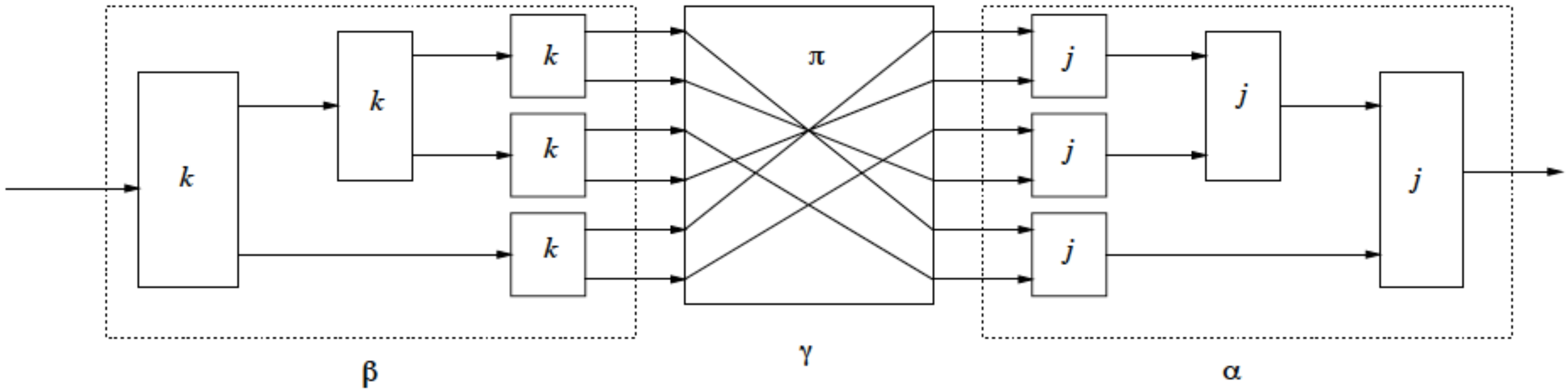


Figure 7: Composition Combinator B

from [AHS02]

Categorical GoI: Constr. of an LCA

- Combinator $Bxyz = x(yz)$

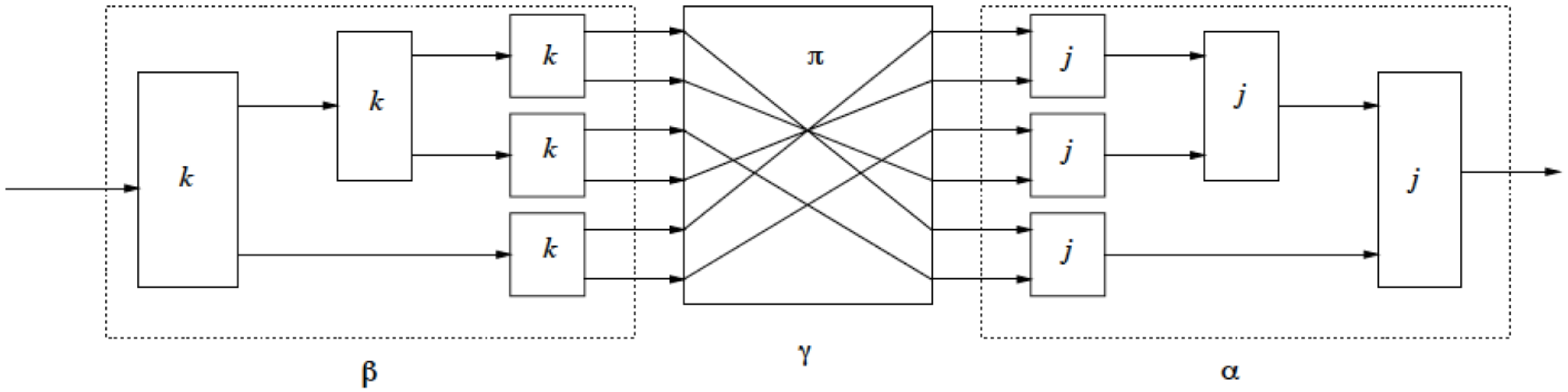


Figure 7: Composition Combinator B

from [AHS02]

Nice dynamic interpretation of (linear) computation!!

Summary: Categorical GoI

Defn. (GoI situation [AHS02])
A *GoI situation* is a triple $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$ where

- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

$$\begin{array}{rcccl}
\boldsymbol{e} & : & \boldsymbol{FF} \lhd \boldsymbol{F} & : & \boldsymbol{e'} \qquad \text{Comultiplication} \\
\boldsymbol{d} & : & \mathbf{id} \lhd \boldsymbol{F} & : & \boldsymbol{d'} \qquad \text{Dereliction} \\
\boldsymbol{c} & : & \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} & : & \boldsymbol{c'} \qquad \text{Contraction} \\
\boldsymbol{w} & : & \boldsymbol{K_I} \lhd \boldsymbol{F} & : & \boldsymbol{w'} \qquad \text{Weakening}
\end{array}$$

  Here $\boldsymbol{K_I}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;
- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

$$\begin{array}{rcccl}
\boldsymbol{j} & : & \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} & : & \boldsymbol{k} \\
 & & \boldsymbol{I} \lhd \boldsymbol{U} & & \\
\boldsymbol{u} & : & \boldsymbol{FU} \lhd \boldsymbol{U} & : & \boldsymbol{v}
\end{array}$$

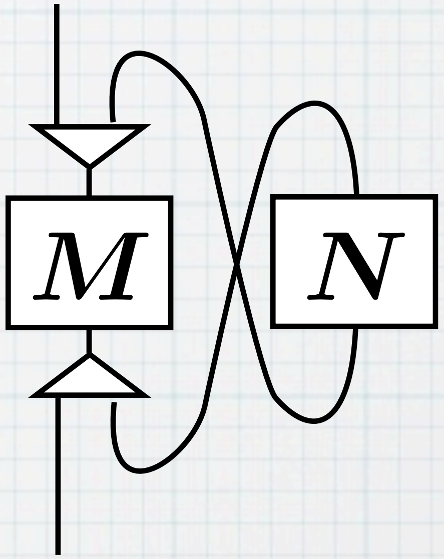
Thm. ([AHS02])
Given a GoI situation $(\mathbb{C}, \boldsymbol{F}, \boldsymbol{U})$, the homset

$$\mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$$

carries a canonical LCA structure.

Why Categorical Generalization?: Examples Other Than Pfn [AHS02]

- Strategy: find a TSMC!



- "Wave-style" examples
  - ⊗ is Cartesian product(-like)
    - in which case,

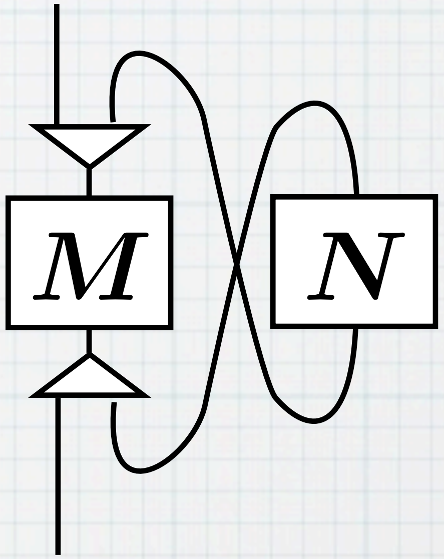
      trace ≈ fixed point operator [Hasegawa/Hyland]
  - An example: $\big(\,(\omega\text{-}\mathbf{Cpo}, \times, 1),\ (\_)^{\mathbb{N}},\ A^{\mathbb{N}}\,\big)$
  - (... less of a dynamic flavor)

Hasuo (Tokyo)

Monday, November 7, 2011

Why Categorical Generalization?: Examples Other Than Pfn [AHS02]

- "Particle-style" examples
  - Obj. X∈C is set-like; ⊗ is coproduct-like
    - The GoI animation is valid



  - Examples:
    - Partial functions $\big(\, (\mathbf{Pfn}, +, \mathbf{0}),\; \mathbb{N}\cdot\_\,,\; \mathbb{N} \,\big)$
    - Binary relations $\big(\, (\mathbf{Rel}, +, \mathbf{0}),\; \mathbb{N}\cdot\_\,,\; \mathbb{N} \,\big)$
    - "Discrete stochastic relations" $\big(\, (\mathbf{DSRel}, +, \mathbf{0}),\; \mathbb{N}\cdot\_\,,\; \mathbb{N} \,\big)$

Why Categorical Generalization?: Examples Other Than Pfn [AHS02]

- **Pfn** (partial functions)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Pfn} \\ \hline\hline X \rightharpoonup Y, \text{ partial function} \\ \hline\hline X \to \mathcal{L}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{L}Y = \{\bot\} + Y$$

- **Rel** (relations)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Rel} \\ \hline\hline R \subseteq X \times Y, \text{ relation} \\ \hline\hline X \to \mathcal{P}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{P} \text{ is the powerset monad}$$

- **DSRel**

$$\begin{array}{c} X \to Y \text{ in } \mathbf{DSRel} \\ \hline\hline X \to \mathcal{D}Y \text{ in } \mathbf{Sets} \end{array}$$

$$\text{where } \mathcal{D}Y = \{d : Y \to [0,1] \mid \sum_{y} d(y) \leq 1\}$$

Why Catego
Examples Other Than Pfn [AHS02]

Categories of sets and
(functions with different branching/partiality)

- **Pfn** (partial functions)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Pfn} \\ \hline\hline X \rightharpoonup Y, \text{ partial function} \\ \hline\hline X \to \mathcal{L}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{L}Y = \{\perp\} + Y$$

- **Rel** (relations)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Rel} \\ \hline\hline R \subseteq X \times Y, \text{ relation} \\ \hline\hline X \to \mathcal{P}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{P} \text{ is the powerset monad}$$

- **DSRel**

$$\begin{array}{c} X \to Y \text{ in } \mathbf{DSRel} \\ \hline\hline X \to \mathcal{D}Y \text{ in } \mathbf{Sets} \end{array}$$

$$\text{where } \mathcal{D}Y = \{d : Y \to [0,1] \mid \sum_{y} d(y) \leq 1\}$$

Hasuo (Tokyo)

Monday, November 7, 2011

Why Catego
Examples Other Than Fun [AHS02]

Categories of sets and
(functions with different branching/partiality)

- **Pfn** (partial functions)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Pfn} \\ \hline\hline X \rightharpoonup Y, \text{ partial function} \\ \hline\hline X \to \mathcal{L}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{L}Y = \{\perp\} + Y$$

(Potential) non-termination

- **Rel** (relations)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Rel} \\ \hline\hline R \subseteq X \times Y, \text{ relation} \\ \hline\hline X \to \mathcal{P}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{P} \text{ is the powerset monad}$$

Non-determinism

- **DSRel**

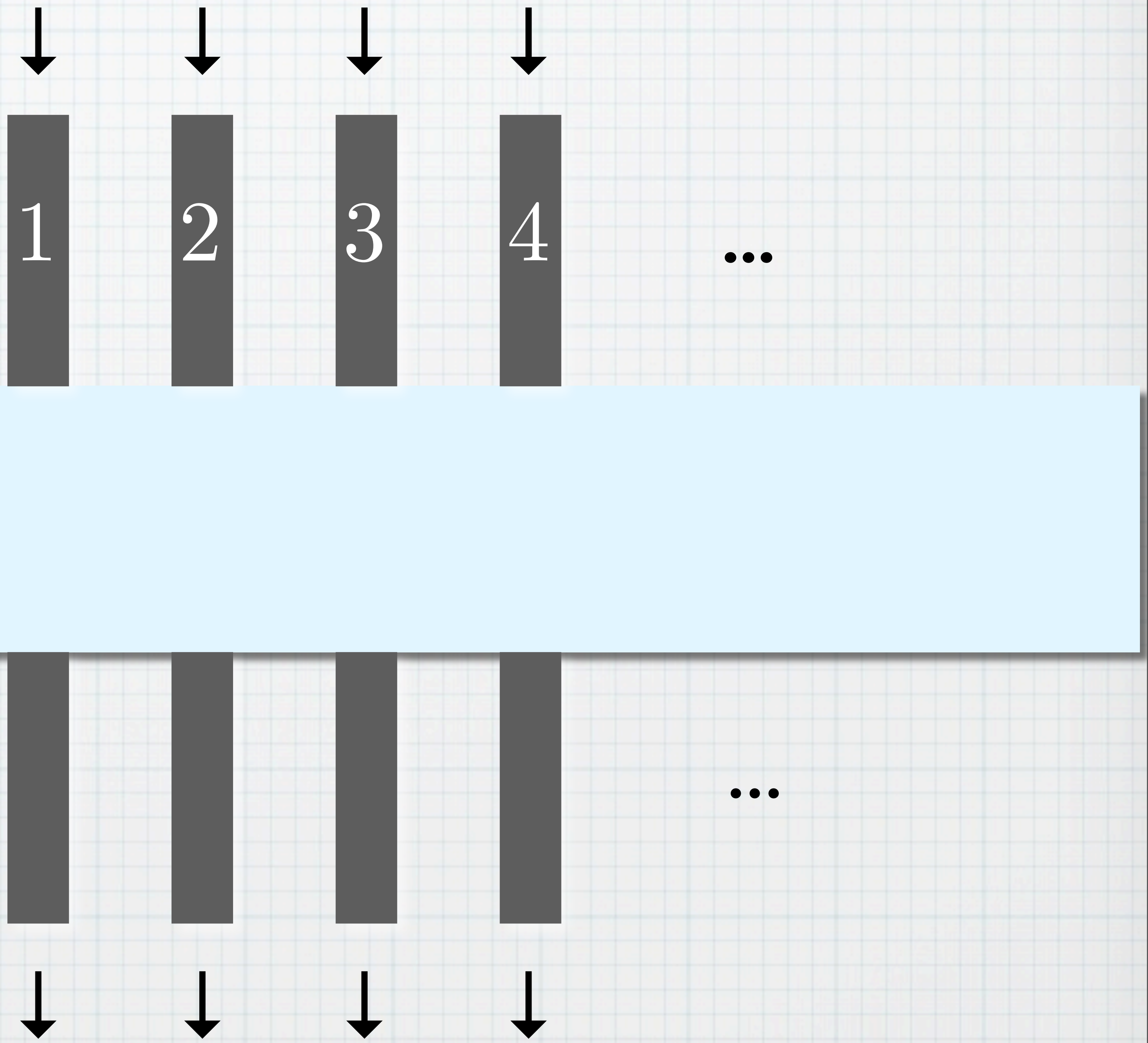
$$\begin{array}{c} X \to Y \text{ in } \mathbf{DSRel} \\ \hline\hline X \to \mathcal{D}Y \text{ in } \mathbf{Sets} \end{array}$$

$$\text{where } \mathcal{D}Y = \{d : Y \to [0,1] \mid \sum_{y} d(y) \le 1\}$$

Probabilistic branching

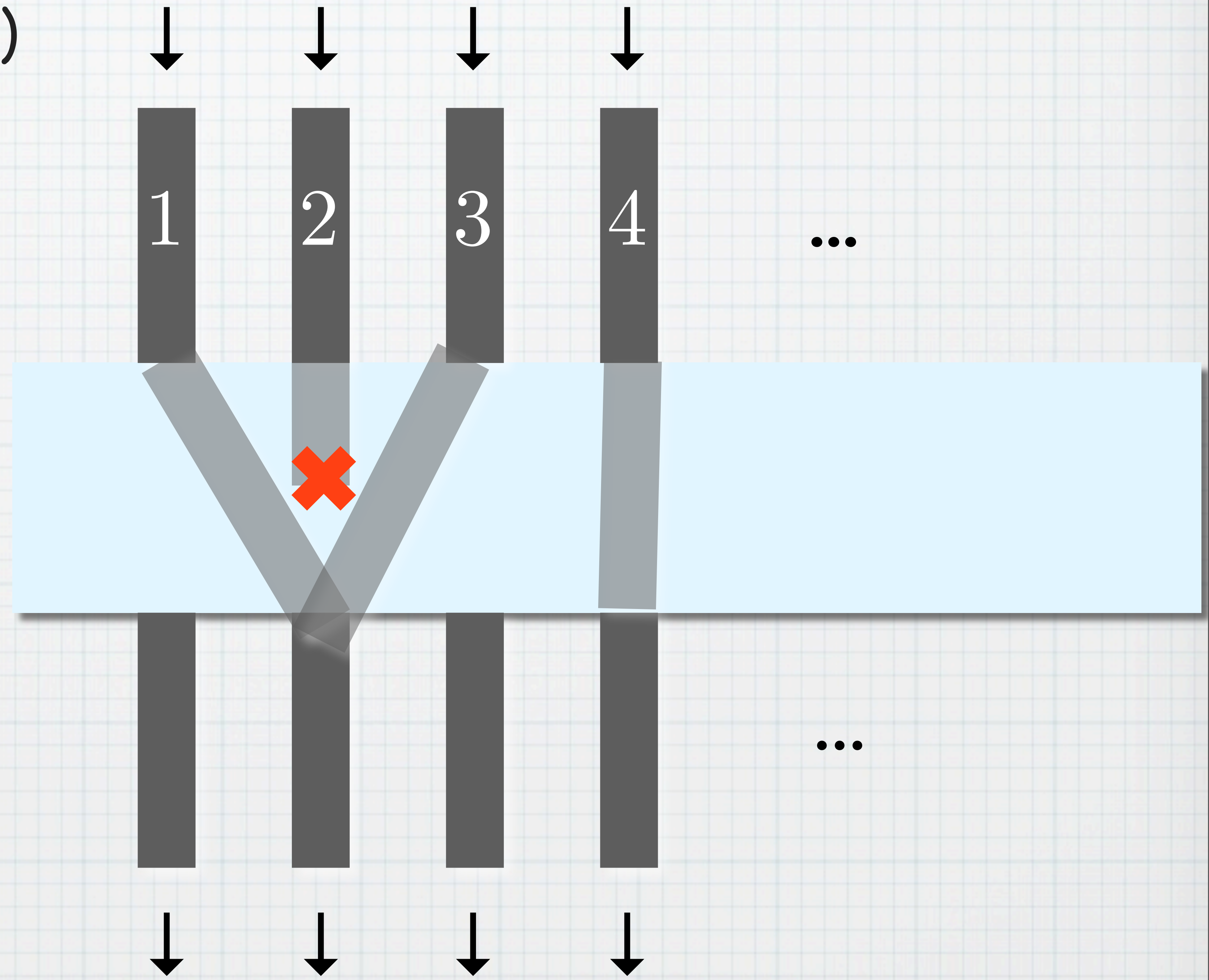
Different Branching in The GoI Animation

- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Different Branching in The GoI Animation

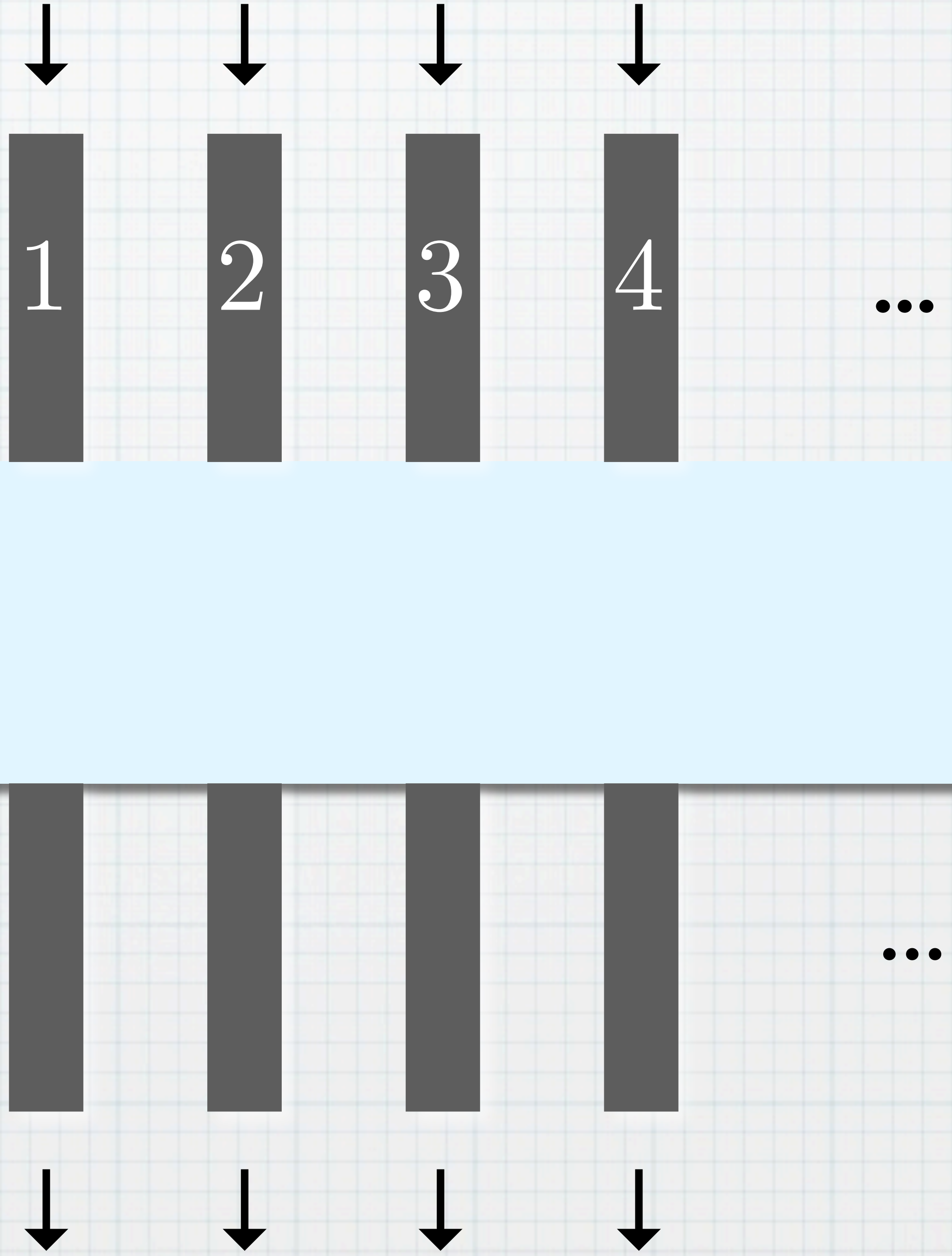
- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Hasuo (Tokyo)

