Coalgebras and Higher-Order Computation: a GoI Approach

Ichiro Hasuo

University of Tokyo (JP)

東京大学
THE UNIVERSITY OF TOKYO

FSCD 2016, Porto, 24 Jun 2016

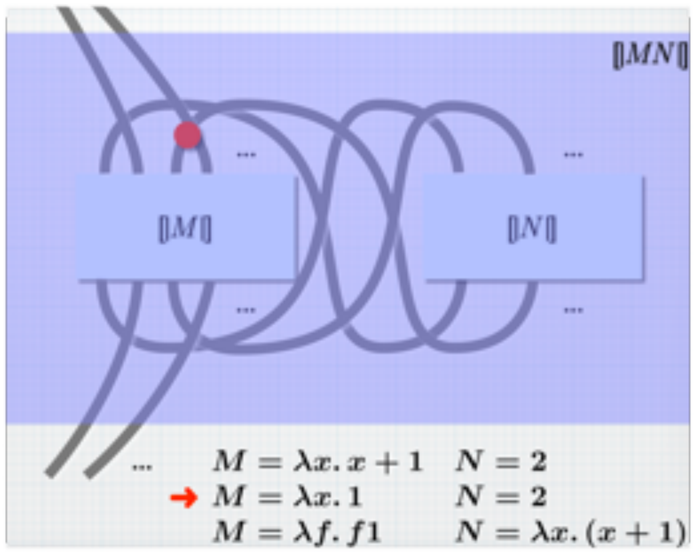
Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

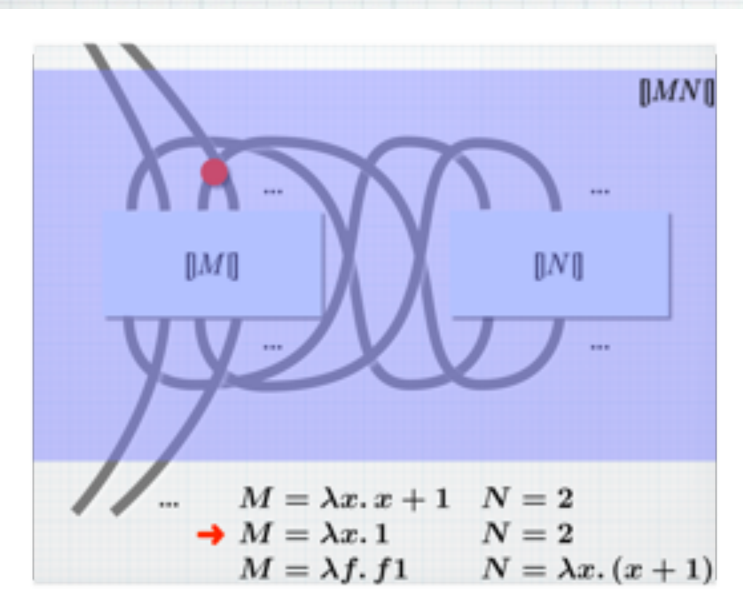
"GoI Animation"

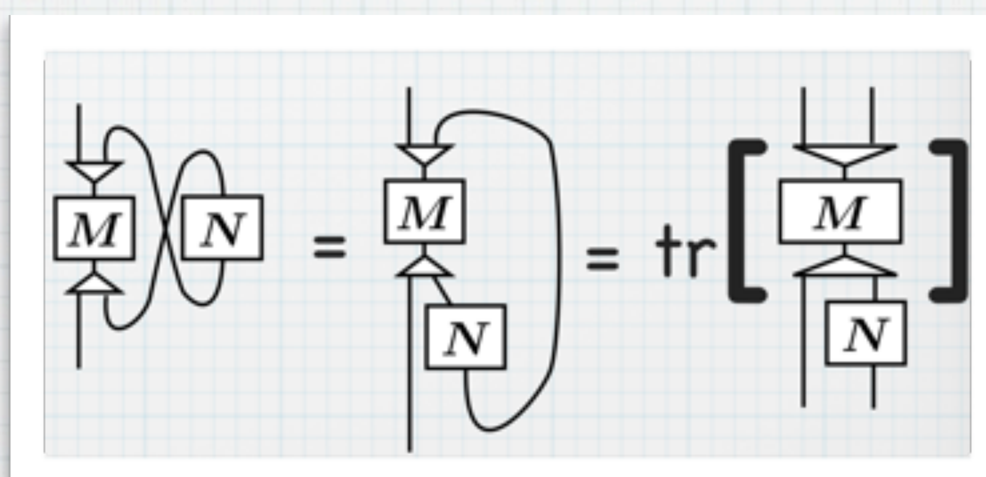


Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

"GoI Animation"





Categorical GoI

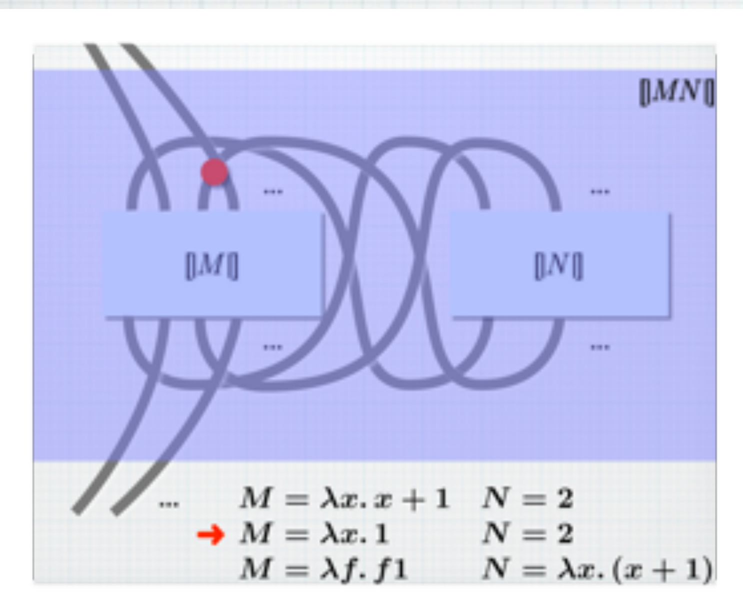
[Abramsky, Haghverdi & Scott, MSCS'02]