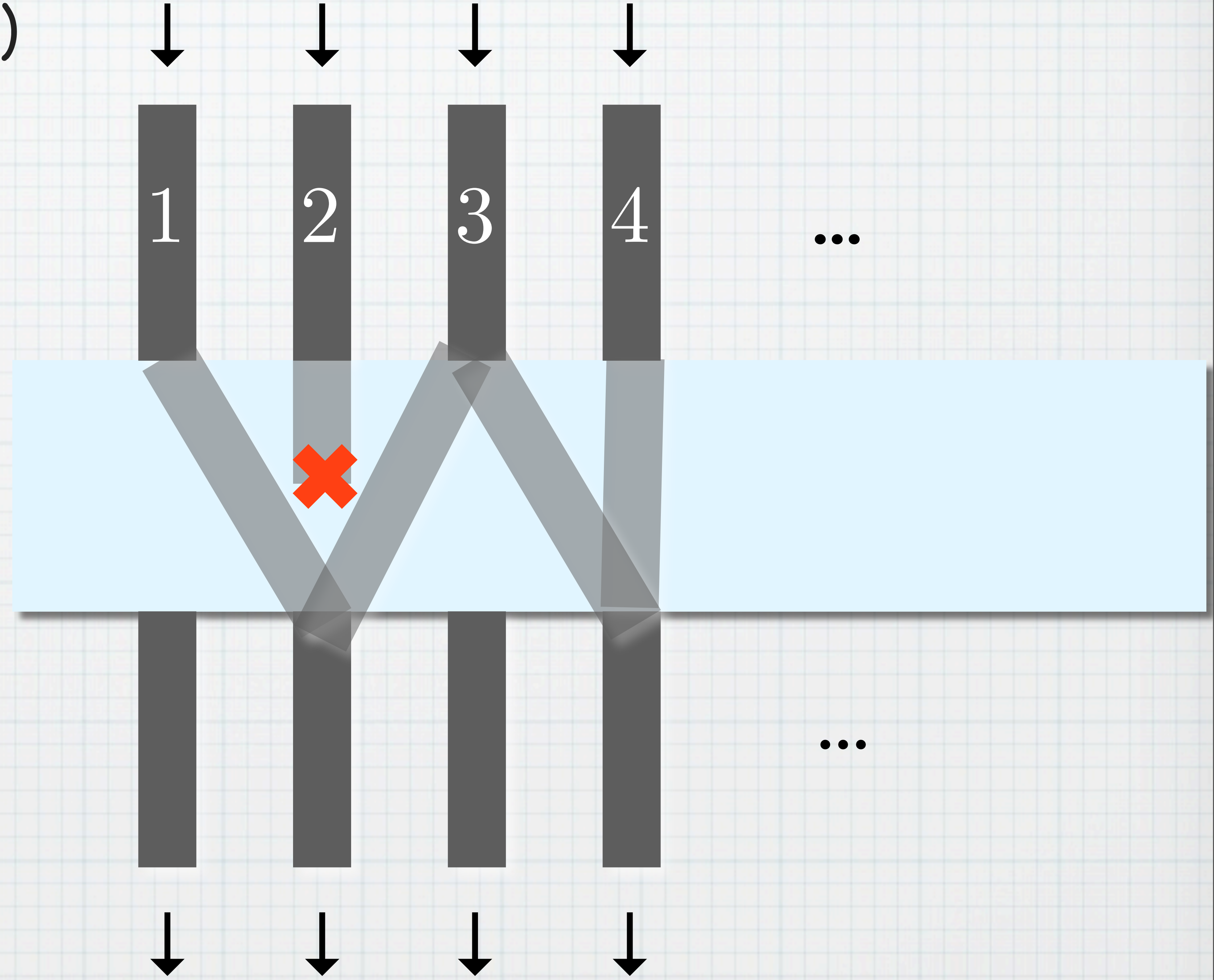
Different Branching in The GoI Animation

- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



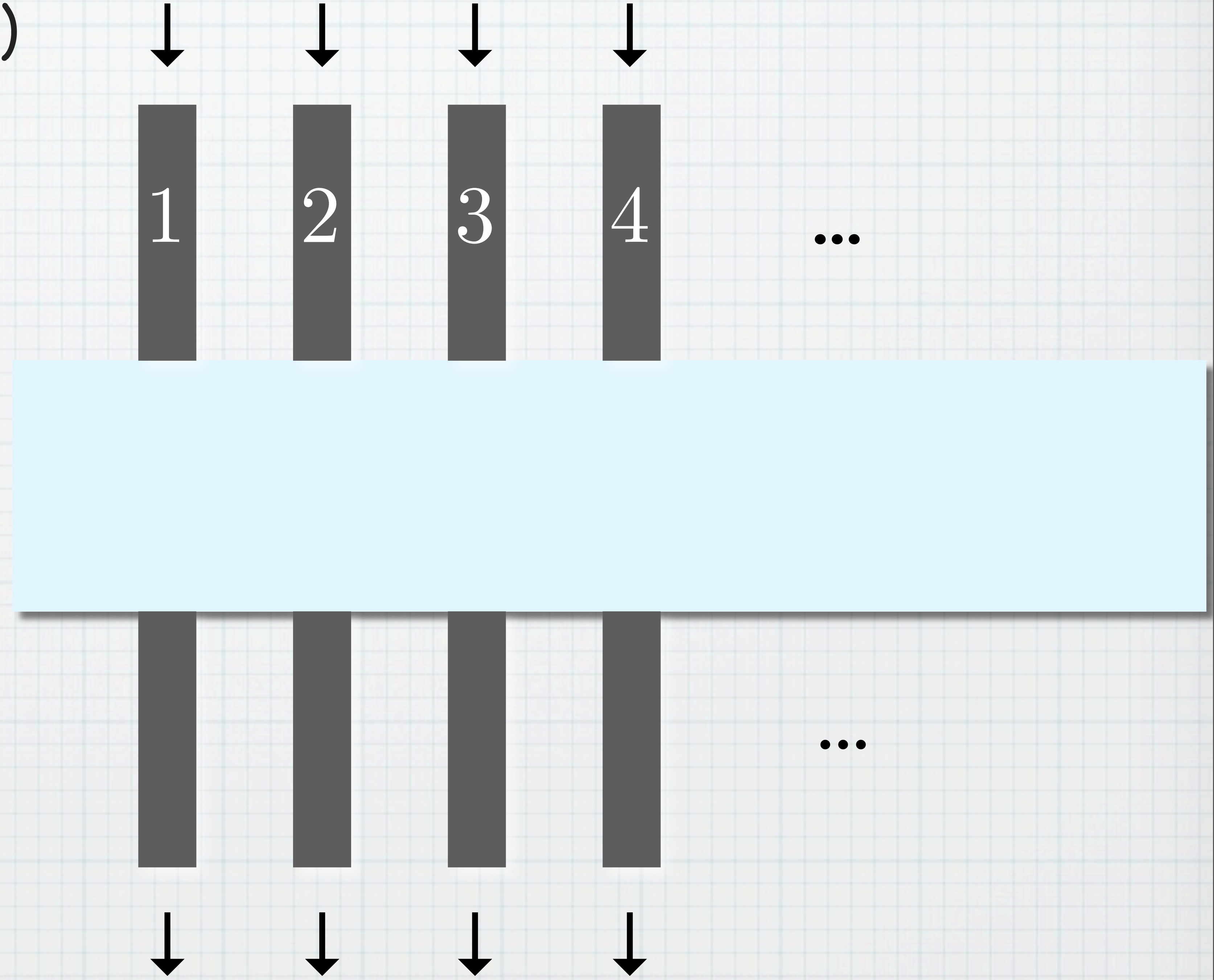
Different Branching in The GoI Animation

- **Pfn** (partial functions)
  - Pipes can be stuck
- ➜ **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Different Branching in The GoI Animation

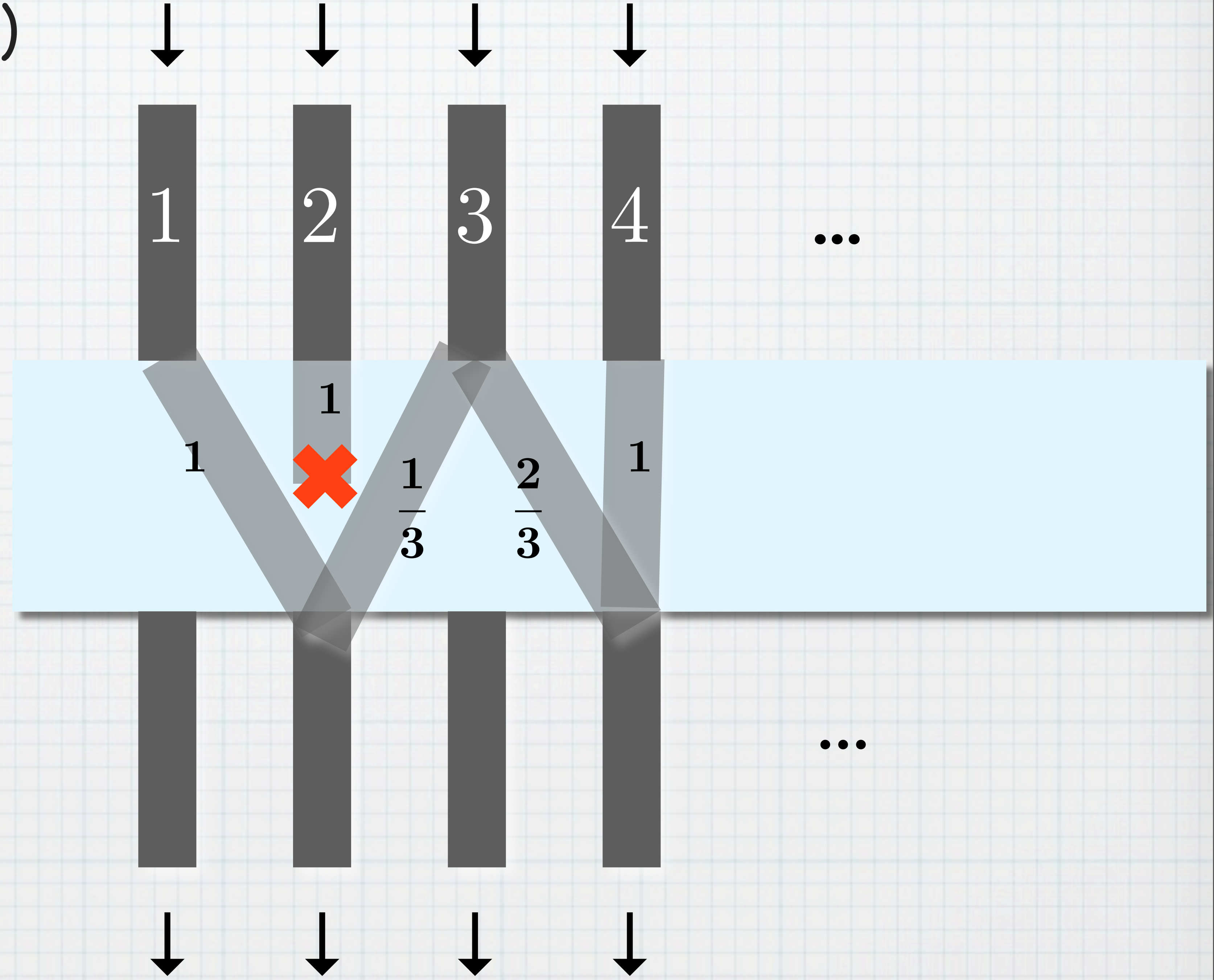
- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Hasuo (Tokyo)

Different Branching in The GoI Animation

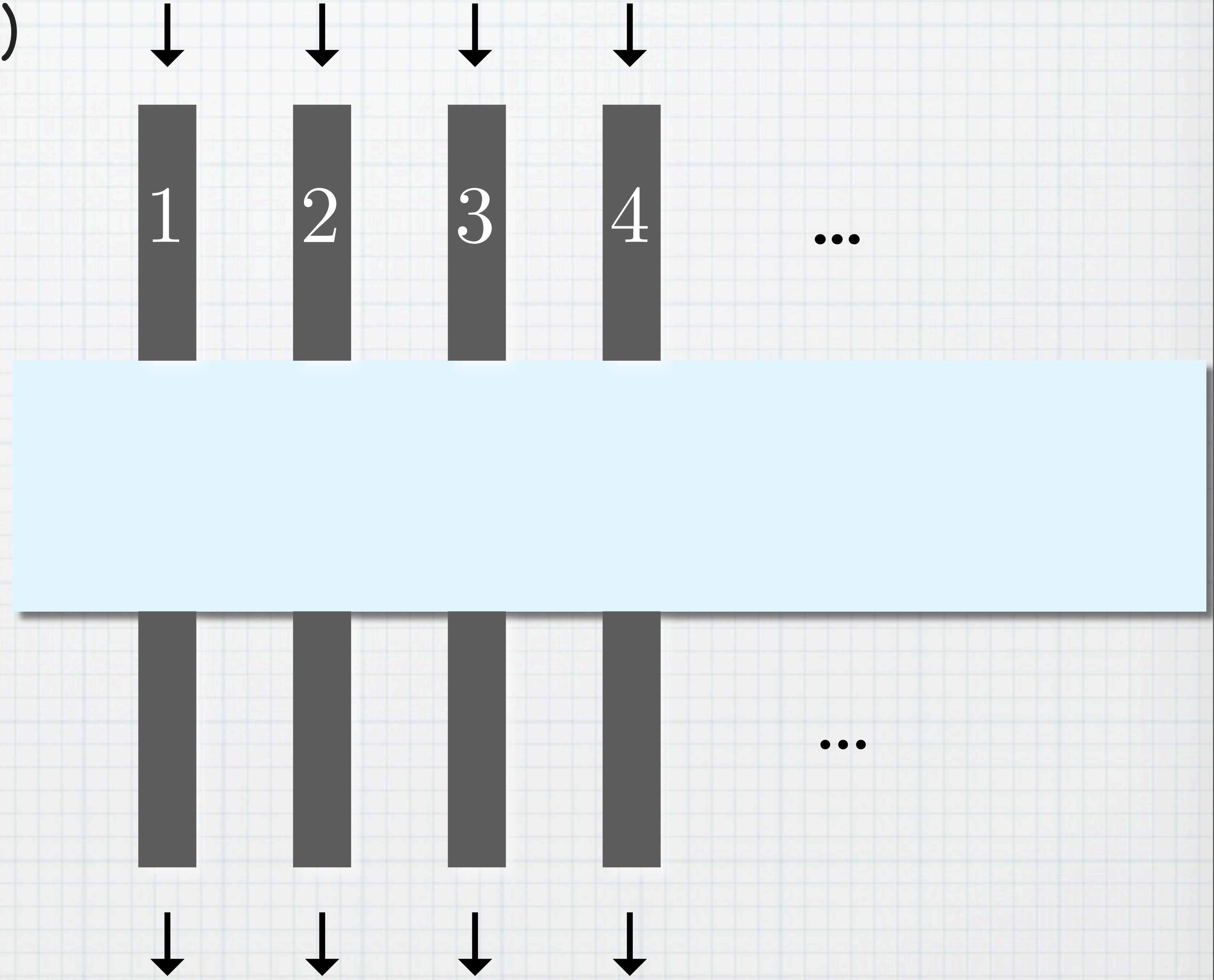
- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Hasuo (Tokyo)

Different Branching in The GoI Animation

- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Hasuo (Tokyo)

Why Categorical Generalization?: Examples Other Than Pfn [AHS02]

- **Pfn** (partial functions)

$$\cfrac{\cfrac{X \to Y \text{ in } \mathbf{Pfn}}{X \rightharpoonup Y, \text{ partial function}}}{X \to \mathcal{L}Y \text{ in } \mathbf{Sets}} \quad \text{where } \mathcal{L}Y = \{\bot\} + Y$$

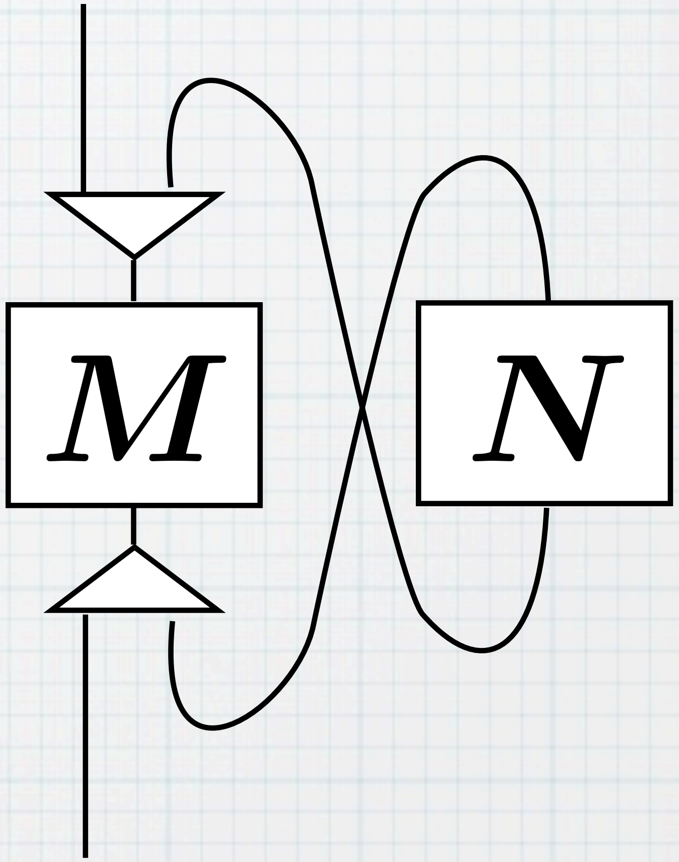
- **Rel** (relations)

$$\cfrac{\cfrac{X \to Y \text{ in } \mathbf{Rel}}{R \subseteq X \times Y, \text{ relation}}}{X \to \mathcal{P}Y \text{ in } \mathbf{Sets}} \quad \text{where } \mathcal{P} \text{ is the powerset monad}$$

- **DSRel**

$$\cfrac{X \to Y \text{ in } \mathbf{DSRel}}{X \to \mathcal{D}Y \text{ in } \mathbf{Sets}}$$

where $\mathcal{D}Y = \{d : Y \to [0,1] \mid \sum_y d(y) \leq 1\}$



Why Categorical Generalization?: Examples Other Than Pfn [AHS02]

- **Pfn** (partial functions)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Pfn} \\ \hline\hline X \rightharpoonup Y, \text{ partial function} \\ \hline\hline X \to \mathcal{L}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{L}Y = \{\bot\} + Y$$

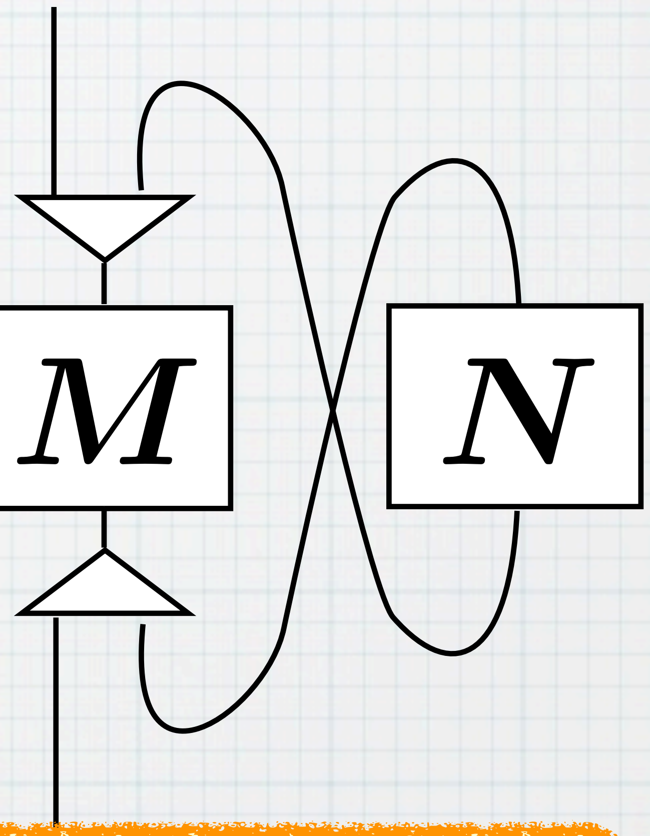
- **Rel** (relations)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Rel} \\ \hline\hline R \subseteq X \times Y, \text{ relation} \\ \hline\hline X \to \mathcal{P}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{P} \text{ is the powerset monad}$$

- **DSRel**

$$\begin{array}{c} X \to Y \text{ in } \mathbf{DSRel} \\ \hline\hline X \to \mathcal{D}Y \text{ in } \mathbf{Sets} \end{array}$$

$$\text{where } \mathcal{D}Y = \{d : Y \to [0,1] \mid \sum_{y} d(y) \le 1\}$$



Essential to have subdistribution, for infinite loops

yo)

The Coauthor

- Naohiko Hoshino
  - DSc (Kyoto, 2011)
    - Supervisor: Masahito "Hassei" Hasegawa
  - Currently at RIMS, Kyoto U.
    - http://www.kurims.kyoto-u.ac.jp/~naophiko/

A Coalgebraic View

- Theory of coalgebra =
  Categorical theory of state-based dynamic systems (LTS, automaton, Markov chain, ...)

- In [Hasuo, Jacobs, Sokolova '07]:
  - Coalgebras in a Kleisli category $Kl(B)$

$$\begin{array}{c} \boldsymbol{X} \to \boldsymbol{Y} \ \text{ in } \mathcal{K}\ell(B) \\ \hline\hline \boldsymbol{X} \to \boldsymbol{BY} \ \text{ in } \mathbf{Sets} \end{array}$$

  - ➜ Generic theory of "trace semantics"

Why Catego
Examples Other Than Fun

Categories of sets and (functions with different branching/partiality)

- **Pfn** (partial functions)

$$\begin{array}{c} X \to Y \ \text{ in } \mathbf{Pfn} \\ \hline\hline X \rightharpoonup Y, \text{ partial function} \\ \hline\hline X \to \mathcal{L}Y \ \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{L}Y = \{\bot\} + Y$$

(Potential) non-termination

- **Rel** (relations)

$$\begin{array}{c} X \to Y \ \text{ in } \mathbf{Rel} \\ \hline\hline R \subseteq X \times Y, \text{ relation} \\ \hline\hline X \to \mathcal{P}Y \ \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{P} \text{ is the powerset monad}$$

Non-determinism

- **DSRel**

$$\begin{array}{c} X \to Y \ \text{ in } \mathbf{DSRel} \\ \hline\hline X \to \mathcal{D}Y \ \text{ in } \mathbf{Sets} \end{array}$$

$$\text{where } \mathcal{D}Y = \{d : Y \to [0,1] \mid \sum_{y} d(y) \leq 1\}$$

Probabilistic branching

Why Catego
Examples

$Kl(B)$ for different branching monads B

- **Pfn** (partial functions)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Pfn} \\ \hline\hline X \rightharpoonup Y, \text{ partial function} \\ \hline\hline X \to \mathcal{L}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{L}Y = \{\bot\} + Y$$

(Potential) non-termination

- **Rel** (relations)

$$\begin{array}{c} X \to Y \text{ in } \mathbf{Rel} \\ \hline\hline R \subseteq X \times Y, \text{ relation} \\ \hline\hline X \to \mathcal{P}Y \text{ in } \mathbf{Sets} \end{array} \quad \text{where } \mathcal{P} \text{ is the powerset monad}$$

Non-determinism

- **DSRel**

$$\begin{array}{c} X \to Y \text{ in } \mathbf{DSRel} \\ \hline\hline X \to \mathcal{D}Y \text{ in } \mathbf{Sets} \end{array}$$

$$\text{where } \mathcal{D}Y = \{d : Y \to [0,1] \mid \sum_y d(y) \leq 1\}$$

Probabilistic branching

Branching Monad: Source of Particle-Style GoI Situations

Thm. ([Jacobs,CMCS10])
Given a "branching monad" $\boldsymbol{B}$ on **Sets**, the monoidal category

$$(\boldsymbol{\mathcal{K}\ell}(\boldsymbol{B}), +, \mathbf{0})$$

is

- a *unique decomposition category* [Haghverdi,PhD00], hence is
- a traced symmetric monoidal category.

Cor.
$\big((\boldsymbol{\mathcal{K}\ell}(\boldsymbol{B}), +, \mathbf{0}),\ \mathbb{N}\cdot\_\ ,\ \mathbb{N} \big)$ is a GoI situation.

Branching Monad: Source of Particle-Style GoI Situations

Thm. ([Jacobs,CMCS10])
Given a "branching monad" $\boldsymbol{B}$ on **Sets**, the monoidal category

$$(\boldsymbol{\mathcal{K}\ell(B), +, 0})$$

is

- a *unique decomposition category* [Haghverdi,PhD00], hence is
- a traced symmetric monoidal category.

Cor.
$\big(\,(\boldsymbol{\mathcal{K}\ell(B), +, 0}),\ \mathbb{N}\cdot\_\,,\ \mathbb{N}\,\big)$ is a GoI situation.

(Roughly) monads in
[Hasuo, Jacobs, Sokolova '07]

- ＊ Kl(B) is $\mathrm{Cpo}_{\perp}$-enriched
- ＊ like $\mathcal{L}$, $\mathcal{P}$, $\mathcal{D}$

Branching Monad: Source of Particle-Style GoI Situations

Thm. ([Jacobs,CMCS10])
Given a "branching monad" $\boldsymbol{B}$ on **Sets**, the monoidal category

$$(\boldsymbol{\mathcal{K}\ell(B)}, +, \mathbf{0})$$

is

- a *unique decomposition category* [Haghverdi,PhD00], hence is
- a traced symmetric monoidal category.

Cor.
$\big(\,(\boldsymbol{\mathcal{K}\ell(B)}, +, \mathbf{0}),\ \mathbb{N}\cdot\_\,,\ \mathbb{N}\,\big)$ is a GoI situation.

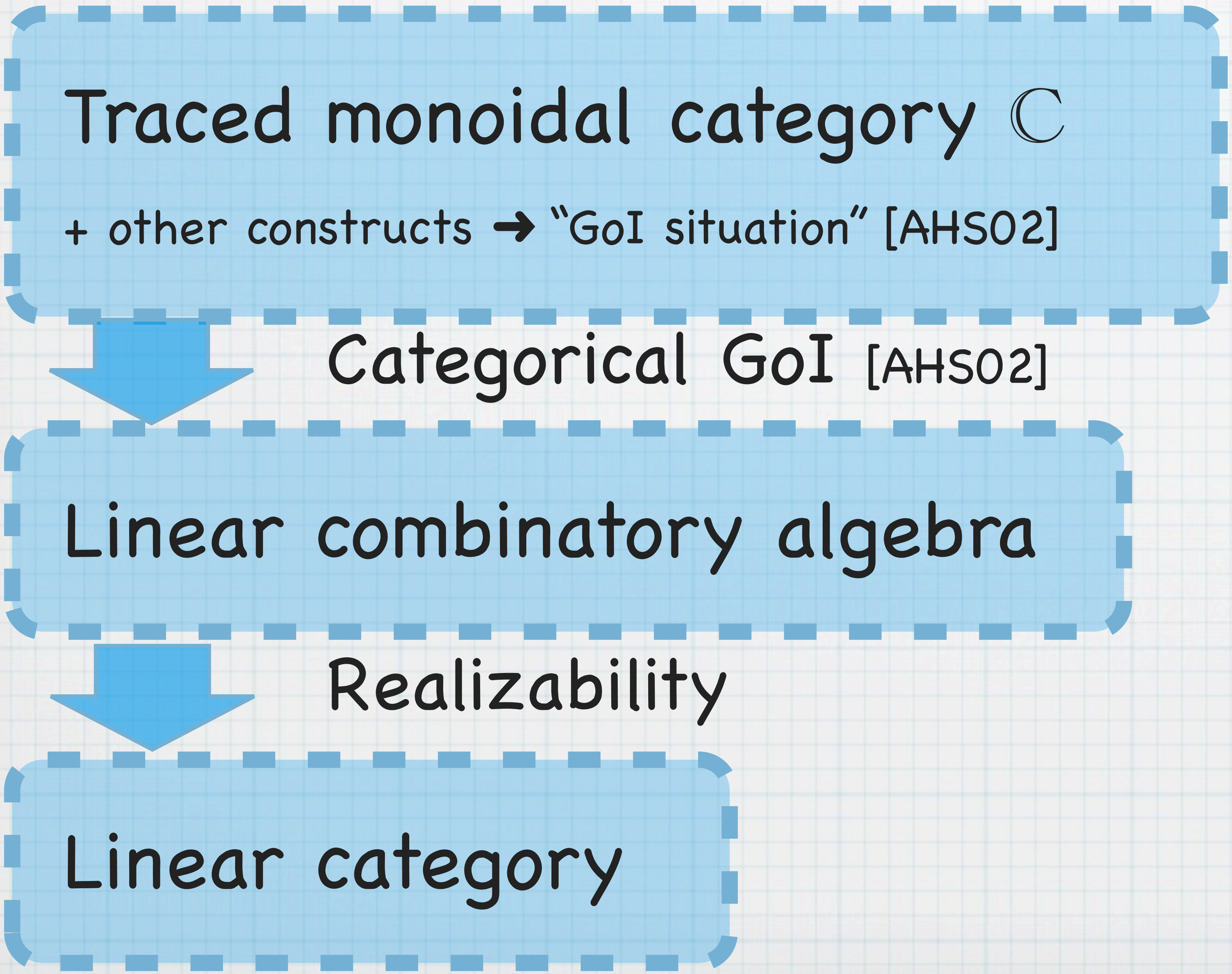
(Roughly) monads in
[Hasuo, Jacobs, Sokolova '07]

- Kl(B) is $\mathrm{Cpo}_{\perp}$-enriched
- like $\mathcal{L}$, $\mathcal{P}$, $\mathcal{D}$

Particle-style: trace via the execution formula

$$\mathbf{tr}(f) = f_{XY} \sqcup \left(\coprod_{n\in\mathbb{N}} f_{ZY} \circ (f_{ZZ})^n \circ f_{XZ} \right)$$

The Categorical GoI Workflow



Traced monoidal category $\mathbb{C}$

+ other constructs ➔ “GoI situation” [AHS02]

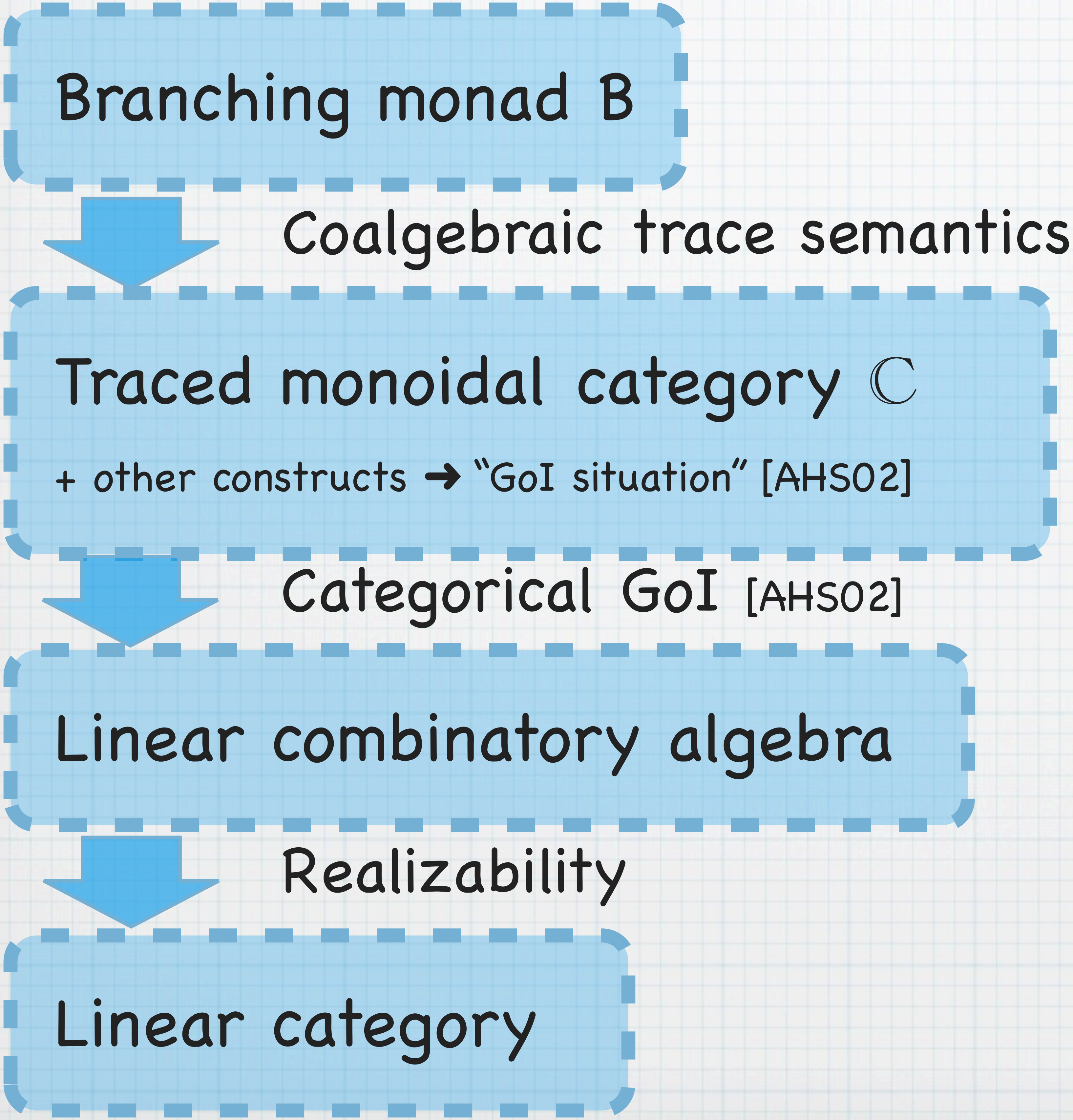
Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

The Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➔ "GoI situation" [AHS02]

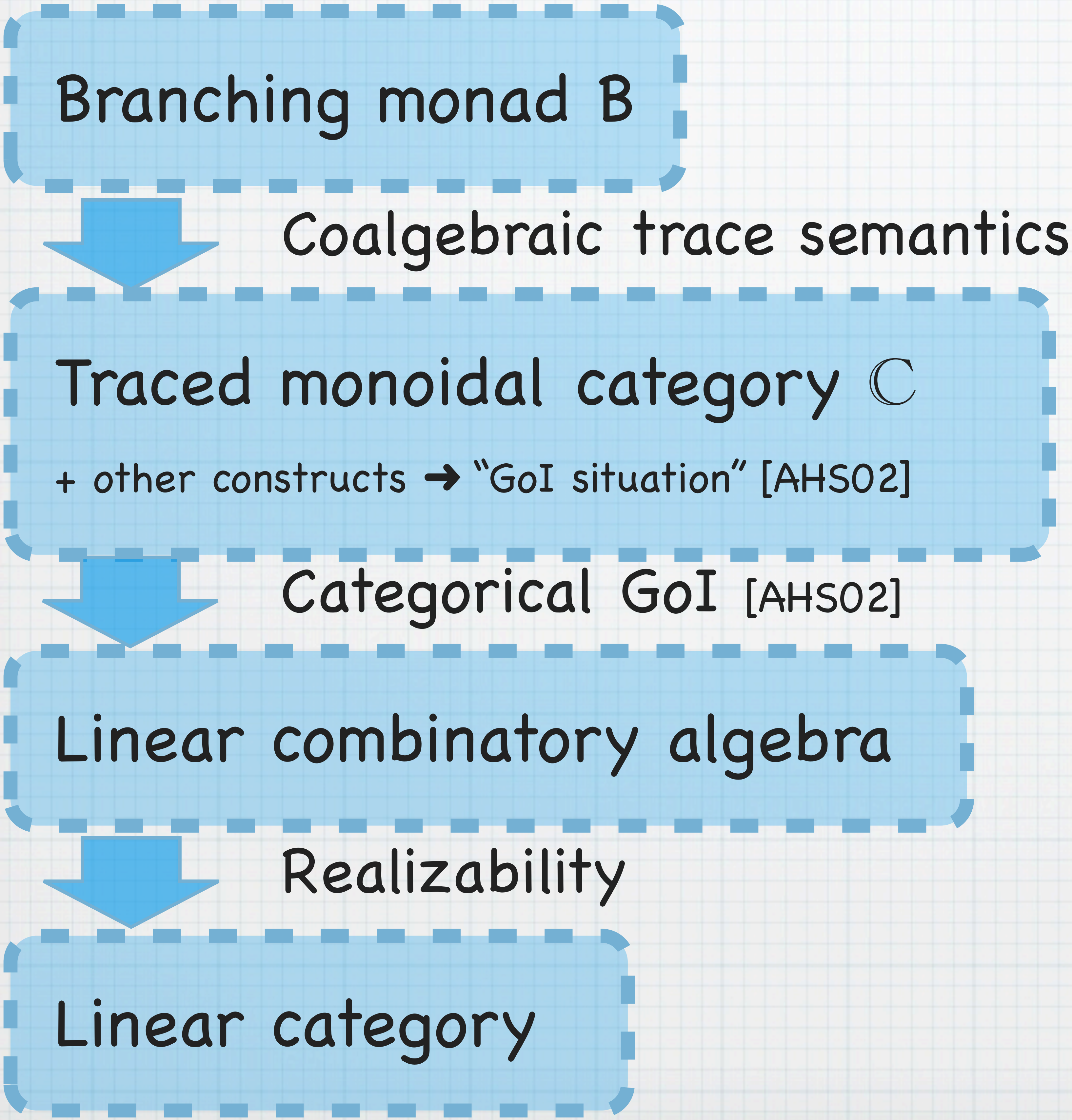
Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

The Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ “GoI situation” [AHS02]

Categorical GoI [AHS02]