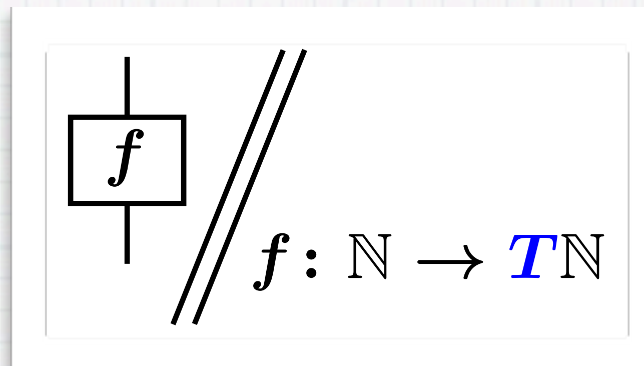
Hasuo (Tokyo)

Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

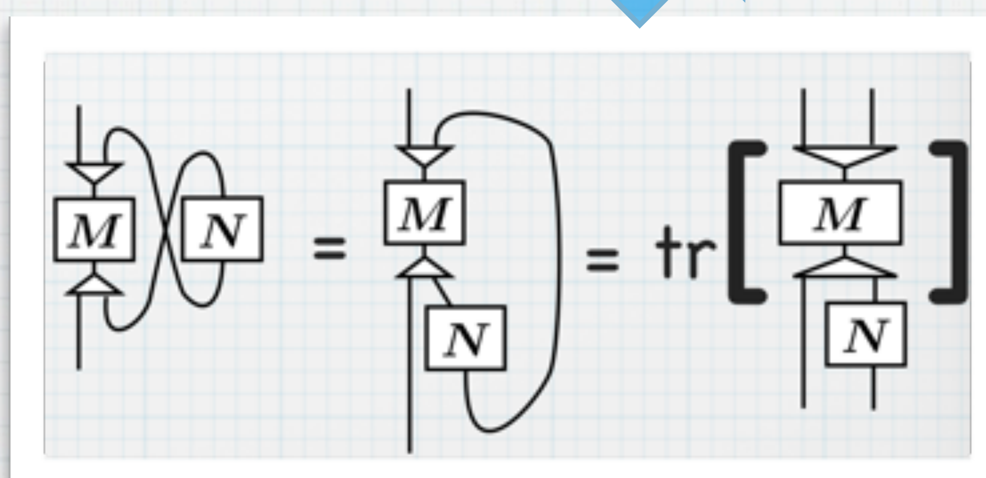
"GoI Animation"





GoI w/ T-branching

[IH & Hoshino, LICS'11]



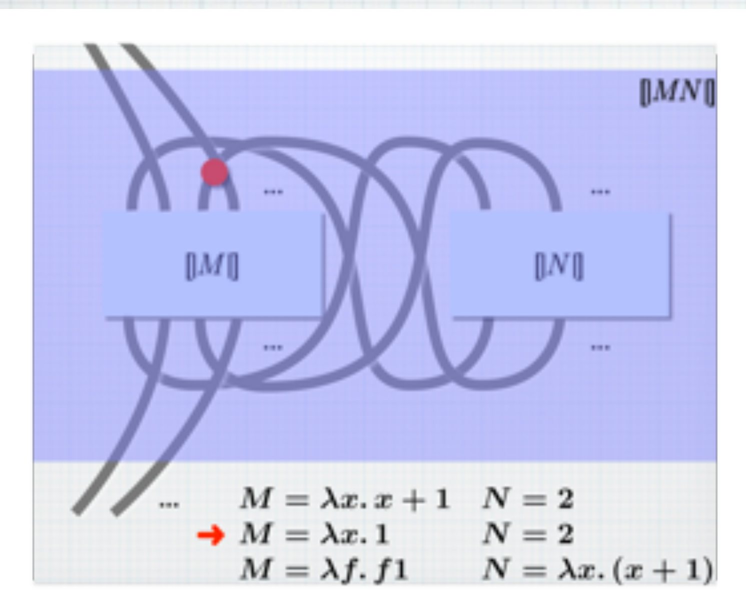
Categorical GoI

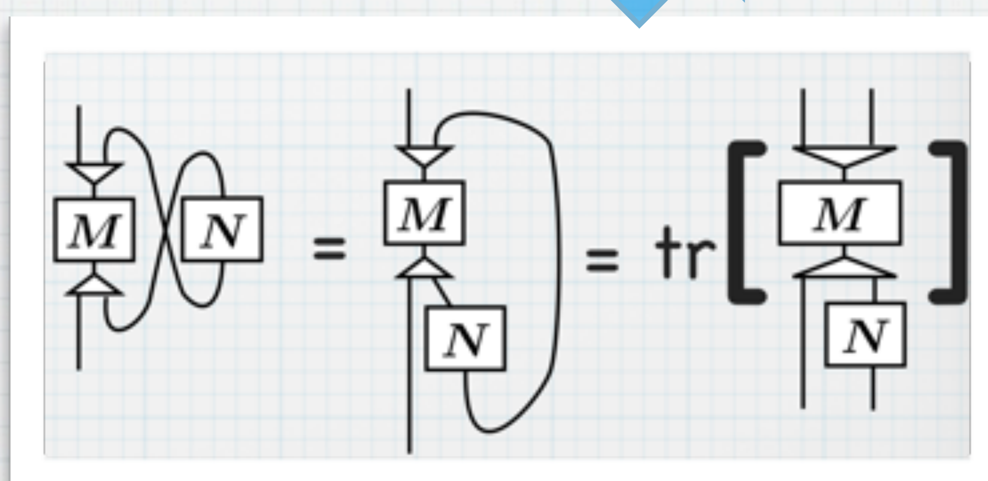
[Abramsky, Haghverdi & Scott, MSCS'02]

Outline

Coalgebra meets **higher-order computation**
in **Geometry of Interaction** [Girard, LC'88]