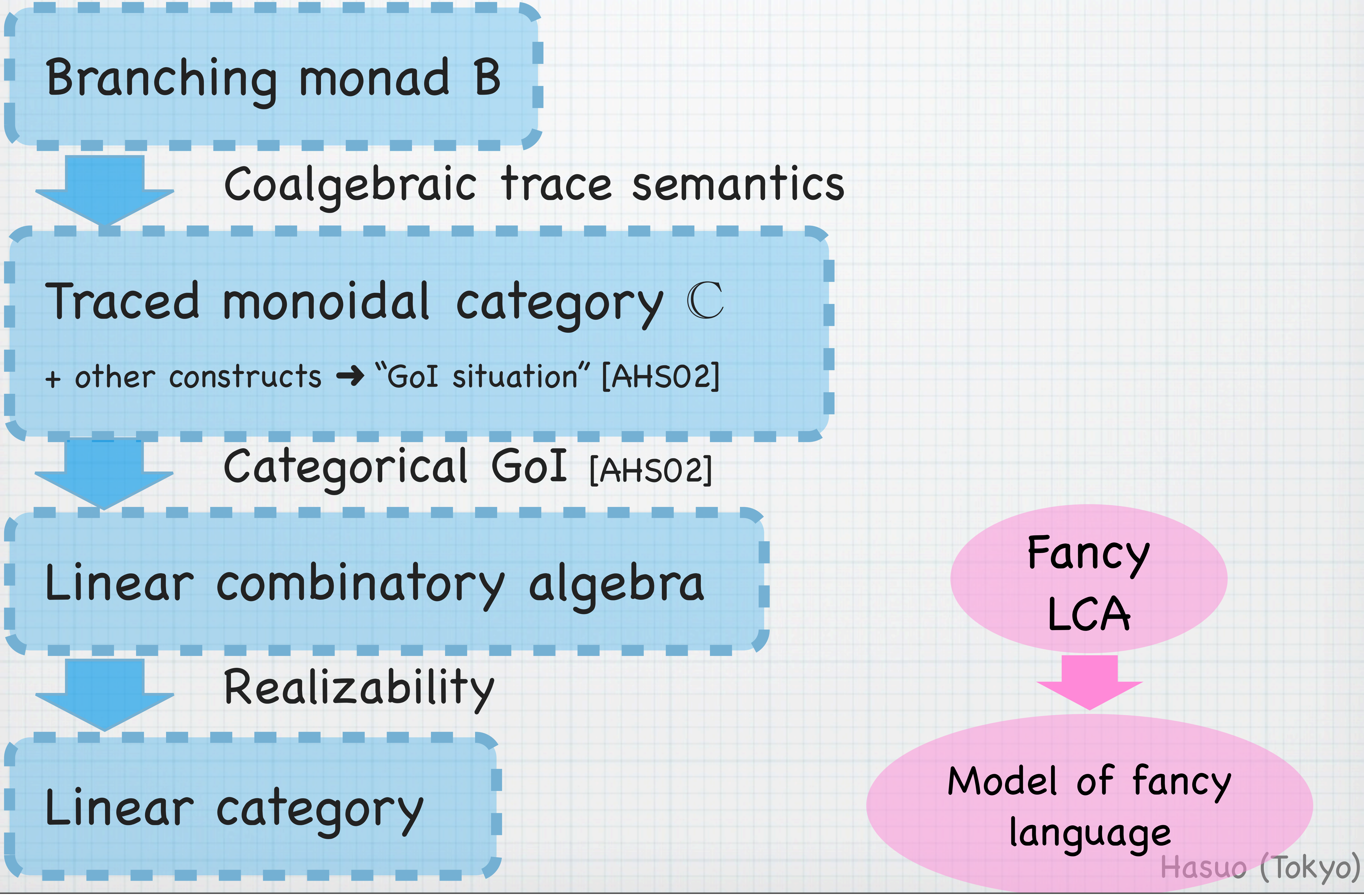
Linear combinatory algebra

Realizability

Linear category

Model of fancy language

The Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

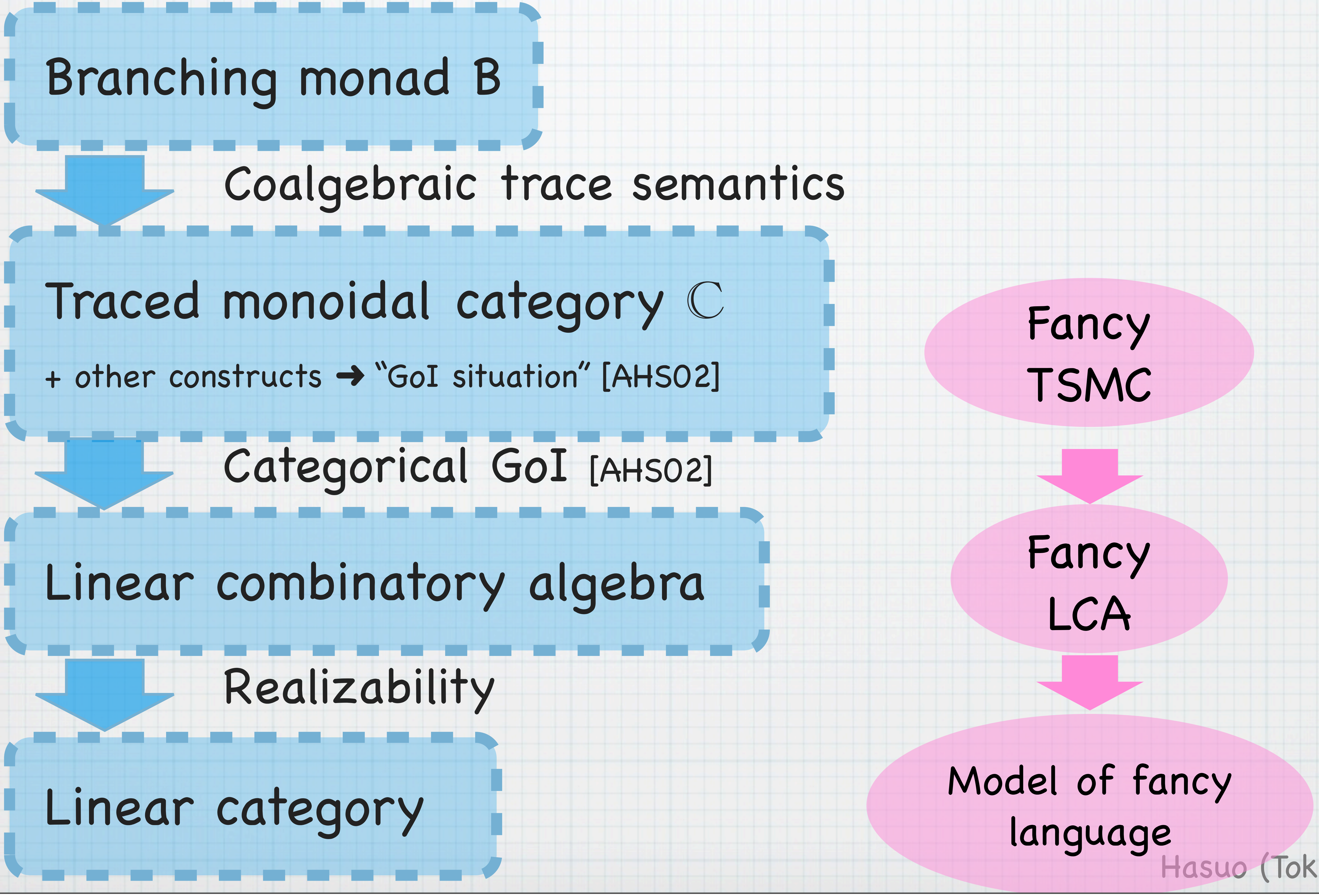
Realizability

Linear category

Fancy LCA

Model of fancy language

The Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

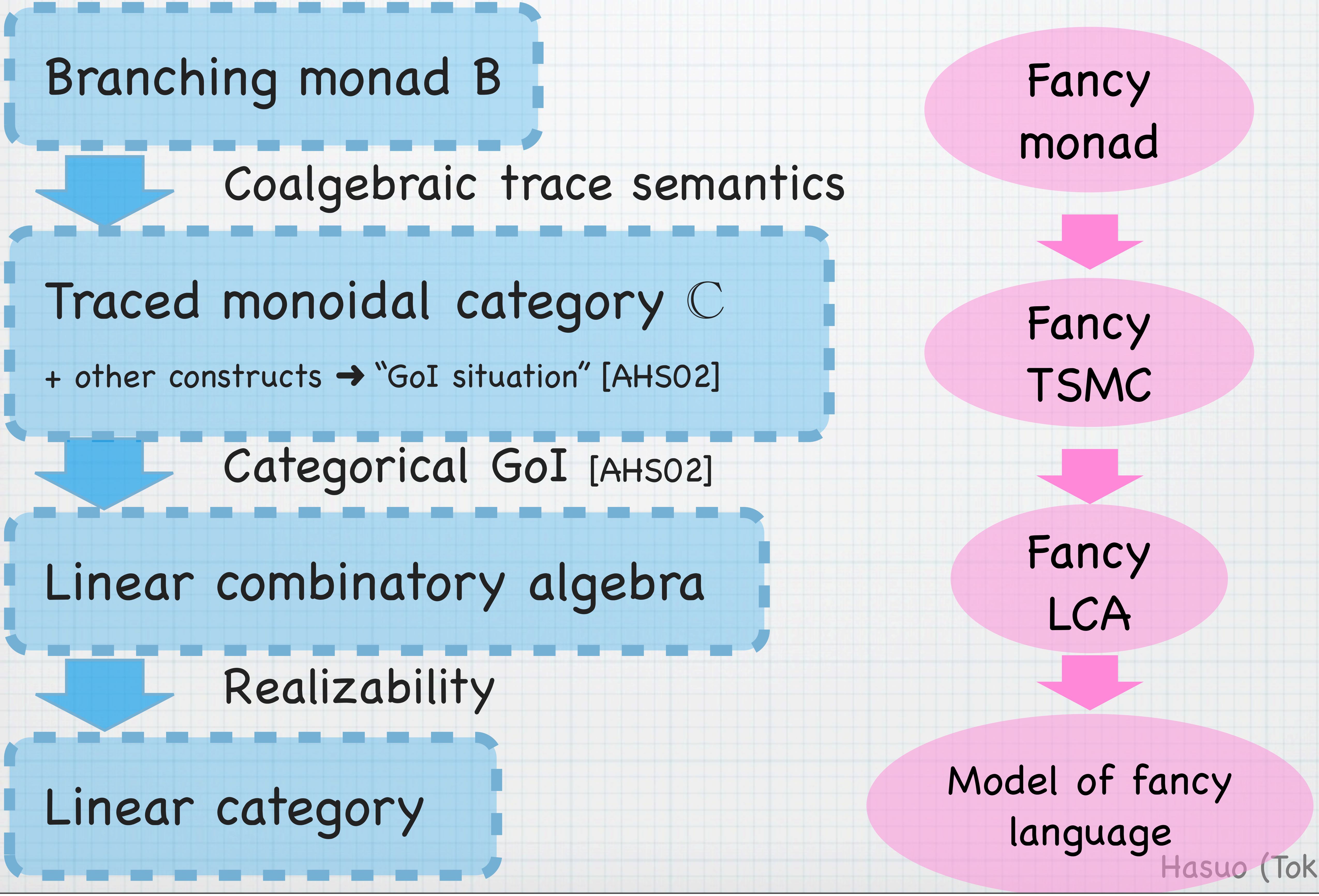
Linear category

Fancy TSMC

Fancy LCA

Model of fancy language

The Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➔ "GoI situation" [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

Fancy monad

Fancy TSMC

Fancy LCA

Model of fancy language

What is Fancy, Nowadays?

What is Fancy, Nowadays?

- Biology?

What is Fancy, Nowadays?

- Biology?
- Hybrid systems?
  - Both discrete and continuous data, typically in **cyber-physical systems** (CPS)
  - ➜ Our approach via **non-standard analysis** [Suenaga, Hasuo ICALP'11]

What is Fancy, Nowadays?

- Biology?
- Hybrid systems?
  - Both discrete and continuous data, typically in **cyber-physical systems** (CPS)
  - ➜ Our approach via **non-standard analysis** [Suenaga, Hasuo ICALP'11]
- Quantum?
  - Yes this worked!

Part 3

Phil Scott.
Tutorial on Geometry of Interaction, FMCS 2004.
Page 47/47

Future Directions

- GoI 2 : Non-converging algms (untyped λ-calc / PCF)
  - uses more topological info on operatn algs
- GoI 3: uses additives & additive proof nets —
- GoI 4 (last month) : von Neumann algebras : $Ex(f, \tau)$ for f arb (not necessarily coming from proof)
- Quantum GoI ?

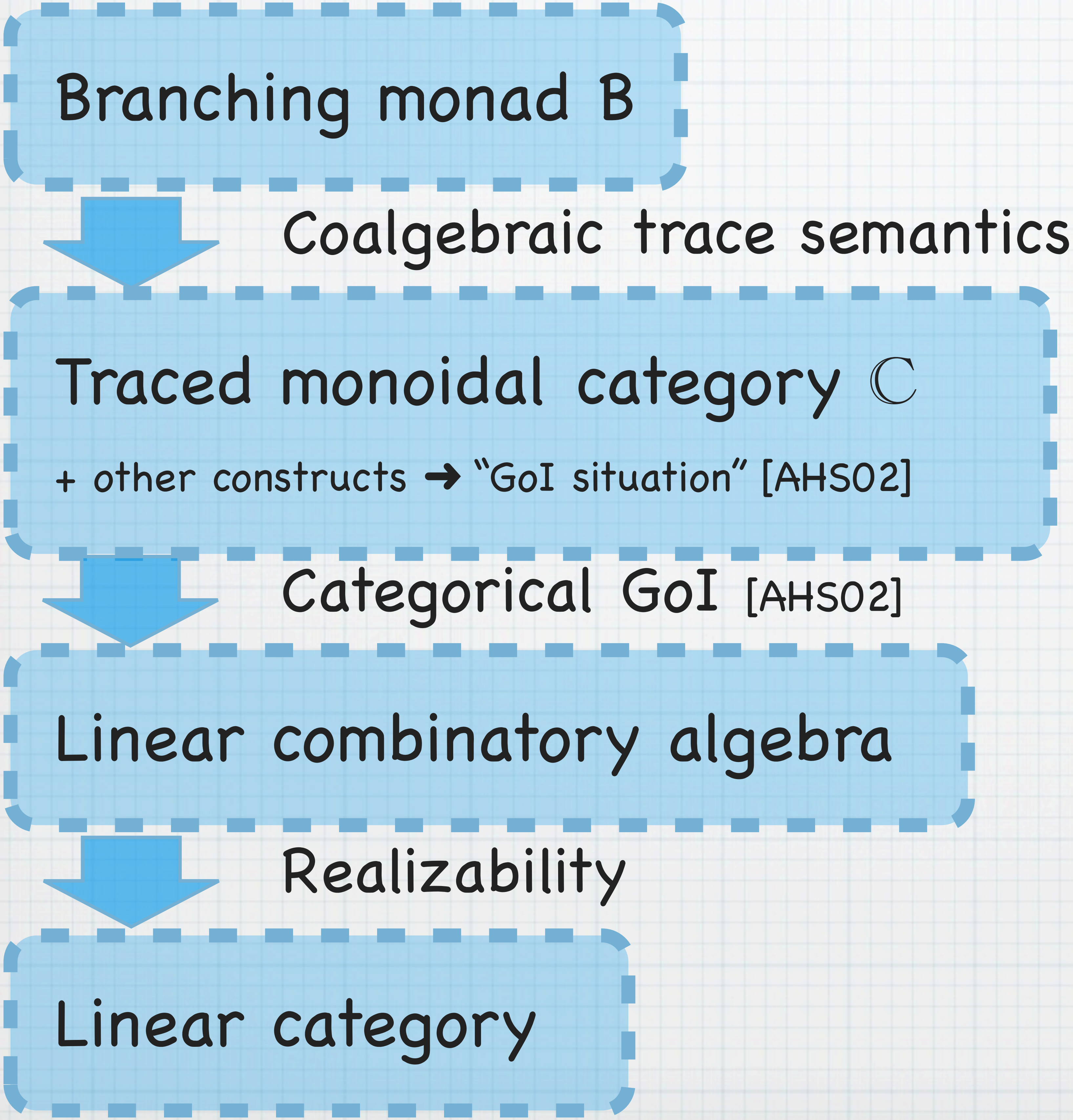
Part 3

Phil Scott.
Tutorial on Geometry of Interaction, FMCS 2004.
Page 47/47

Future Directions

- GoI 2 : Non-converging algms (untyped λ-calc / PCF)
  - uses more topological info on operatr algs
- GoI 3: uses additives & additive proof nets —
- GoI 4 (last mnth): Von Neumann Algebras: $Ex(f, \tau)$ for f arb (not necessarily coming from proof)
- $\boxed{\text{Quantum GoI}}$?

The Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ “GoI situation” [AHS02]

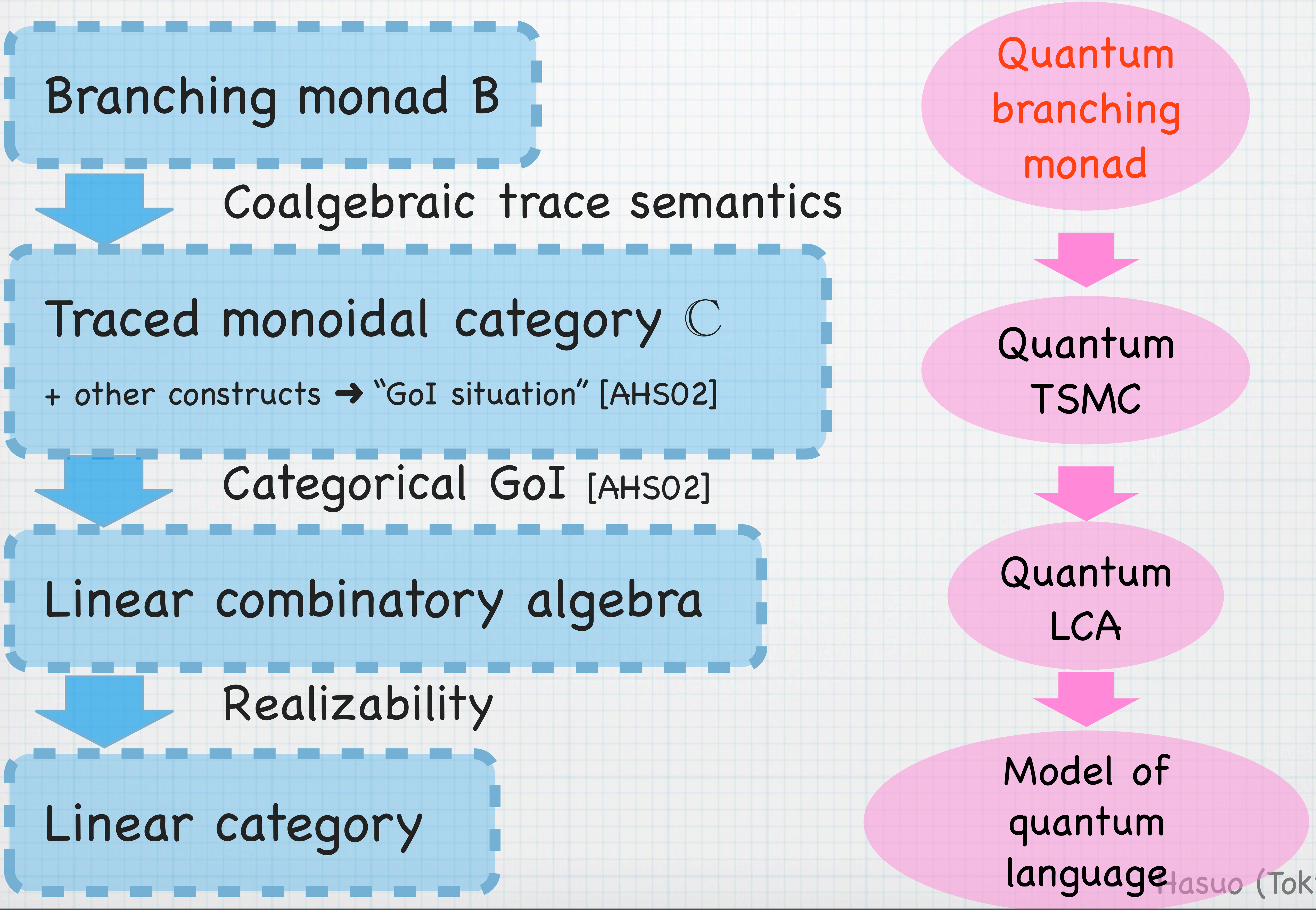
Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

The Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

Quantum branching monad

Quantum TSMC

Quantum LCA

Model of quantum language

The Quantum Branching Monad

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \mathbf{QO}_{m,n} \;\middle|\; \text{the trace condition} \right\}$$

The Quantum Branching M

$$\mathbf{QO}_{\boldsymbol{m},\boldsymbol{n}} := \left\{ \begin{array}{l} \text{quantum operations,} \\ \text{from dim. } \boldsymbol{m} \text{ to dim. } \boldsymbol{n} \end{array} \right\}$$

$$\boldsymbol{QY} = \left\{ \boldsymbol{c} : \boldsymbol{Y} \to \prod_{\boldsymbol{m},\boldsymbol{n}\in\mathbb{N}} \underline{\mathbf{QO}_{\boldsymbol{m},\boldsymbol{n}}} \;\middle|\; \text{the trace condition} \right\}$$

The Quantum Branching M

$$\mathbf{QO}_{m,n} := \left\{ \begin{array}{l} \text{quantum operations,} \\ \text{from dim. } m \text{ to dim. } n \end{array} \right\}$$

$$QY = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \underline{\mathbf{QO}_{m,n}} \;\middle|\; \underline{\text{the trace condition}} \right\}$$

$$\sum_{y\in Y} \sum_{n\in\mathbb{N}} \mathsf{tr}\left[\left(c(y)\right)_{m,n}(\rho)\right] \leq 1 \ ,$$

$$\forall m \in \mathbb{N},\ \forall \rho \in D_m .$$

The Quantum Branching M

$$\mathbf{QO}_{m,n} := \left\{ \begin{array}{l} \text{quantum operations,} \\ \text{from dim. } \boldsymbol{m} \text{ to dim. } \boldsymbol{n} \end{array} \right\}$$

$$\mathcal{Q}Y = \Big\{ c : Y \to \prod_{m,n\in\mathbb{N}} \underline{\mathbf{QO}_{m,n}} \;\Big|\; \underline{\text{the trace condition}} \Big\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\big[(c(y))_{m,n}(\rho)\big] \le 1\ , \qquad \forall m\in\mathbb{N},\ \forall\rho\in D_m.$$

- Compare with

$$\mathcal{P}Y = \Big\{ c : Y \to 2 \Big\}$$

$$\mathcal{D}Y = \Big\{ c : Y \to [0,1] \;\Big|\; \sum_{y\in Y} c(y) \le 1 \Big\}$$

The Quantum Branching M

$$\mathbf{QO}_{m,n} := \left\{ \begin{array}{l} \text{quantum operations,} \\ \text{from dim. } m \text{ to dim. } n \end{array} \right\}$$

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n \in \mathbb{N}} \underline{\mathbf{QO}_{m,n}} \;\middle|\; \underline{\text{the trace condition}} \right\}$$

$$\sum_{y \in Y} \sum_{n \in \mathbb{N}} \mathbf{tr}\big[\big(c(y)\big)_{m,n}(\rho)\big] \le 1 \;,$$

$$\forall m \in \mathbb{N},\; \forall \rho \in D_m .$$

- Compare with

$$\mathcal{P}Y = \left\{ c : Y \to 2 \right\}$$

$$\mathcal{D}Y = \left\{ c : Y \to [0,1] \;\middle|\; \sum_{y \in Y} c(y) \le 1 \right\}$$

The Quantum Branching M

$$\mathbf{QO}_{m,n} := \left\{ \begin{array}{l} \text{quantum operations,} \\ \text{from dim. } m \text{ to dim. } n \end{array} \right\}$$

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \underline{\mathbf{QO}_{m,n}} \;\middle|\; \underline{\text{the trace condition}} \right\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathrm{tr}\big[(c(y))_{m,n}(\rho)\big] \leq 1 \ , \qquad \forall m\in\mathbb{N},\ \forall \rho\in D_m .$$

- Compare with

$$\mathcal{P}Y = \left\{ c : Y \to 2 \right\}$$

$$\mathcal{D}Y = \left\{ c : Y \to [0,1] \;\middle|\; \underline{\sum_{y\in Y} c(y) \leq 1} \right\}$$

The Quantum Branching Monad

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \mathbf{QO}_{m,n} \;\middle|\; \underline{\text{the trace condition}} \right\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\big[(c(y))_{m,n}(\rho)\big] \le 1 \ ,$$

$$\forall m \in \mathbb{N},\ \forall \rho \in D_m .$$

$$\frac{X \xrightarrow{f} Y \text{ in } \mathcal{K}\ell(\mathcal{Q})}{\overline{X \xrightarrow{f} \mathcal{Q}Y \text{ in } \mathbf{Sets}}}$$

- Given $x \in X$, $y \in Y$, $m \in \mathbb{N}$, $n \in \mathbb{N}$ determines a quantum operation

$$\Big(f(x)(y) \Big)_{m,n} \;:\; D_m \to D_n$$

- Subject to the **trace condition**

Hasuo (Tokyo)

The Quantum Branching Monad

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \mathbf{QO}_{m,n} \;\middle|\; \text{the trace condition} \right\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\big[(c(y))_{m,n}(\rho)\big] \le 1 \;,\quad \forall m\in\mathbb{N},\ \forall\rho\in D_m .$$

$$\frac{X \xrightarrow{f} Y \ \text{ in } \mathcal{K}\ell(\mathcal{Q})}{X \xrightarrow{f} \mathcal{Q}Y \ \text{ in } \mathbf{Sets}}$$

- Given $x \in X$, $y \in Y$, $m \in \mathbb{N}$, $n \in \mathbb{N}$ determines a quantum operation

$$\Big(f(x)(y) \Big)_{m,n} \;:\; D_m \to D_n$$

- Subject to the **trace condition**

Any opr. on quantum data:

combination of

- ● preparation
- ● unitary transf.
- ● measurement

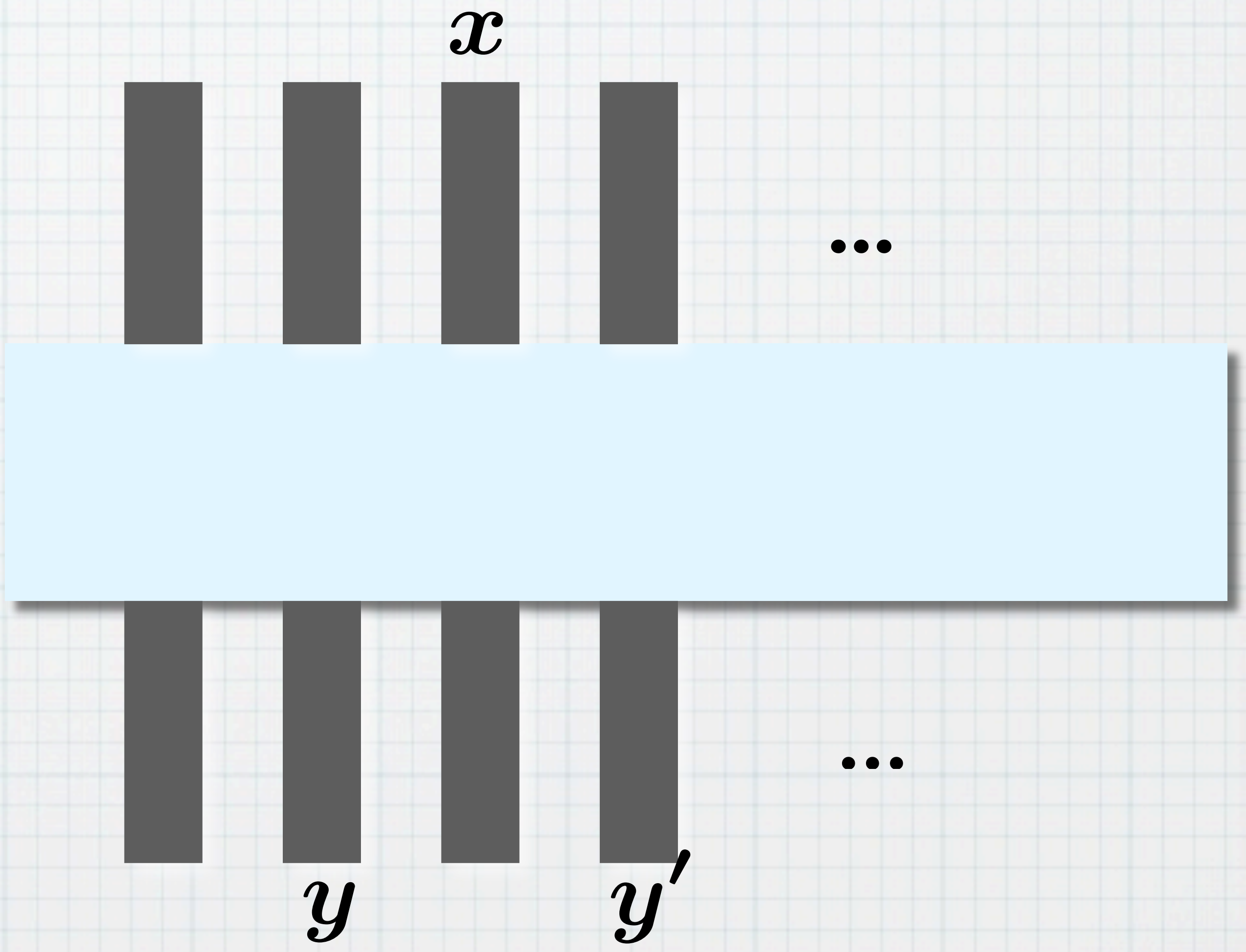
The Quantum Branching Monad

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \mathbf{QO}_{m,n} \;\middle|\; \text{the trace condition} \right\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\Big[\big(c(y)\big)_{m,n}(\rho)\Big] \le 1 \;, \qquad \forall m\in\mathbb{N},\ \forall \rho \in D_m .$$

$$\begin{array}{c} X \xrightarrow{f} Y \ \text{ in } \mathcal{K}\ell(\mathcal{Q}) \\ \hline\hline X \xrightarrow{f} \mathcal{Q}Y \ \text{ in } \mathbf{Sets} \end{array}$$

- Given $x\in X,\ y\in Y,\ m\in\mathbb{N},\ n\in\mathbb{N}$ determines a quantum operation $\Big(f(x)(y)\Big)_{m,n}$
- **trace cond.:**



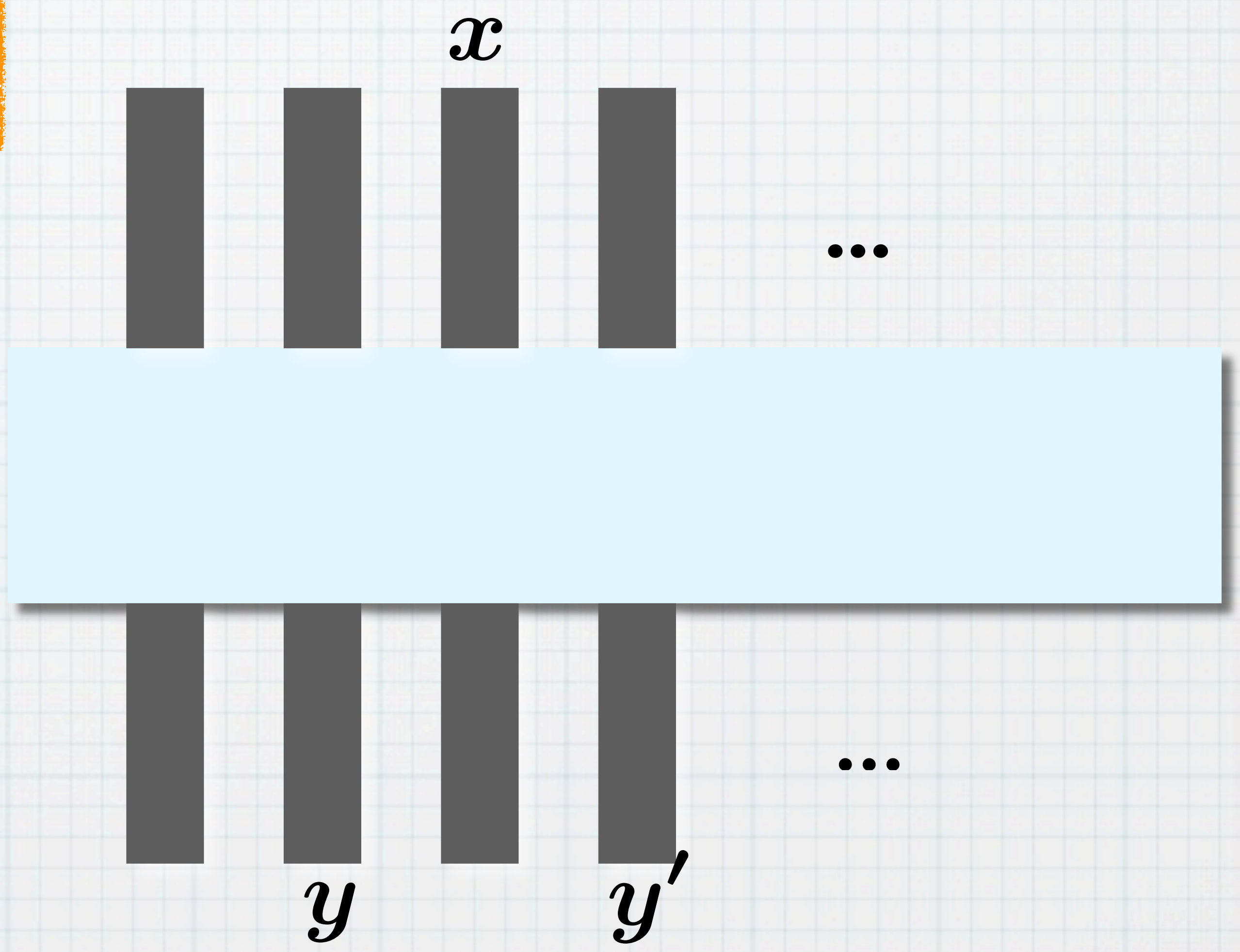
The Quantum Branching Monad

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \mathbf{QO}_{m,n} \;\middle|\; \underline{\text{the trace condition}} \right\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\Big[\big(c(y)\big)_{m,n}(\rho)\Big] \le 1 \ , \qquad \forall m\in\mathbb{N},\ \forall\rho\in D_m .$$

$$\begin{array}{c} X \xrightarrow{f} Y \ \text{ in } \mathcal{K}\ell(\mathcal{Q}) \\ \hline\hline X \xrightarrow{f} \mathcal{Q}Y \ \text{ in } \mathbf{Sets} \end{array}$$

- Given $\overset{\text{entrance}}{x \in X}$, $\overset{\text{exit}}{y \in Y}$, $\overset{\text{in dim.}}{m \in \mathbb{N}}$, $\overset{\text{out dim.}}{n \in \mathbb{N}}$ determines a quantum operation $\Big(f(x)(y)\Big)_{m,n}$
- **trace cond.:**



The Quantum Branching Monad

$$\mathcal{Q}Y = \Big\{ c : Y \to \prod_{m,n\in\mathbb{N}} \mathbf{QO}_{m,n} \;\Big|\; \text{the trace condition} \Big\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\Big[\big(c(y)\big)_{m,n}(\rho)\Big] \le 1 \;,$$
$$\forall m \in \mathbb{N},\ \forall \rho \in D_m .$$

$$\frac{X \xrightarrow{f} Y \ \text{ in } \mathcal{K}\ell(\mathcal{Q})}{X \xrightarrow{f} \mathcal{Q}Y \ \text{ in Sets}}$$

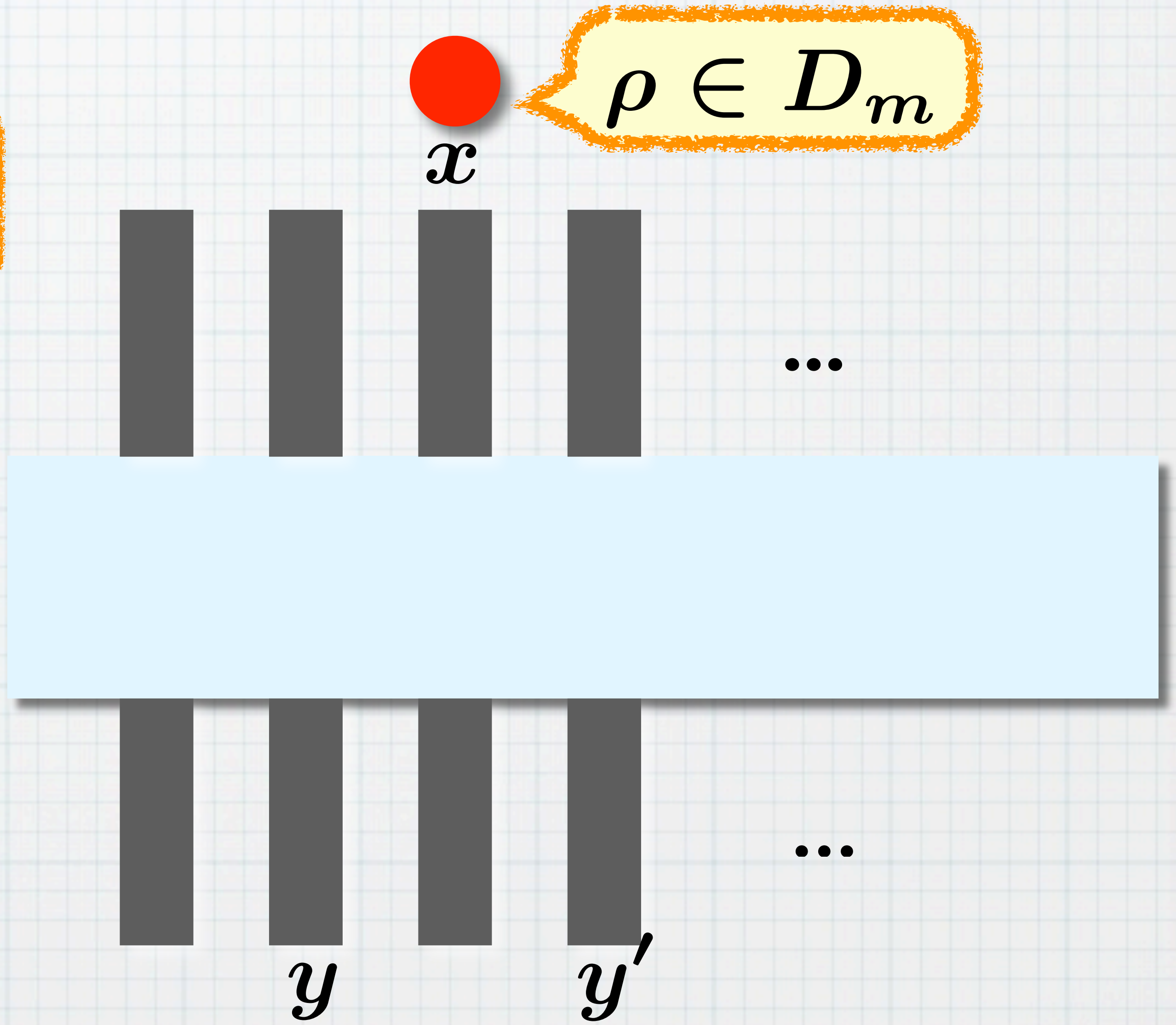
entrance

exit

in dim.

out dim.

- * Given $x \in X$, $y \in Y$, $m \in \mathbb{N}$, $n \in \mathbb{N}$ determines a quantum operation $\Big(f(x)(y)\Big)_{m,n}$
- * **trace cond.:**


$\rho \in D_m$

The Quantum Branching Monad

$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n\in\mathbb{N}} \mathbf{QO}_{m,n} \;\middle|\; \text{the trace condition} \right\}$$

$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\big[\big(c(y)\big)_{m,n}(\rho)\big] \le 1 \;,\quad \forall m\in\mathbb{N},\ \forall\rho\in D_m.$$

$$\frac{X \xrightarrow{f} Y \ \text{ in } \mathcal{K}\ell(\mathcal{Q})}{X \xrightarrow{f} \mathcal{Q}Y \ \text{ in Sets}}$$

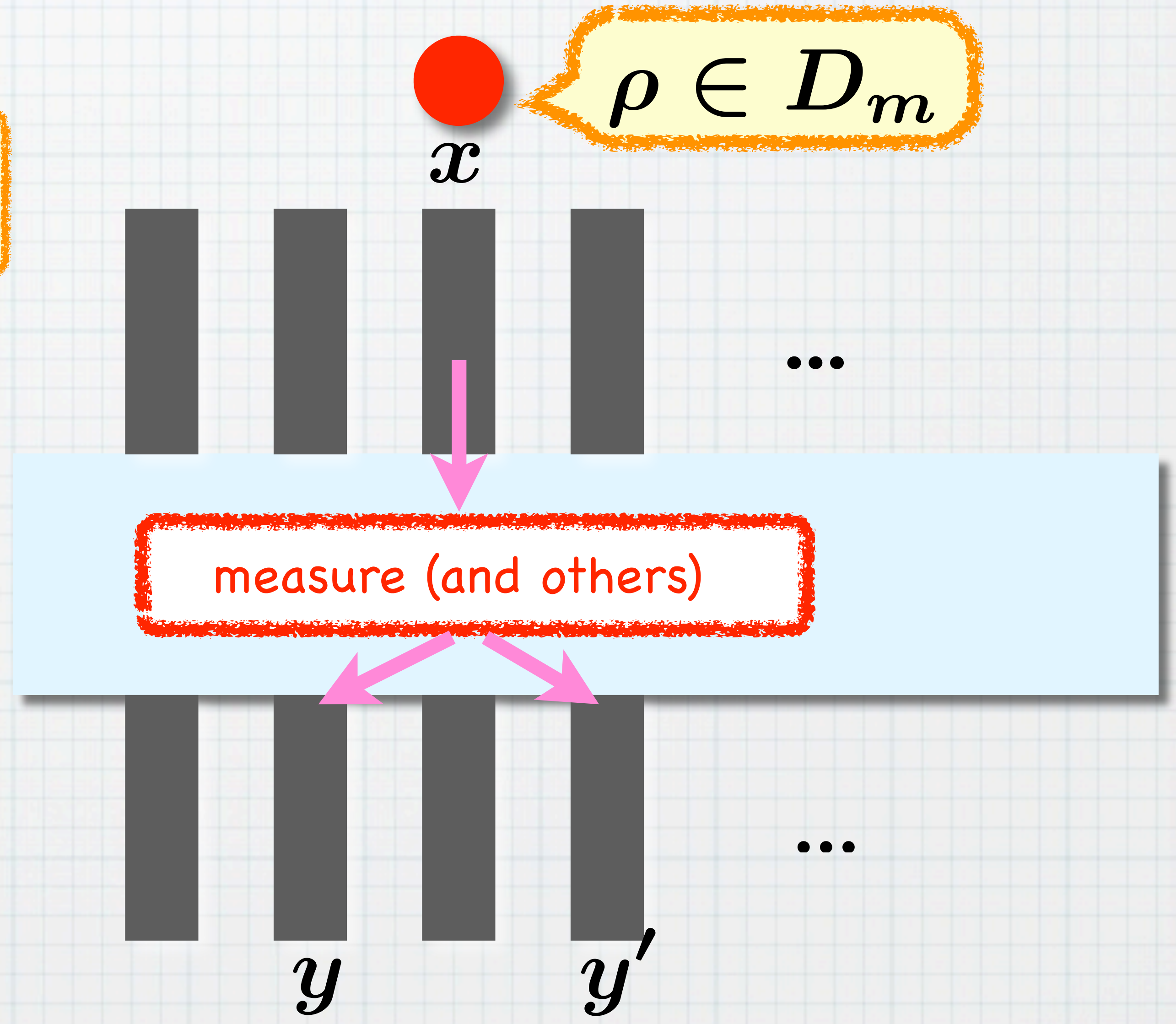
entrance

exit

in dim.

out dim.