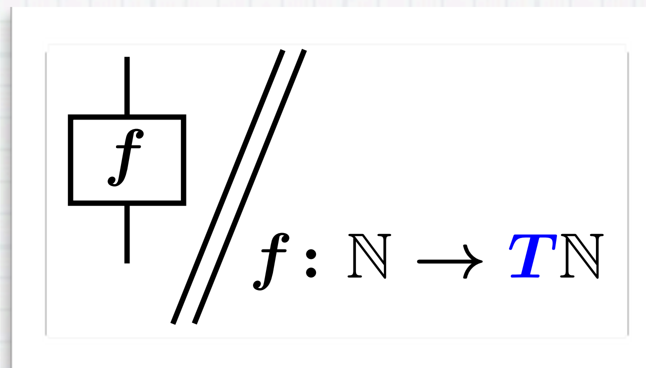
"GoI Animation"





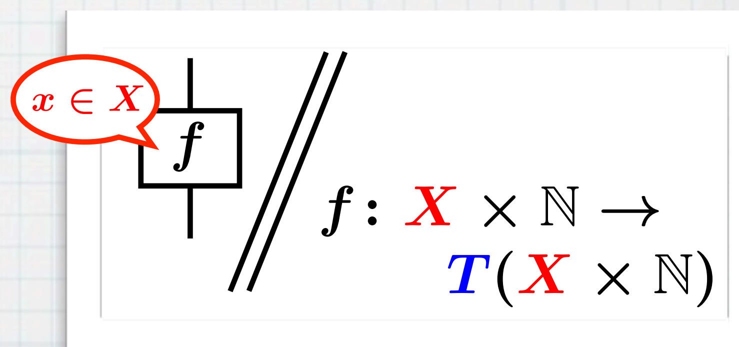
Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]



GoI w/ T-branching

[IH & Hoshino, LICS'11]



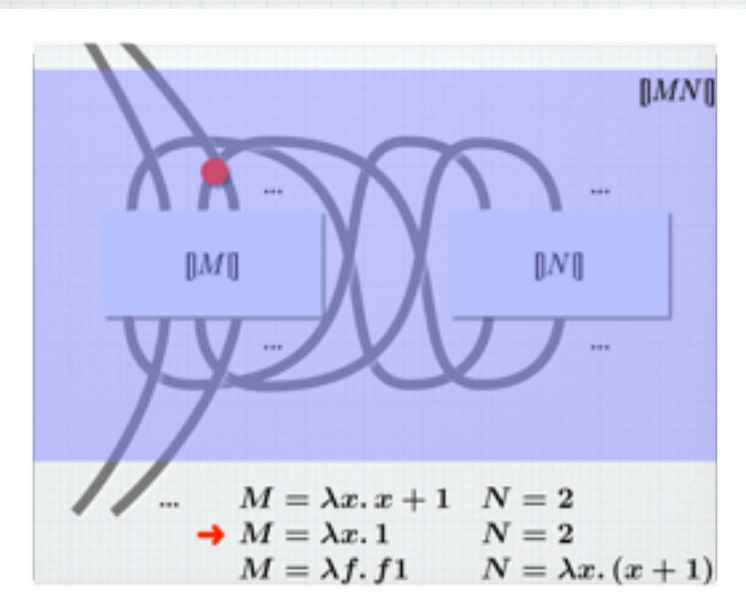
Memoryful GoI

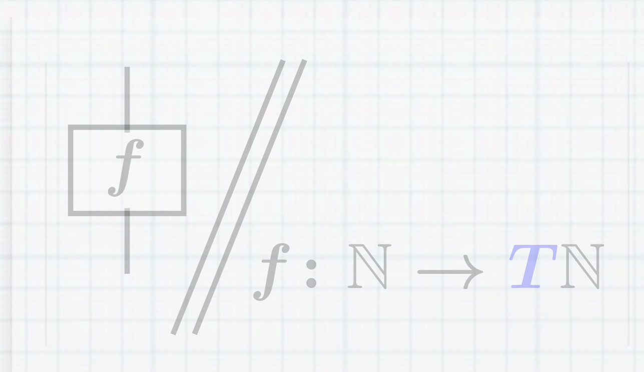
[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

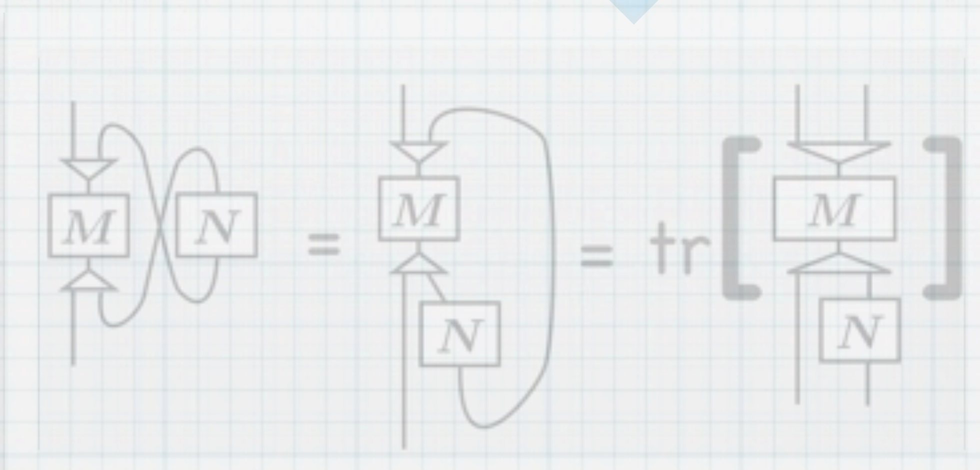
"GoI Animation"





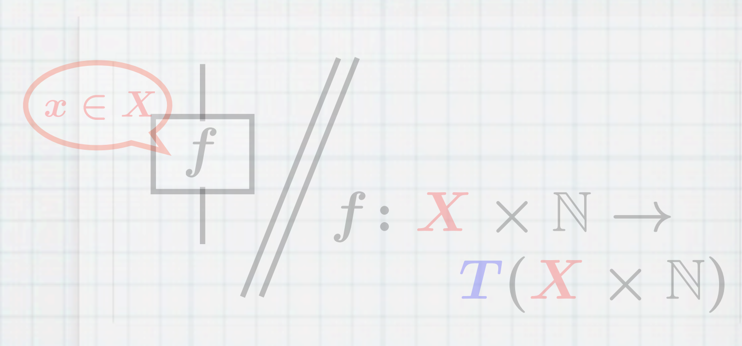
GoI w/ T-branching

[IH & Hoshino, LICS'11]



Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]



Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

Hasuo (Tokyo)

Collaborators

Collaborators

Naohiko Hoshino

(Kyoto U)

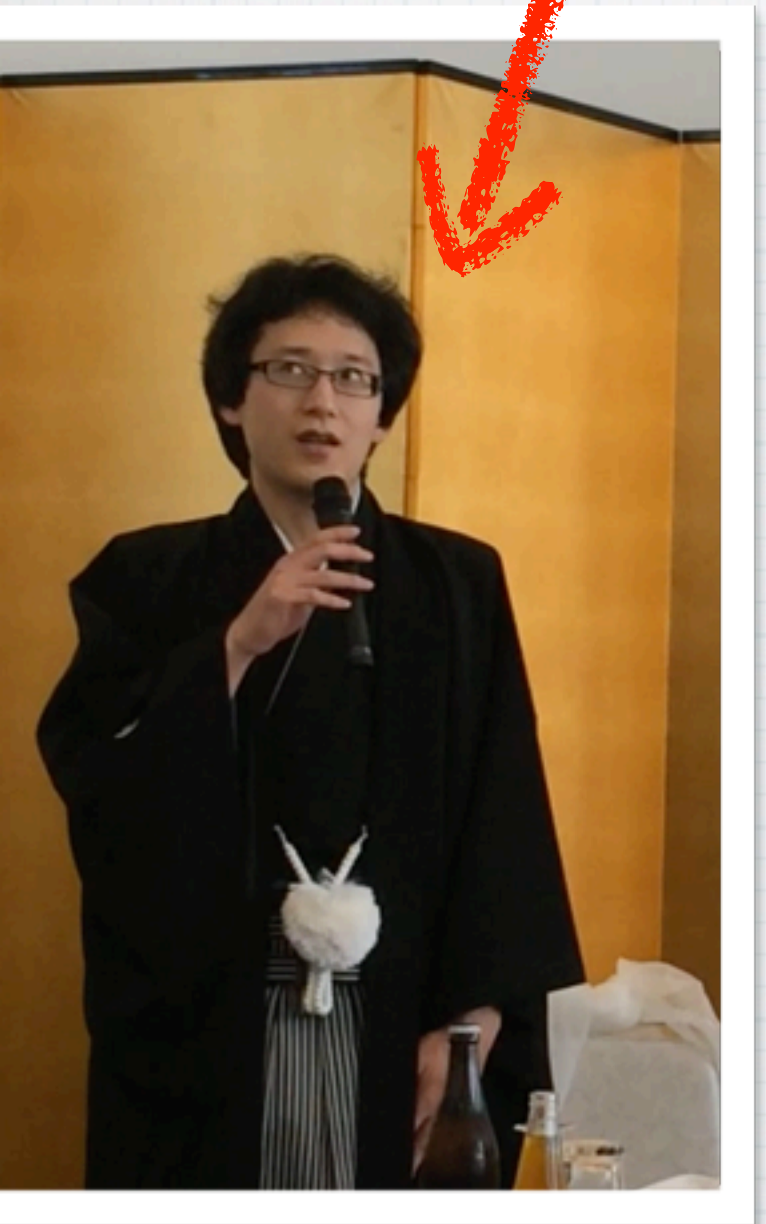
Collaborators

Koko Muroya

(Tokyo => Birmingham)

Naohiko Hoshino

(Kyoto U)



Hasuo (Tokyo)

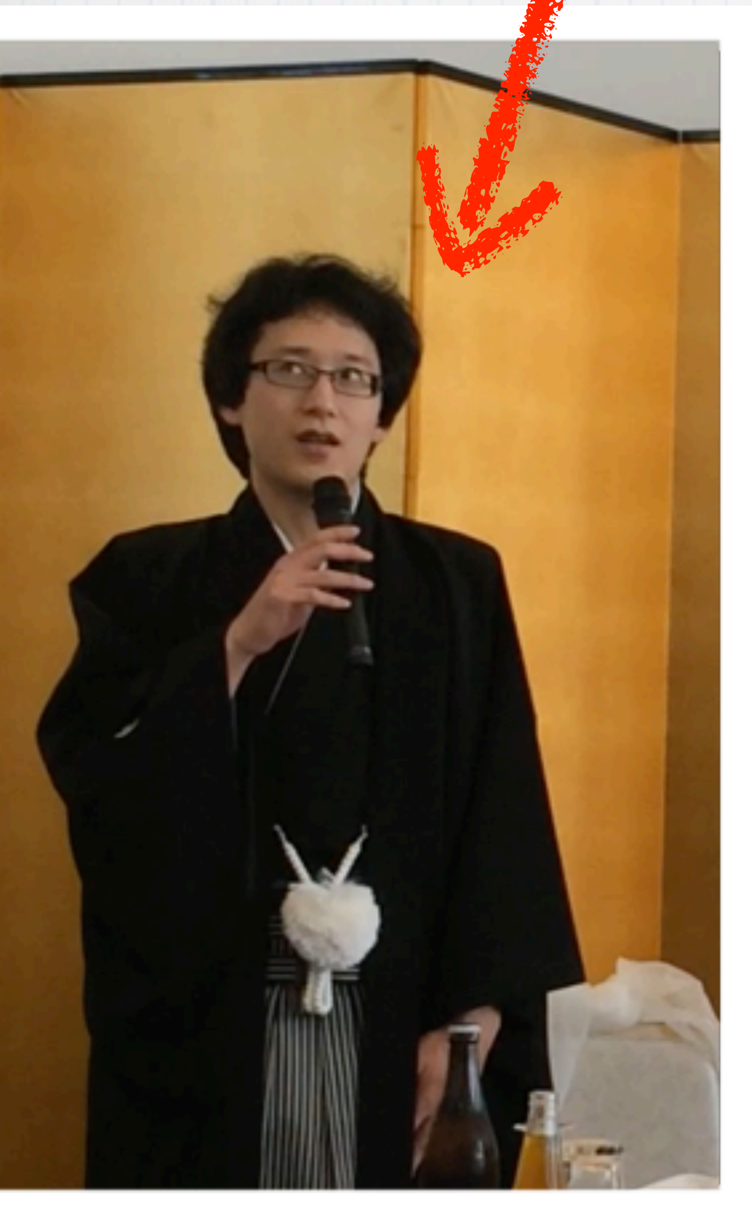
Collaborators

Koko Muroya

(Tokyo => Birmingham)