- Given $x\in X,\ y\in Y,\ m\in\mathbb{N},\ n\in\mathbb{N}$ determines a quantum operation $\big(f(x)(y)\big)_{m,n}$
- trace cond.:



The Quantum Branching Monad

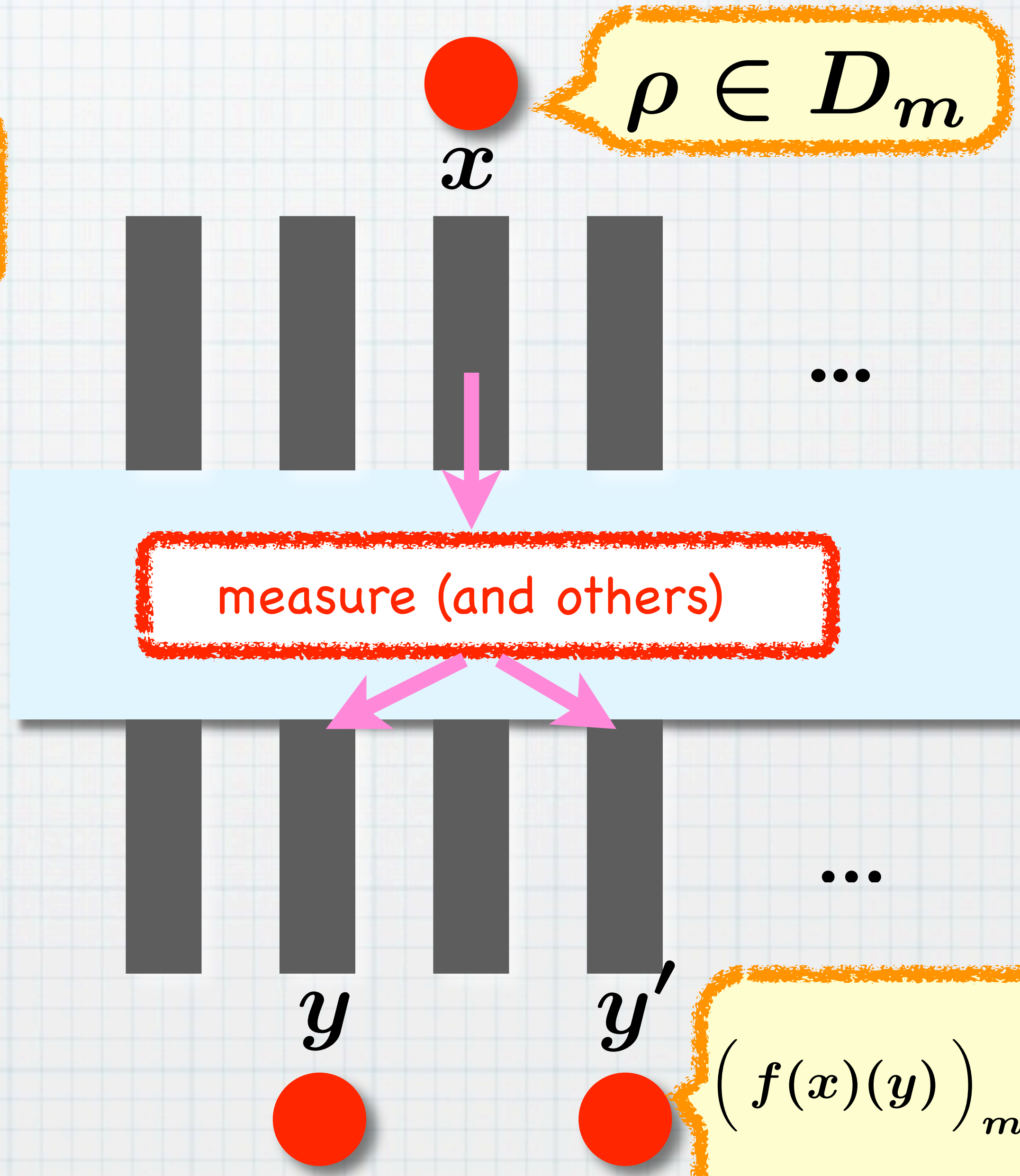
$$\mathcal{Q}Y = \left\{ c : Y \to \prod_{m,n \in \mathbb{N}} \mathbf{QO}_{m,n} \;\middle|\; \text{the trace condition} \right\}$$

$$\sum_{y \in Y} \sum_{n \in \mathbb{N}} \mathbf{tr}\big[\big(c(y)\big)_{m,n}(\rho) \big] \leq 1 \ , \quad \forall m \in \mathbb{N},\ \forall \rho \in D_m .$$

$$\begin{array}{c} X \xrightarrow{f} Y \ \text{ in } \mathcal{K}\ell(\mathcal{Q}) \\ \hline\hline X \xrightarrow{f} \mathcal{Q}Y \ \text{ in } \mathbf{Sets} \end{array}$$

entrance exit in dim. out dim.

- Given $x \in X,\ y \in Y,\ m \in \mathbb{N},\ n \in \mathbb{N}$ determines a quantum operation $\Big(f(x)(y) \Big)_{m,n}$
- **trace cond.:**



$\rho \in D_m$

$\Big(f(x)(y) \Big)_{m,n}(\rho) \in D_n$

for each n

The Quantum Branching Monad

$$\frac{X \xrightarrow{f} Y \ \text{ in } \mathcal{K}\ell(\mathcal{Q})}{X \xrightarrow{f} \mathcal{Q}Y \ \text{ in Sets}}$$

$$\mathcal{Q}Y = \Big\{ c : Y \to \prod_{m,n \in \mathbb{N}} \mathbf{QO}_{m,n} \ \Big| \ \text{the trace condition} \Big\}$$

$$\sum_{y \in Y} \sum_{n \in \mathbb{N}} \mathbf{tr}\big[(c(y))_{m,n}(\rho)\big] \leq 1 \ , \qquad \forall m \in \mathbb{N}, \ \forall \rho \in D_m .$$

entrance

exit

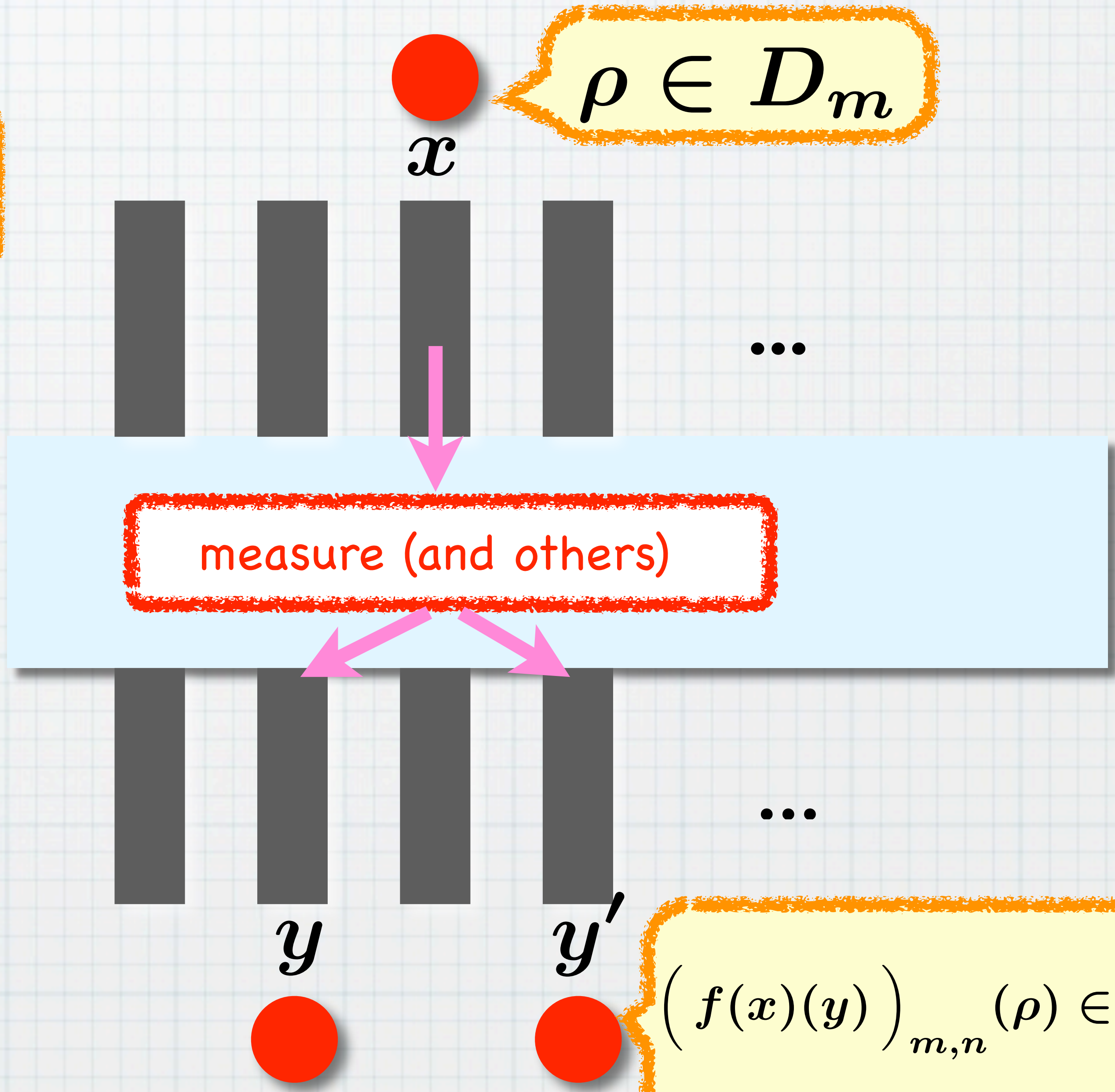
in dim.

out dim.

- Given $x \in X$, $y \in Y$, $m \in \mathbb{N}$, $n \in \mathbb{N}$ determines a quantum operation $\big(f(x)(y)\big)_{m,n}$
- **trace cond.:**

$$\sum_{y,n} \Pr\left(\begin{array}{c}\text{Token led} \\ \text{to } y \\ \text{with dim. } n\end{array} \right) \leq 1$$

$\rho \in D_m$

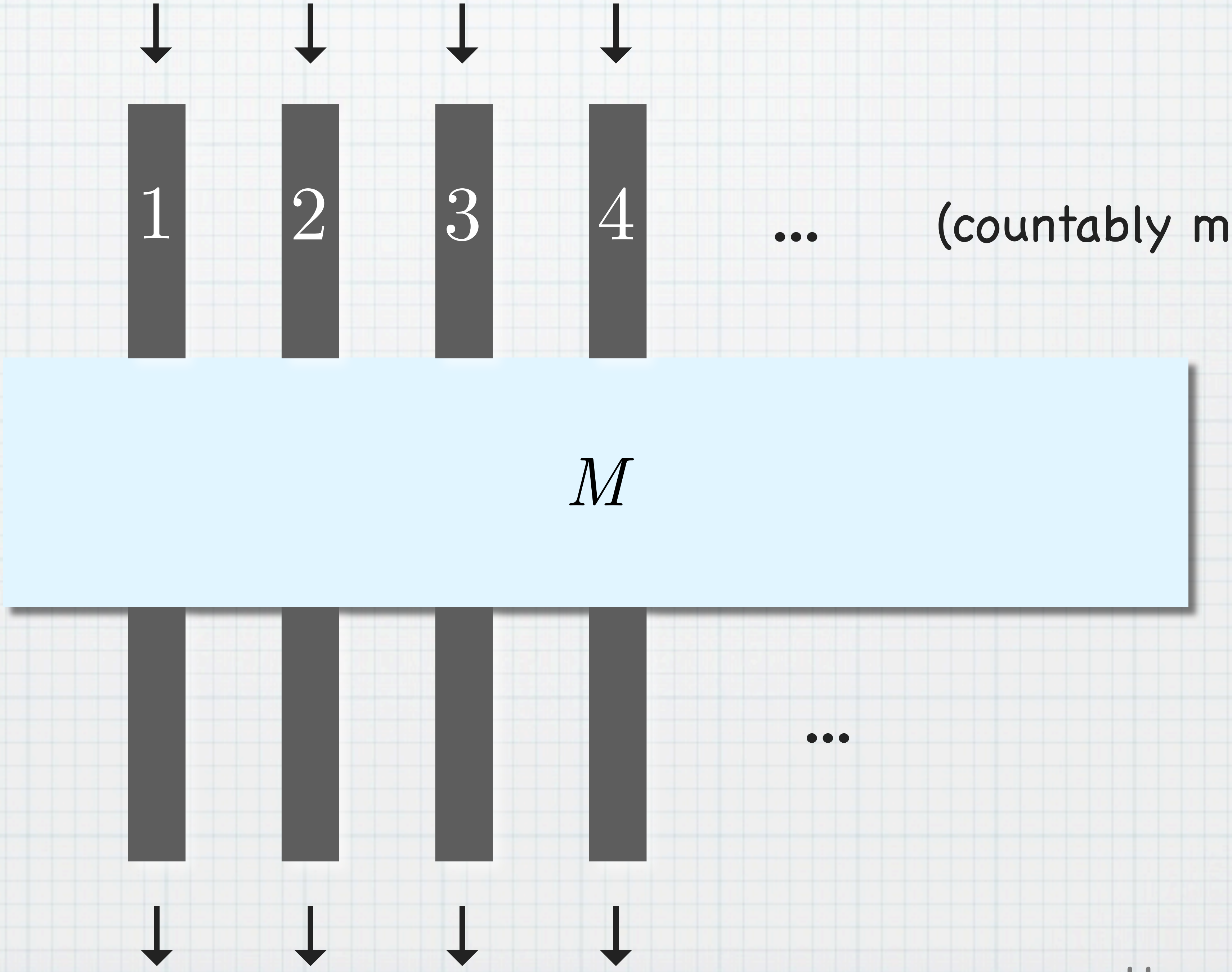


$\big(f(x)(y)\big)_{m,n}(\rho) \in D_n$

for each n

Quantum
Geometry of Interaction

$[\![M]\!] =$

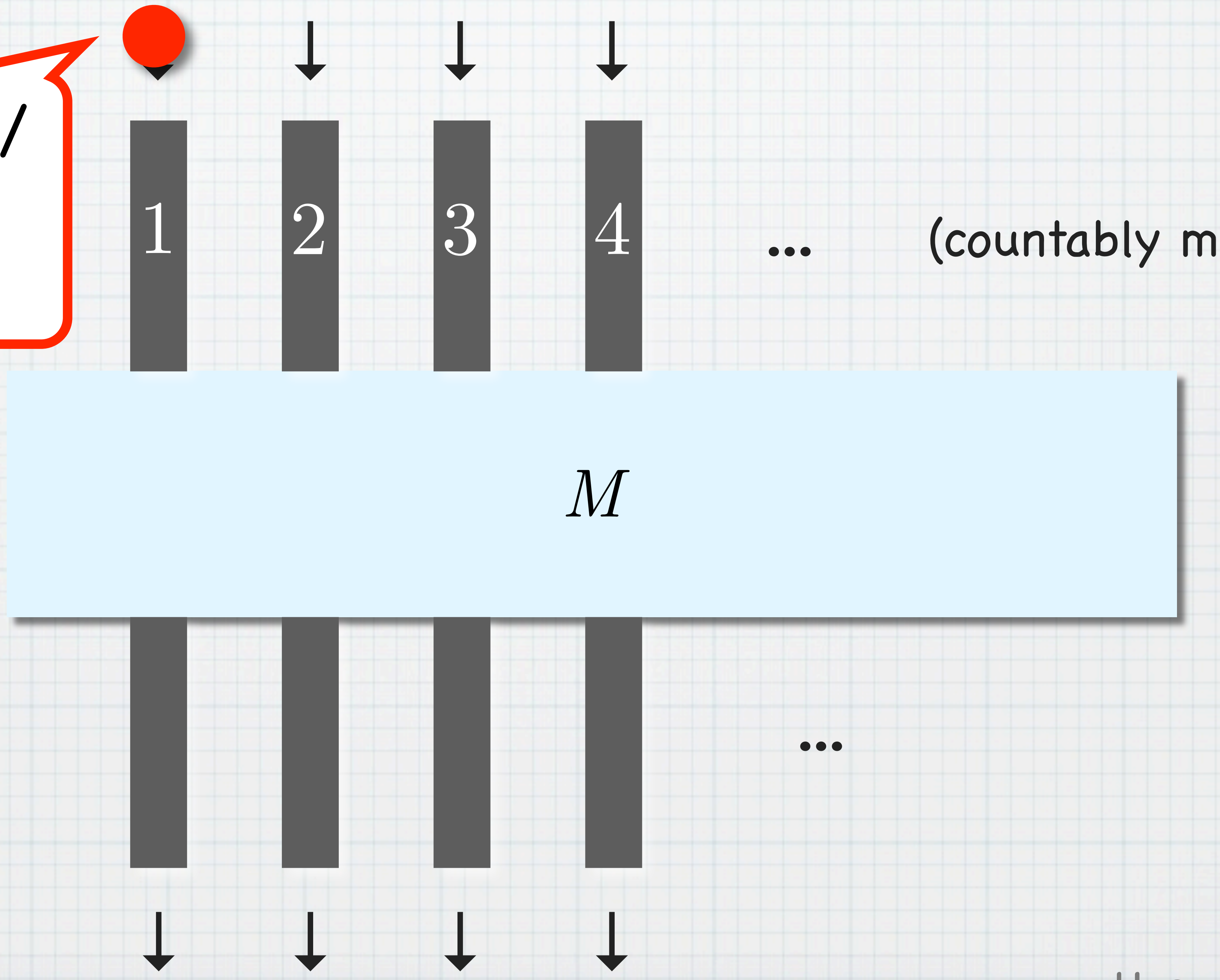


(countably many)

Quantum
Geometry of Interaction

Not just a token/ particle, but quantum state!

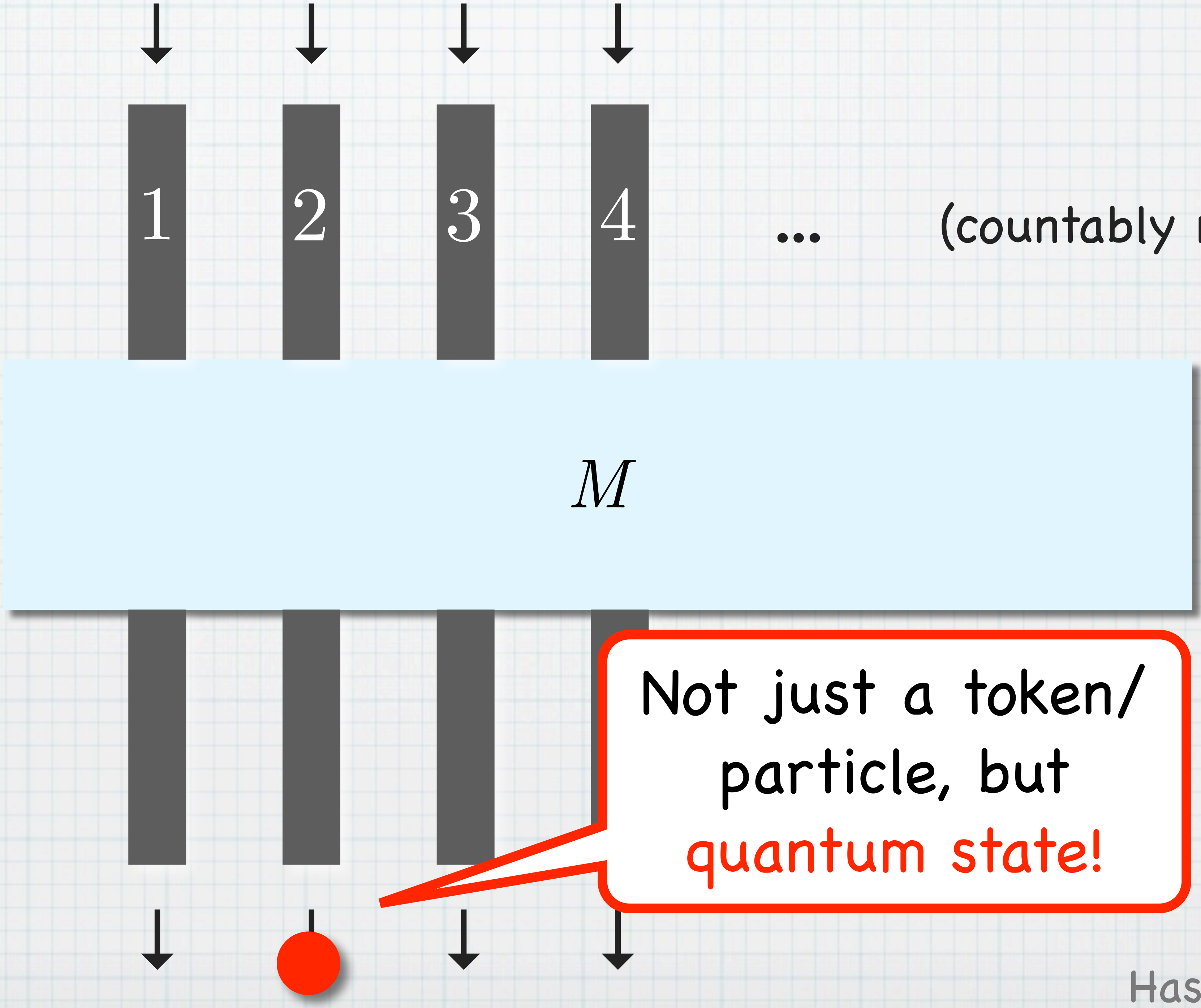
$$[\![M]\!] =$$



(countably many)

Quantum
Geometry of Interaction

$$[\![M]\!] =$$



(countably many)

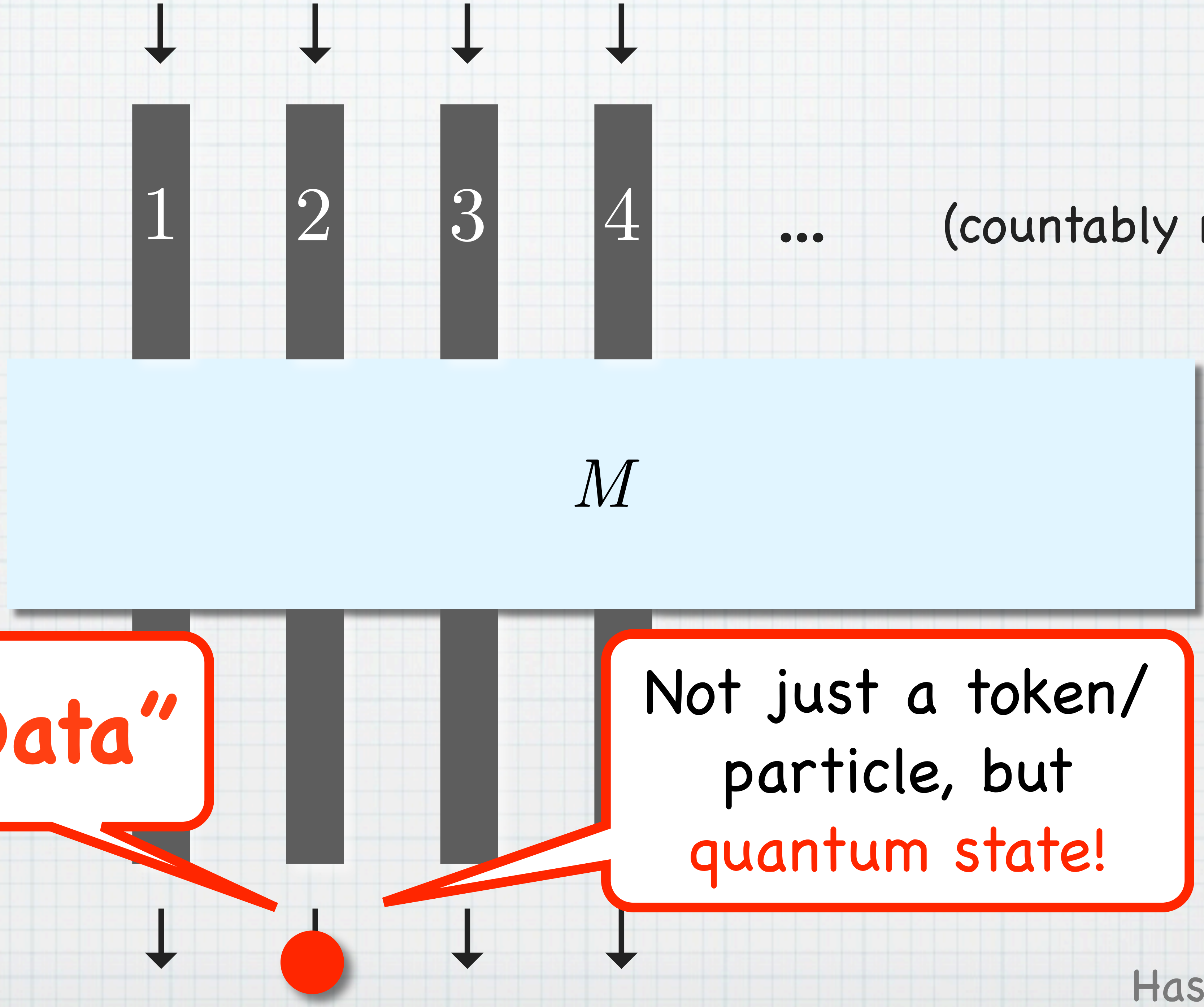
Not just a token/ particle, but quantum state!

Hasuo (Tokyo)

Quantum

Geometry of Interaction

$[\![M]\!] =$



(countably many)

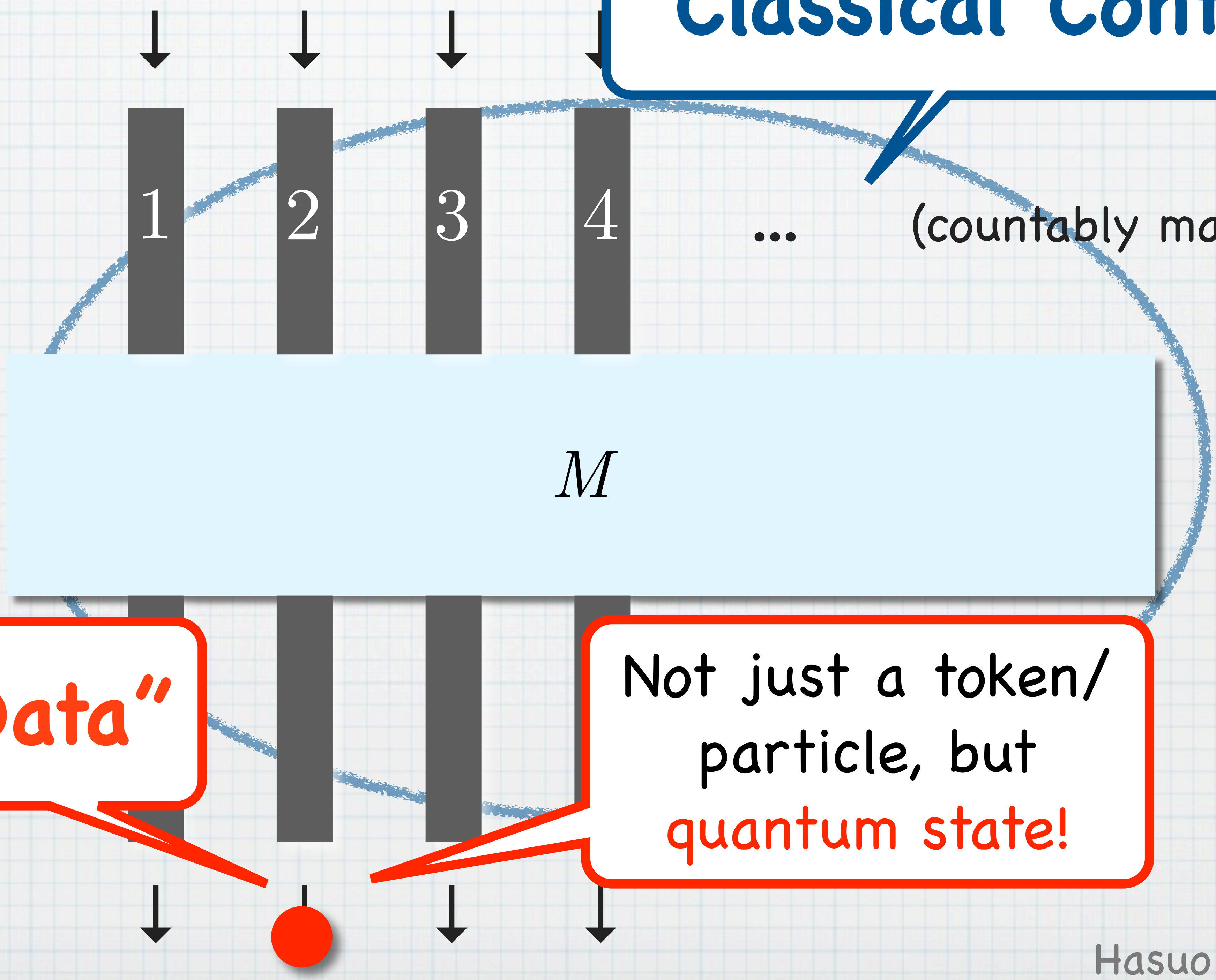
"Quantum Data"

Not just a token/ particle, but quantum state!

Hasuo (Tokyo)

Quantum
Geometry of Interaction

$$[\![M]\!] =$$



(countably many)

“Classical Control”

“Quantum Data”

Not just a token/ particle, but quantum state!

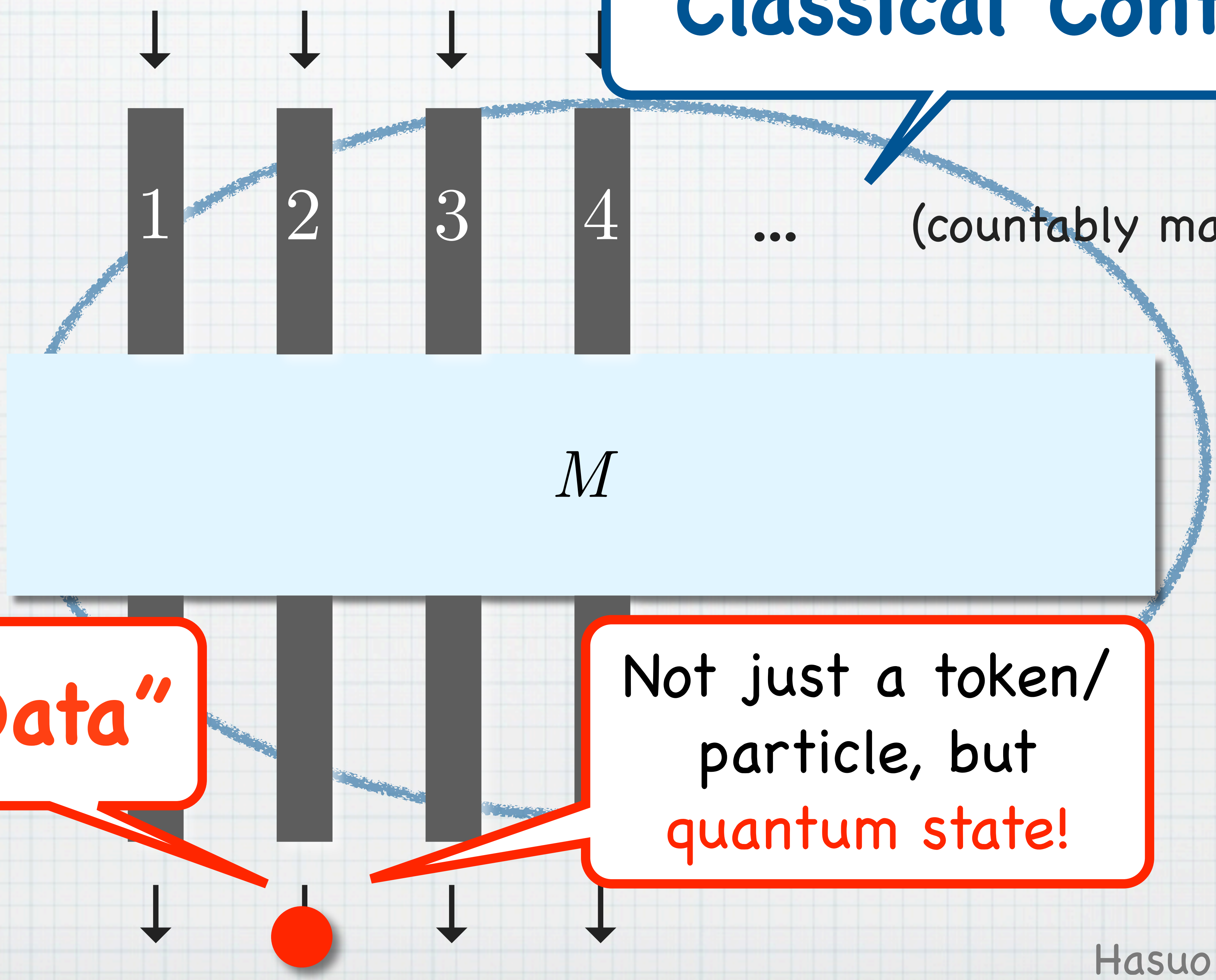
Hasuo (Tokyo)

Quan
Geometry o

- "in which pipe"
- (measurement → case-distinction) leads a token to different pipes

"Classical Control"

$[\![M]\!] =$



(countably many)

"Quantum Data"

Not just a token/ particle, but quantum state!

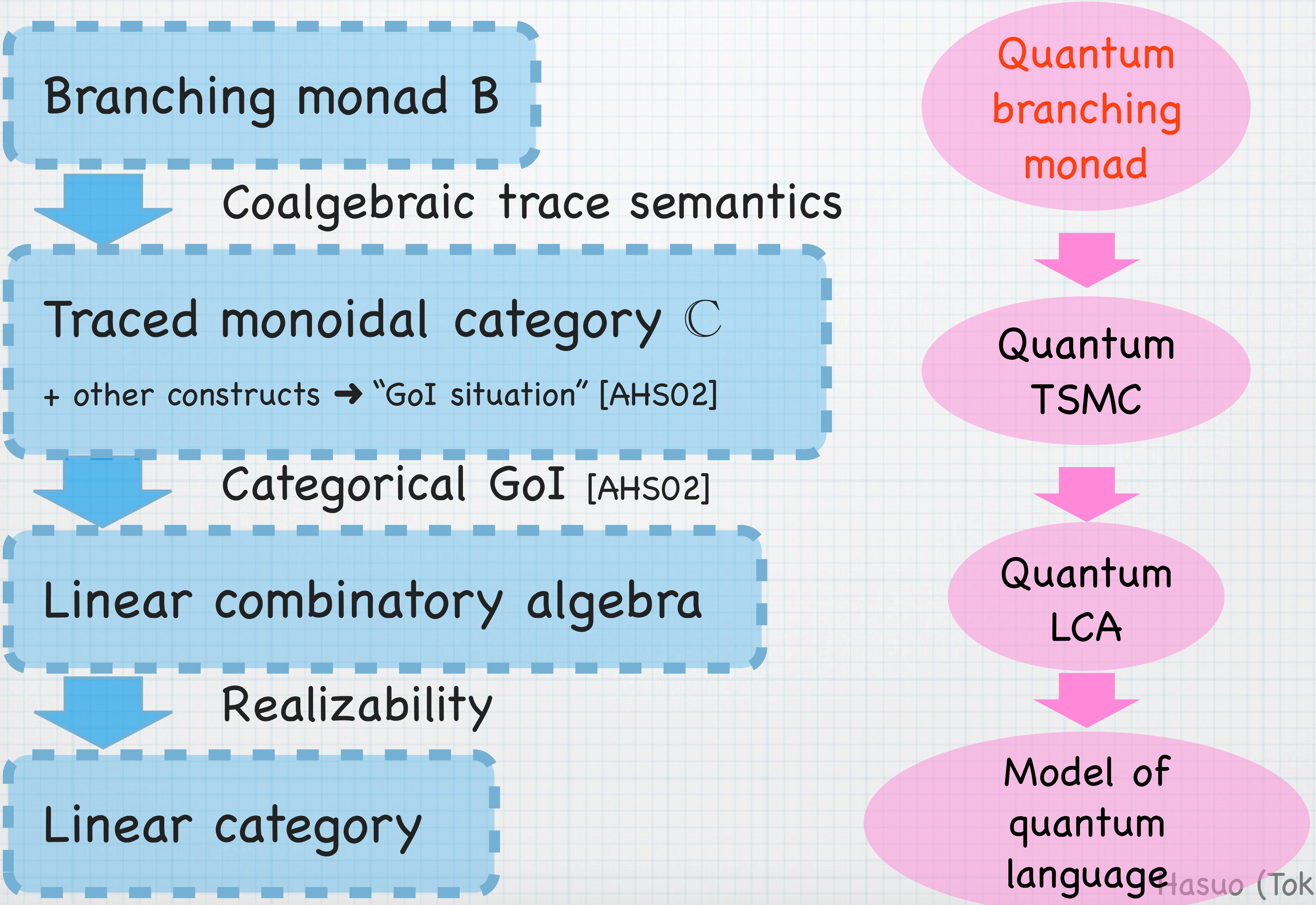
Indeed...

- The monad $\mathcal{Q}$ qualifies as a "branching monad"
- The quantum GoI workflow leads to a linear category $\mathbf{PER}_{\mathcal{Q}}$
- From which we construct an adequate denotational model

End of the Story?

- No! All the technicalities are yet to come:
  - CPS-style interpretation (for partial measurement)
  - Result type: a final coalgebra in $\mathbf{PER}_Q$
  - **Admissible PERs** for recursion
  - ...
- On the next occasion :-)

Conclusion: the Categorical GoI Workflow



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ “GoI situation” [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

Quantum branching monad

Quantum TSMC

Quantum LCA

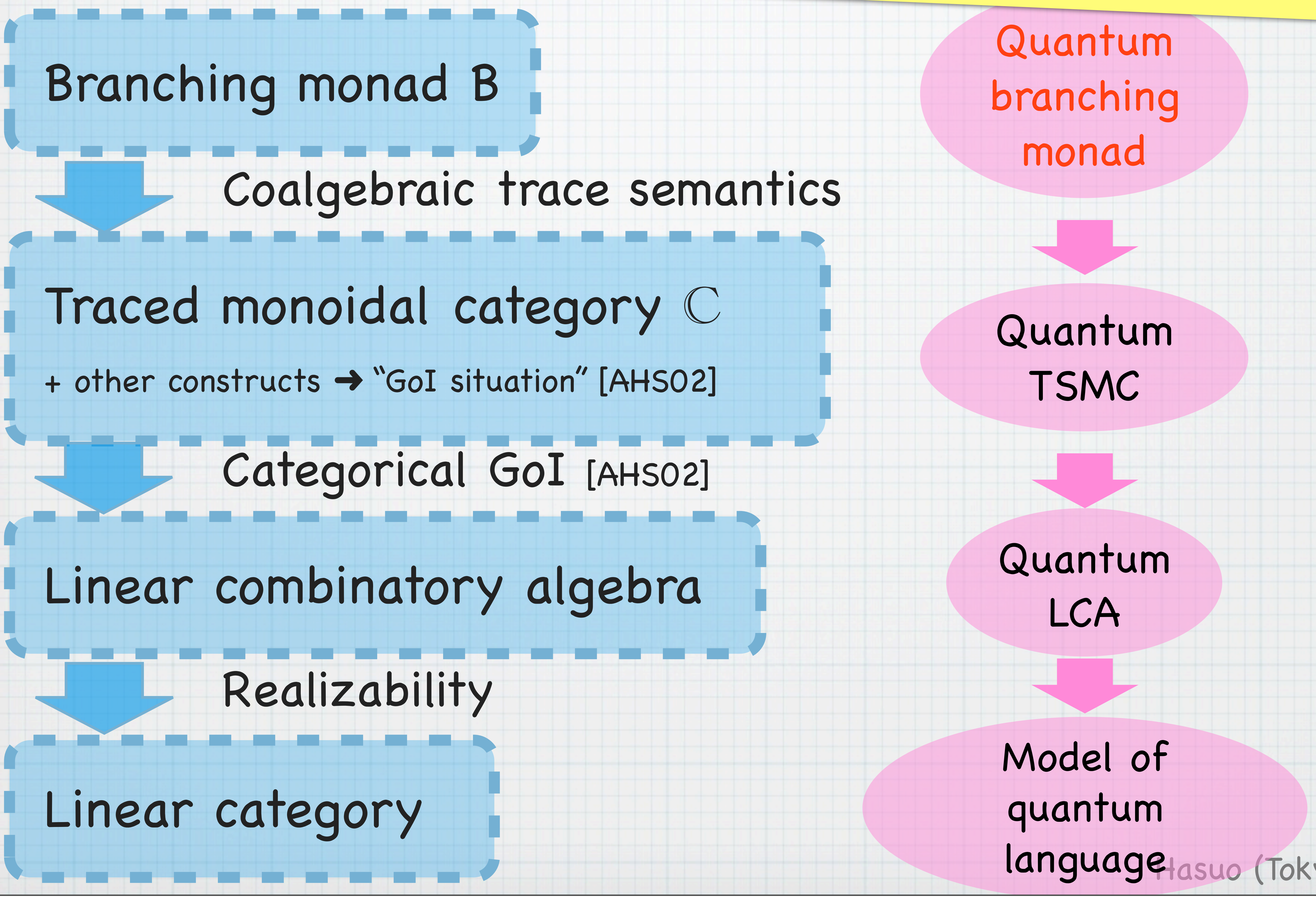
Model of quantum language

Thank you for your attention!