Naohiko Hoshino

(Kyoto U)



Bart Jacobs

(Nijmegen)

Hasuo (Tokyo)

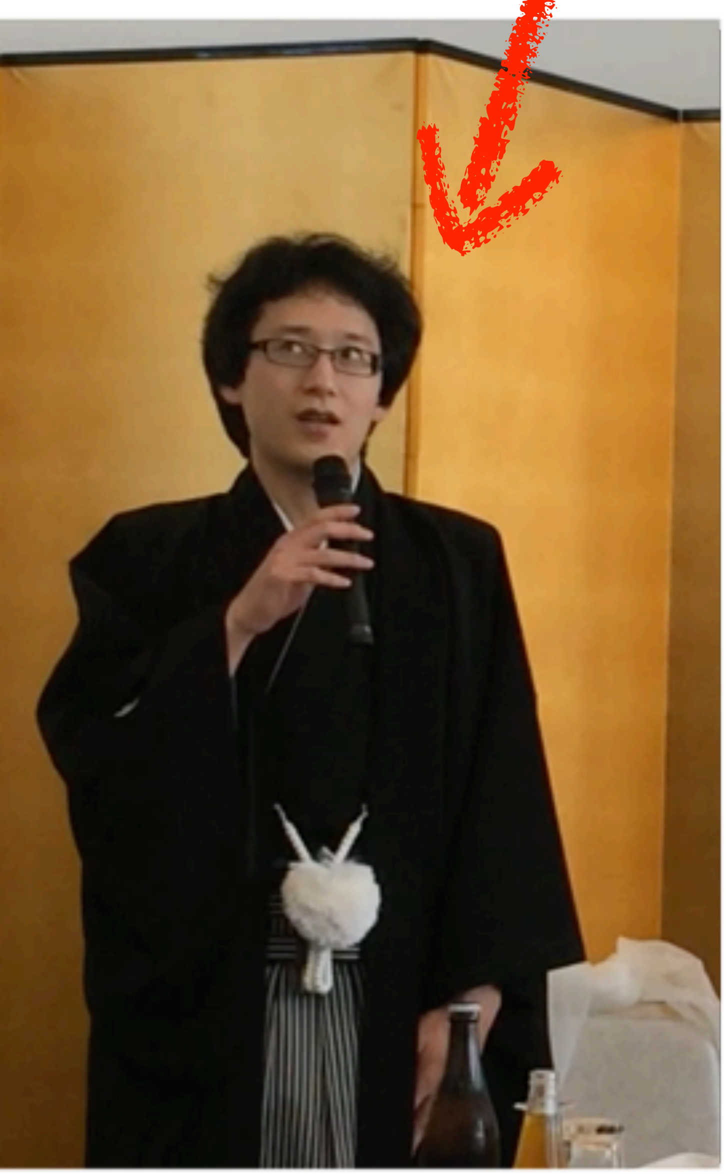
Collaborators

Koko Muroya

(Tokyo => Birmingham)

Naohiko Hoshino

(Kyoto U)



Bart Jacobs

(Nijmegen)

Toshiki Kataoka

(Tokyo)

Hasuo (Tokyo)

References

* [LICS 2011] IH and Naohiko Hoshino. **Semantics of Higher-Order Quantum Computation via Geometry of Interaction.** (Extended ver. to appear in Annals Pure & Appl. Logic)
* [CSL-LICS 2014] Naohiko Hoshino, Koko Muroya and IH. **Memoryful Geometry of Interaction: From Coalgebraic Components to Algebraic Effects.**
* [POPL 2016] Koko Muroya, Naohiko Hoshino and IH. **Memoryful Geometry of Interaction II: Recursion and Adequacy.**
* [LOLA 2014] Koko Muroya, Toshiki Kataoka, IH and Naohiko Hoshino. **Compiling Effectful Terms to Transducers: Prototype Implementation of Memoryful Geometry of Interaction (Preliminary Report).**
* [Math. Str. in Comp. Sci. 2011] IH and Bart Jacobs. **Traces for Coalgebraic Components.**

Geometry of Interaction (GoI)

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

Geometry of Interaction (GoI)

* J.-Y. Girard, at Logic Colloquium '88
* Provides **"denotational" semantics** (w/ operational flavor) for linear λ-term M

Geometry of Interaction (GoI)

* J.-Y. Girard, at Logic Colloquium '88
* Provides **"denotational" semantics** (w/ operational flavor) for linear λ-term M
* As a **compilation technique**

  [Mackie, POPL'95] [Pinto, TLCA'01] [Ghica et al., POPL'07, POPL'11, ICFP'11, ...]