Ichiro Hasuo (Dept. CS, U Tokyo)

http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro/

Conclusion: the Cat



Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ “GoI situation” [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

Realizability

Linear category

Quantum branching monad

Quantum TSMC

Quantum LCA

Model of quantum language

The Language $q\lambda_\ell$

- Roughly: linear λ + quantum primitives
- "Quantum data, classical control"
  - No superposed threads
- Based on [Selinger&Valiron'09]
  - With slight modifications
  - Notably: quantum $\otimes$ vs. linear logic ☒
    - The same in [Selinger&Valiron'09]
      ➜ clean type system, aids programming
    - But... problem with GoI-style semantics

The Language $q\lambda_\ell$

The *types* of $\mathbf{q}\boldsymbol{\lambda}_\ell$ are:

$$\boldsymbol{A}, \boldsymbol{B} ::= \boldsymbol{n}\text{-}\mathtt{qbit} \mid \boldsymbol{!}\,\boldsymbol{A} \mid \boldsymbol{A} \multimap \boldsymbol{B} \mid \top \mid \boldsymbol{A} \boxtimes \boldsymbol{B} \mid \boldsymbol{A} + \boldsymbol{B} \;,$$

with conventions $\mathtt{qbit} := \mathbf{1}\text{-}\mathtt{qbit}$ and $\mathtt{bit} := \top + \top$.

The *terms* of $\mathbf{q}\boldsymbol{\lambda}_\ell$ are:

$$\begin{aligned}
&\boldsymbol{M}, \boldsymbol{N}, \boldsymbol{P} ::= \\
&\quad \boldsymbol{x} \mid \boldsymbol{\lambda x^A . M} \mid \boldsymbol{M\,N} \mid \langle \boldsymbol{M}, \boldsymbol{N} \rangle \mid * \mid \\
&\quad \mathtt{let}\,\langle \boldsymbol{x^A}, \boldsymbol{y^B} \rangle = \boldsymbol{M}\;\mathtt{in}\;\boldsymbol{N} \mid \mathtt{let}\,* = \boldsymbol{M}\;\mathtt{in}\;\boldsymbol{N} \mid \\
&\quad \mathtt{inj}_\ell^{\boldsymbol{B}}\;\boldsymbol{M} \mid \mathtt{inj}_r^{\boldsymbol{A}}\;\boldsymbol{M} \mid \\
&\quad \mathtt{match}\;\boldsymbol{P}\;\mathtt{with}\;(\boldsymbol{x^A} \mapsto \boldsymbol{M} \mid \boldsymbol{y^B} \mapsto \boldsymbol{N}) \mid \\
&\quad \mathtt{letrec}\;\boldsymbol{f^A x} = \boldsymbol{M}\;\mathtt{in}\;\boldsymbol{N} \mid \\
&\quad \mathtt{new}\,|\mathbf{0}\rangle \mid \mathtt{meas}_i^{n+1} \mid \boldsymbol{U} \mid \mathtt{cmp}_{\boldsymbol{m},\boldsymbol{n}} \;,
\end{aligned}$$

with conventions $\mathtt{tt} := \mathtt{inj}_\ell^{\top}(*)$ and $\mathtt{ff} := \mathtt{inj}_r^{\top}(*)$.

The Langua

Different from quantum ⊗
(Unlike [Selinger-Valiron'09]);
same as the one in **PER**

The *types* of $\mathbf{q}\boldsymbol{\lambda}_\ell$ are:

$$\boldsymbol{A}, \boldsymbol{B} ::= \boldsymbol{n}\text{-}\mathtt{qbit} \mid \boldsymbol{!}\,\boldsymbol{A} \mid \boldsymbol{A} \multimap \boldsymbol{B} \mid \top \mid \underline{\boldsymbol{A} \boxtimes \boldsymbol{B}} \mid \boldsymbol{A} + \boldsymbol{B} \ ,$$

with conventions $\mathtt{qbit} := \mathbf{1}\text{-}\mathtt{qbit}$ and $\mathtt{bit} := \top + \top$.

The *terms* of $\mathbf{q}\boldsymbol{\lambda}_\ell$ are:

$$\begin{aligned}
&\boldsymbol{M}, \boldsymbol{N}, \boldsymbol{P} ::= \\
&\quad \boldsymbol{x} \mid \boldsymbol{\lambda x^A . M} \mid \boldsymbol{M\,N} \mid \langle \boldsymbol{M}, \boldsymbol{N} \rangle \mid * \mid \\
&\quad \mathtt{let}\ \langle \boldsymbol{x^A}, \boldsymbol{y^B} \rangle = \boldsymbol{M}\ \mathtt{in}\ \boldsymbol{N} \mid \mathtt{let}\ * = \boldsymbol{M}\ \mathtt{in}\ \boldsymbol{N} \mid \\
&\quad \mathtt{inj}_\ell^{\boldsymbol{B}}\ \boldsymbol{M} \mid \mathtt{inj}_r^{\boldsymbol{A}}\ \boldsymbol{M} \mid \\
&\quad \mathtt{match}\ \boldsymbol{P}\ \mathtt{with}\ (\boldsymbol{x^A} \mapsto \boldsymbol{M} \mid \boldsymbol{y^B} \mapsto \boldsymbol{N}) \mid \\
&\quad \mathtt{letrec}\ \boldsymbol{f^A x} = \boldsymbol{M}\ \mathtt{in}\ \boldsymbol{N} \mid \\
&\quad \mathtt{new}\ |\mathbf{0}\rangle \mid \mathtt{meas}_i^{n+1} \mid \boldsymbol{U} \mid \mathtt{cmp}_{\boldsymbol{m},\boldsymbol{n}} \ ,
\end{aligned}$$

with conventions $\mathtt{tt} := \mathtt{inj}_\ell^{\top}(*)$ and $\mathtt{ff} := \mathtt{inj}_r^{\top}(*)$.

The Langua

Different from quantum ⊗ (Unlike [Selinger-Valiron'09]); same as the one in **PER**

2-qbit ≅ qbit ⊗ qbit

$$A, B ::= n\text{-}\mathtt{qbit} \mid\, !A \mid A \multimap B \mid \top \mid A \boxtimes B \mid A + B\ ,$$

with conventions $\mathtt{qbit} := \mathbf{1}\text{-}\mathtt{qbit}$ and $\mathtt{bit} := \top + \top$.

The *terms* of $\mathbf{q\lambda}_\ell$ are:

$$\begin{aligned}
&M, N, P ::= \\
&\quad x \mid \lambda x^A.M \mid MN \mid \langle M, N\rangle \mid * \mid \\
&\quad \mathtt{let}\ \langle x^A, y^B\rangle = M\ \mathtt{in}\ N \mid \mathtt{let}\ * = M\ \mathtt{in}\ N \mid \\
&\quad \mathtt{inj}_\ell^B\ M \mid \mathtt{inj}_r^A\ M \mid \\
&\quad \mathtt{match}\ P\ \mathtt{with}\ (x^A \mapsto M \mid y^B \mapsto N) \mid \\
&\quad \mathtt{letrec}\ f^A x = M\ \mathtt{in}\ N \mid \\
&\quad \mathtt{new}\ |\mathbf{0}\rangle \mid \mathtt{meas}_i^{n+1} \mid U \mid \mathtt{cmp}_{m,n}\ ,
\end{aligned}$$

with conventions $\mathtt{tt} := \mathtt{inj}_\ell^\top(*)$ and $\mathtt{ff} := \mathtt{inj}_r^\top(*)$.

Hasuo (Tokyo)

Monday, November 7, 2011

The Langua

Different from quantum ⊗ (Unlike [Selinger-Valiron'09]); same as the one in **PER**

2-qbit ≅ qbit ⊗ qbit

$$A, B ::= \underline{n\text{-}\mathtt{qbit}} \mid\, !A \mid A \multimap B \mid \top \mid \underline{A \boxtimes B} \mid A + B \ ,$$

with conventions $\mathtt{qbit} := \mathbf{1}\text{-}\mathtt{qbit}$ and $\mathtt{bit} := \top + \top$.

The *terms* of $\mathbf{q}\boldsymbol{\lambda}_\ell$ are:

$$\begin{aligned}
&M, N, P ::= \\
&\quad x \mid \lambda x^A.M \mid M\,N \mid \langle M, N\rangle \mid * \mid \\
&\quad \mathtt{let}\ \langle x^A, y^B\rangle = M\ \mathtt{in}\ N \mid \mathtt{let}\ * = M\ \mathtt{in}\ N \mid \\
&\quad \mathtt{inj}_\ell^B\ M \mid \mathtt{inj}_r^A\ M \mid \\
&\quad \mathtt{match}\ P\ \mathtt{with}\ (x^A \mapsto M \mid y^B \mapsto N) \mid \\
&\quad \underline{\mathtt{letrec}}\ f^A x = M\ \mathtt{in}\ N \mid \\
&\quad \mathtt{new}\ |\mathbf{0}\rangle \mid \mathtt{meas}_i^{n+1} \mid U \mid \mathtt{cmp}_{m,n} \ ,
\end{aligned}$$

Recursion

with conventions $\mathtt{tt} := \mathtt{inj}_\ell^\top(*)$ and $\mathtt{ff} := \mathtt{inj}_r^\top(*)$.

The Langua

Different from quantum ⊗
(Unlike [Selinger-Valiron'09]);
same as the one in **PER**

2-qbit ≅ qbit ⊗ qbit

$$A, B ::= n\text{-}\mathtt{qbit} \mid {!A} \mid A \multimap B \mid \top \mid A \boxtimes B \mid A + B\ ,$$

with conventions $\mathtt{qbit} := \mathbf{1}\text{-}\mathtt{qbit}$ and $\mathtt{bit} := \top + \top$.

The *terms* of $\mathbf{q}\boldsymbol{\lambda}_\ell$ are:

$$\begin{aligned} M, N, P ::= \\ & x \mid \lambda x^A.M \mid MN \mid \langle M, N\rangle \mid * \mid \\ & \mathtt{let}\ \langle x^A, y^B\rangle = M\ \mathtt{in}\ N \mid \mathtt{let} * = M\ \mathtt{in}\ N \mid \\ & \mathtt{inj}_\ell^B\ M \mid \mathtt{inj}_r^A\ M \mid \\ & \mathtt{match}\ P\ \mathtt{with}\ (x^A \mapsto M \mid y^B \mapsto N) \mid \\ & \mathtt{letrec}\ f^A x = M\ \mathtt{in}\ N \mid \\ & \mathtt{new}\ |\mathbf{0}\rangle \mid \mathtt{meas}_i^{n+1} \mid U \mid \mathtt{cmp}_{m,n}\ , \end{aligned}$$

Recursion

Quantum primitives

with conventions $\mathtt{tt} := \mathtt{inj}_\ell^\top(*)$ and $\mathtt{ff} := \mathtt{inj}_r^\top(*)$.

Hasuo (Tokyo)

Monday, November 7, 2011

Implicit linearity tracking via subtyping <:

e.g. !A <: A, !A <: !!A

(following [Selinger-Valiron'09])

$$\frac{n = 0 \Rightarrow m = 0 \ (*)}{!^n\, k\text{-}\mathtt{qbit} <: !^m\, k\text{-}\mathtt{qbit}}\ (k\text{-}\mathtt{qbit}) \qquad \frac{n = 0 \Rightarrow m = 0}{!^n \top <: !^m \top}\ (\top)$$

$$\frac{A_1 <: B_1 \quad A_2 <: B_2 \quad (*)}{!^n(A_1 \boxdot A_2) <: !^m(B_1 \boxdot B_2)}\ (\boxdot) \text{ with } \boxdot \in \{\boxtimes, +\}$$

$$\frac{B_1 <: A_1 \quad A_2 <: B_2 \quad (*)}{!^n(A_1 \multimap A_2) <: !^m(B_1 \multimap B_2)}\ (\multimap)$$

Measurements

$$\begin{array}{rcl}
A_{\mathrm{new}|0\rangle} & := & \mathtt{qbit} \\
A_{\mathrm{meas}_i^{n+1}} & := & (n+1)\text{-}\mathtt{qbit} \multimap (\mathtt{bit} \boxtimes n\text{-}\mathtt{qbit}) \text{ for } n \geq 1 \\
A_{\mathrm{meas}_1^1} & := & \mathtt{qbit} \multimap \mathtt{bit} \\
A_U & := & n\text{-}\mathtt{qbit} \multimap n\text{-}\mathtt{qbit} \quad \text{for a } 2^n \times 2^n \text{ matrix } U \\
A_{\mathrm{cmp}_{m,n}} & := & (m\text{-}\mathtt{qbit} \boxtimes n\text{-}\mathtt{qbit}) \multimap (m+n)\text{-}\mathtt{qbit}
\end{array}$$

Bookkeeping

(due to ⊗ vs. ⊠)

$$\frac{A <: A'}{!\,\Delta, x : A \vdash x : A'}\ (\mathrm{Ax.1}) \qquad \frac{!\,A_c <: A}{!\,\Delta \vdash c : A}\ (\mathrm{Ax.2})$$

$$\frac{\Delta \vdash M : !^n A}{\Delta \vdash \mathtt{inj}_\ell^B M : !^n(A+B)}\ (+.\mathrm{I}_1)$$

$$\frac{\Delta \vdash N : !^n B}{\Delta \vdash \mathtt{inj}_r^A N : !^n(A+B)}\ (+.\mathrm{I}_2)$$

$$\frac{!\,\Delta, \Gamma_1 \vdash P : !^n(A+B) \quad \begin{array}{c} !\,\Delta, \Gamma_2, x : !^n A \vdash M : C \\ !\,\Delta, \Gamma_2, y : !^n B \vdash N : C \end{array}}{\begin{array}{l} !\,\Delta, \Gamma_1, \Gamma_2 \\ \qquad \vdash \mathtt{match}\, P\, \mathtt{with}\, (x^{!^n A} \mapsto M \mid y^{!^n B} \mapsto N) : C \end{array}}\ (+.\mathrm{E}), (\dagger)$$

$$\frac{x : A, \Delta \vdash M : B}{\Delta \vdash \lambda x^A.M : A \multimap B}\ (\multimap.\mathrm{I}_1)$$

$$\frac{x : A, !\,\Delta \vdash M : B}{!\,\Delta \vdash \lambda x^A.M : !^n(A \multimap B)}\ (\multimap.\mathrm{I}_2)$$

$$\frac{!\,\Delta, \Gamma_1 \vdash M : A \multimap B \quad !\,\Delta, \Gamma_2 \vdash N : A}{!\,\Delta, \Gamma_1, \Gamma_2 \vdash MN : B}\ (\multimap.\mathrm{E}), (\dagger)$$

$$\frac{!\,\Delta, \Gamma_1 \vdash M_1 : !^n A_1 \quad !\,\Delta, \Gamma_2 \vdash M_2 : !^n A_2}{!\,\Delta, \Gamma_1, \Gamma_2 \vdash \langle M_1, M_2 \rangle : !^n(A_1 \boxtimes A_2)}\ (\boxtimes.\mathrm{I}), (\dagger)$$

$$\frac{}{!\,\Delta \vdash * : !^n \top}\ (\top.\mathrm{I})$$

$$\frac{\begin{array}{c} !\,\Delta, \Gamma_2, x_1 : !^n A_1, x_2 : !^n A_2 \vdash N : A \\ !\,\Delta, \Gamma_1 \vdash M : !^n(A_1 \boxtimes A_2) \end{array}}{!\,\Delta, \Gamma_1, \Gamma_2 \vdash \mathtt{let}\, \langle x_1^{!^n A_1}, x_2^{!^n A_2} \rangle = M\, \mathtt{in}\, N : A}\ (\boxtimes.\mathrm{E}), (\dagger)$$

$$\frac{!\,\Delta, \Gamma_1 \vdash M : \top \quad !\,\Delta, \Gamma_2 \vdash N : A}{!\,\Delta, \Gamma_1, \Gamma_2 \vdash \mathtt{let}\, * = M\, \mathtt{in}\, N : A}\ (\top.\mathrm{E}), (\dagger)$$

$$\frac{\begin{array}{c} !\,\Delta, \Gamma, f : !(A \multimap B) \vdash N : C \\ !\,\Delta, f : !(A \multimap B), x : A \vdash M : B \end{array}}{!\,\Delta, \Gamma \vdash \mathtt{letrec}\, f^{A \multimap B} x = M\, \mathtt{in}\, N : C}\ (\mathrm{rec}), (\dagger)$$

Operational Semantics

$$\begin{aligned}
&E[\,(\lambda x^{A}.M)V\,] \to_1 E[\,M[V/x]\,]\\
&E[\,\mathtt{let}\,\langle x^{A}, y^{B}\rangle = \langle V, W\rangle\,\mathtt{in}\,M\,] \to_1 E[\,M[V/x, W/y]\,]\\
&E[\,\mathtt{let}\,* = *\,\mathtt{in}\,M\,] \to_1 E[\,M\,]\\
&E[\,\mathtt{match}\,(\mathtt{inj}^{B}_{\ell}V)\,\mathtt{with}\,(x^{!^n A} \mapsto M \mid y^{!^n B} \mapsto N)\,]\\
&\qquad\qquad\qquad\qquad\qquad\qquad \to_1 E[\,M[V/x]\,]\\
&E[\,\mathtt{match}\,(\mathtt{inj}^{A}_{r}V)\,\mathtt{with}\,(x^{!^n A} \mapsto M \mid y^{!^n B} \mapsto N)\,]\\
&\qquad\qquad\qquad\qquad\qquad\qquad \to_1 E[\,N[V/y]\,]\\
&E[\,\mathtt{letrec}\,f^{A\multimap B}x = M\,\mathtt{in}\,N\,]\\
&\qquad\qquad \to_1 E[\,N[\lambda x^{A}.\mathtt{letrec}\,f^{A\multimap B}x = M\,\mathtt{in}\,M/f]\,]\\
&E[\,\mathtt{meas}^{n+1}_{i}(\mathtt{new}\,\rho)\,] \to_1 E[\,\langle\,\mathtt{tt},\,\mathtt{new}\,\langle 0_i|\rho|0_i\rangle\,\rangle\,]\\
&E[\,\mathtt{meas}^{n+1}_{i}(\mathtt{new}\,\rho)\,] \to_1 E[\,\langle\,\mathtt{ff},\,\mathtt{new}\,\langle 1_i|\rho|1_i\rangle\,\rangle\,]\\
&E[\,\mathtt{meas}^{1}_{1}(\mathtt{new}\,\rho)\,] \to_{\langle 0|\rho|0\rangle} E[\,\mathtt{tt}\,]\\
&E[\,\mathtt{meas}^{1}_{1}(\mathtt{new}\,\rho)\,] \to_{\langle 1|\rho|1\rangle} E[\,\mathtt{ff}\,]\\
&E[\,U(\mathtt{new}\,\rho)\,] \to_1 E[\,\mathtt{new}\,(U\rho)\,]\\
&E[\,\mathtt{cmp}_{m,n}\langle\mathtt{new}\,\rho, \mathtt{new}\,\sigma\rangle\,] \to_1 E[\,\mathtt{new}\,(\rho\otimes\sigma)\,]
\end{aligned}$$

- Standard small-step one, CBV, but with probabilistic branching (measurement)