Geometry of Interaction (GoI)

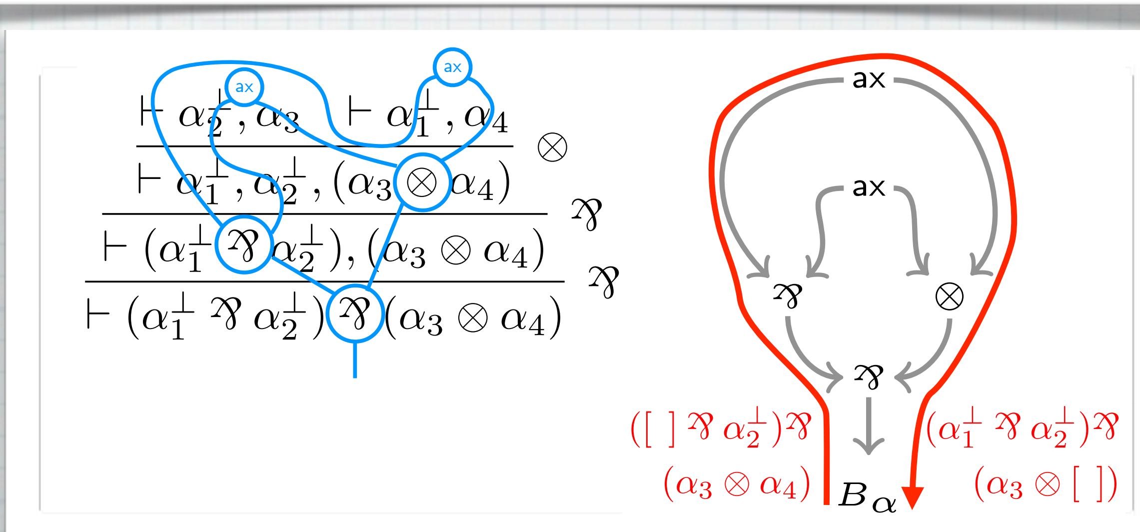
* J.-Y. Girard, at Logic Colloquium '88
* Provides **"denotational" semantics** (w/ operational flavor) for linear λ-term M
* As a **compilation technique**

  [Mackie, POPL'95] [Pinto, TLCA'01] [Ghica et al., POPL'07, POPL'11, ICFP'11, ...]
* Two presentations:
  * (Operator-) Algebraic [Girard]
  * Token machines/ interaction abstract machines

    [Danos & Regnier, TCS'99] [Mackie, POPL'95]

$$\dfrac{\dfrac{\overline{\vdash A,\ A^\perp} \quad \overline{\vdash A^\perp,\ A}}{\vdash A,\ A^\perp,\ A^\perp \otimes A} \quad \dfrac{\overline{\vdash A,\ A^\perp}}{\vdash A ⅋ A^\perp}}{\vdash [A^\perp \otimes A],\ A,\ A^\perp} \qquad \Pi^\bullet = \begin{pmatrix} 0 & 0 & p & q \\ 0 & pq^* + qp^* & 0 & 0 \\ p^* & 0 & 0 & 0 \\ q^* & 0 & 0 & 0 \end{pmatrix} \quad \sigma = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

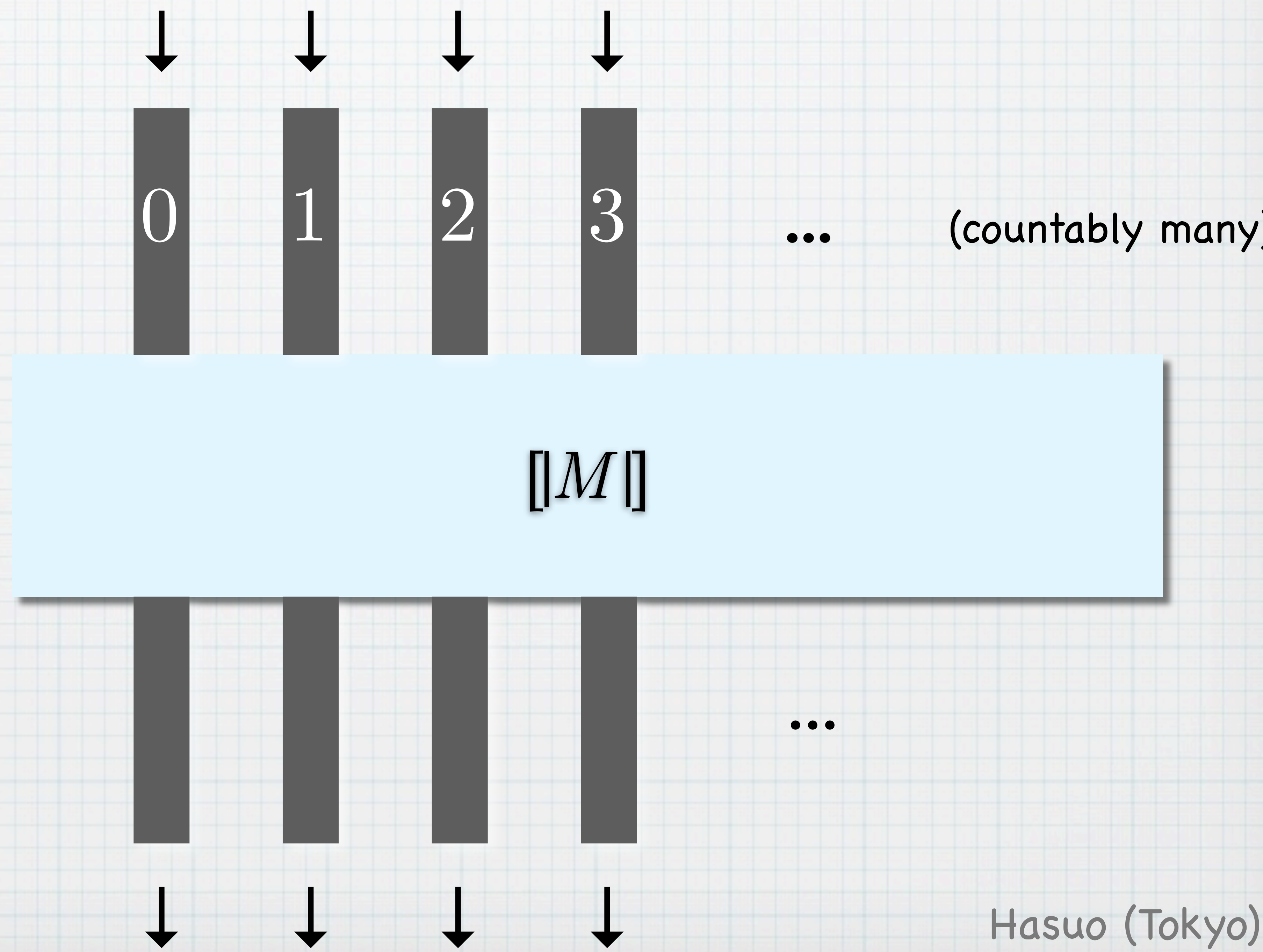
$$(p^*p = q^*q = 1,\ p^*q = q^*p = 0)$$



The GoI Animation

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

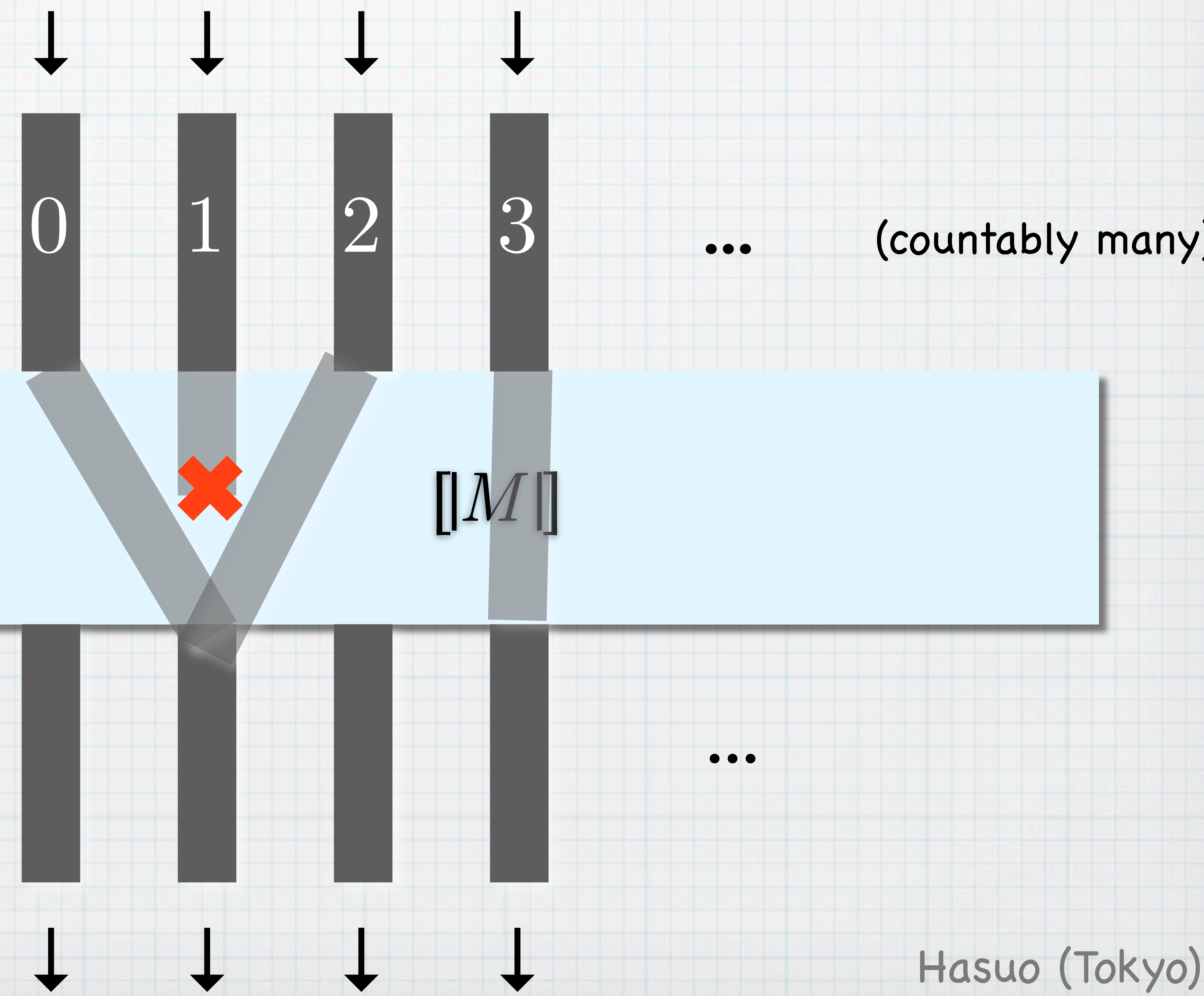
$$=\; \text{“piping”}$$



The GoI Animation

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

$$=\ \text{“piping”}$$

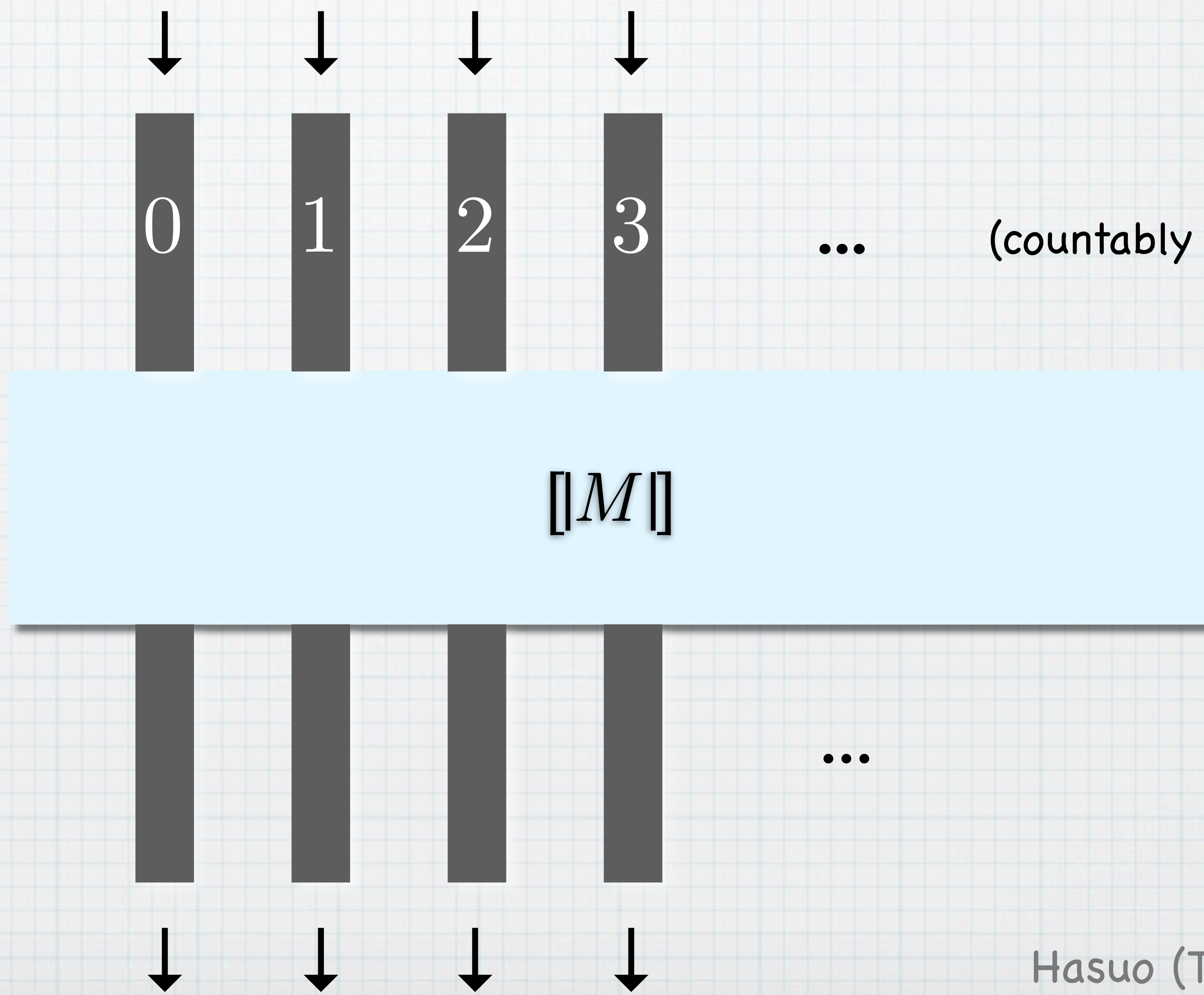


Hasuo (Tokyo)

The GoI Animation

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

$= \;$ "piping"

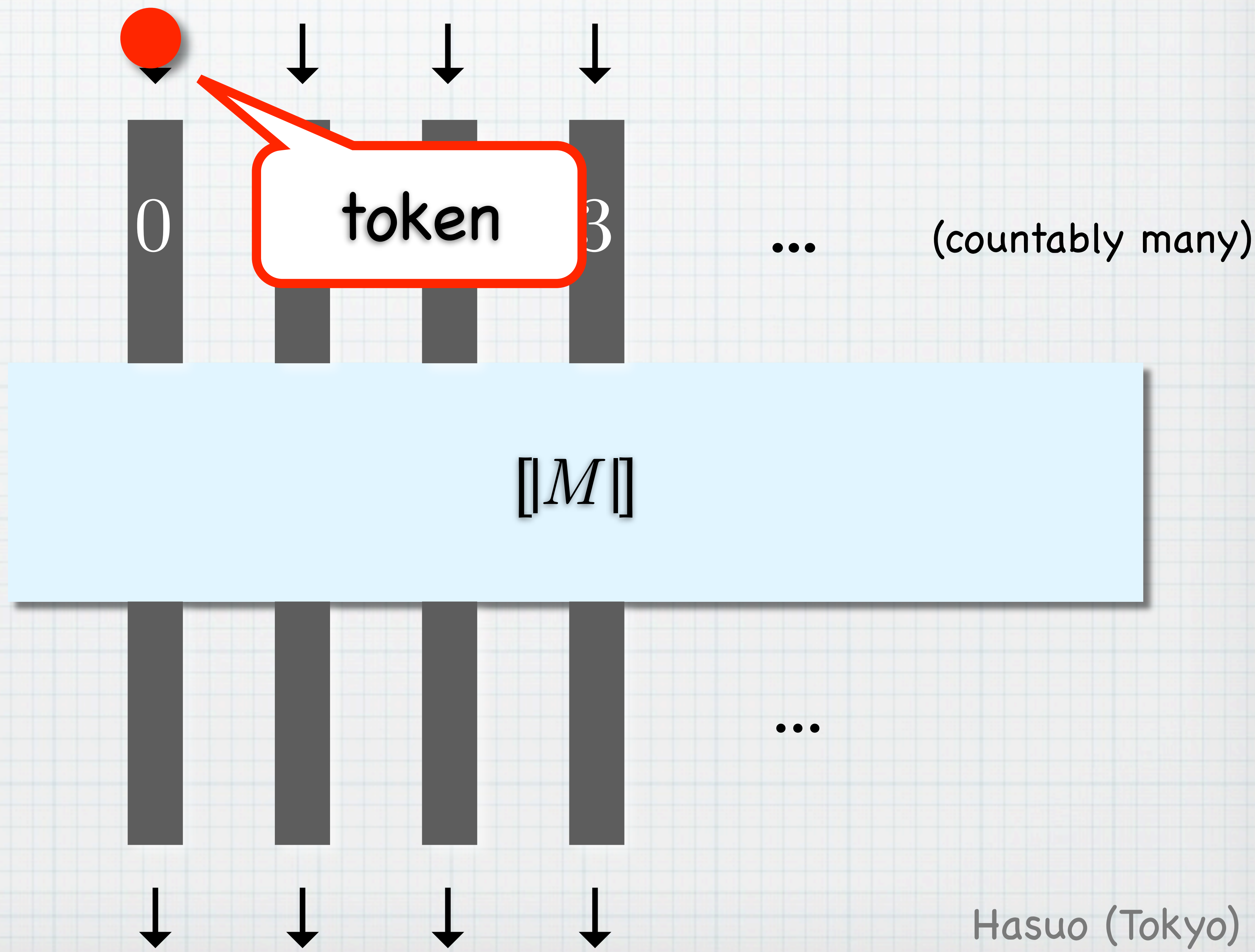


(countably many)

The GoI Animation

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

$=$ "piping"

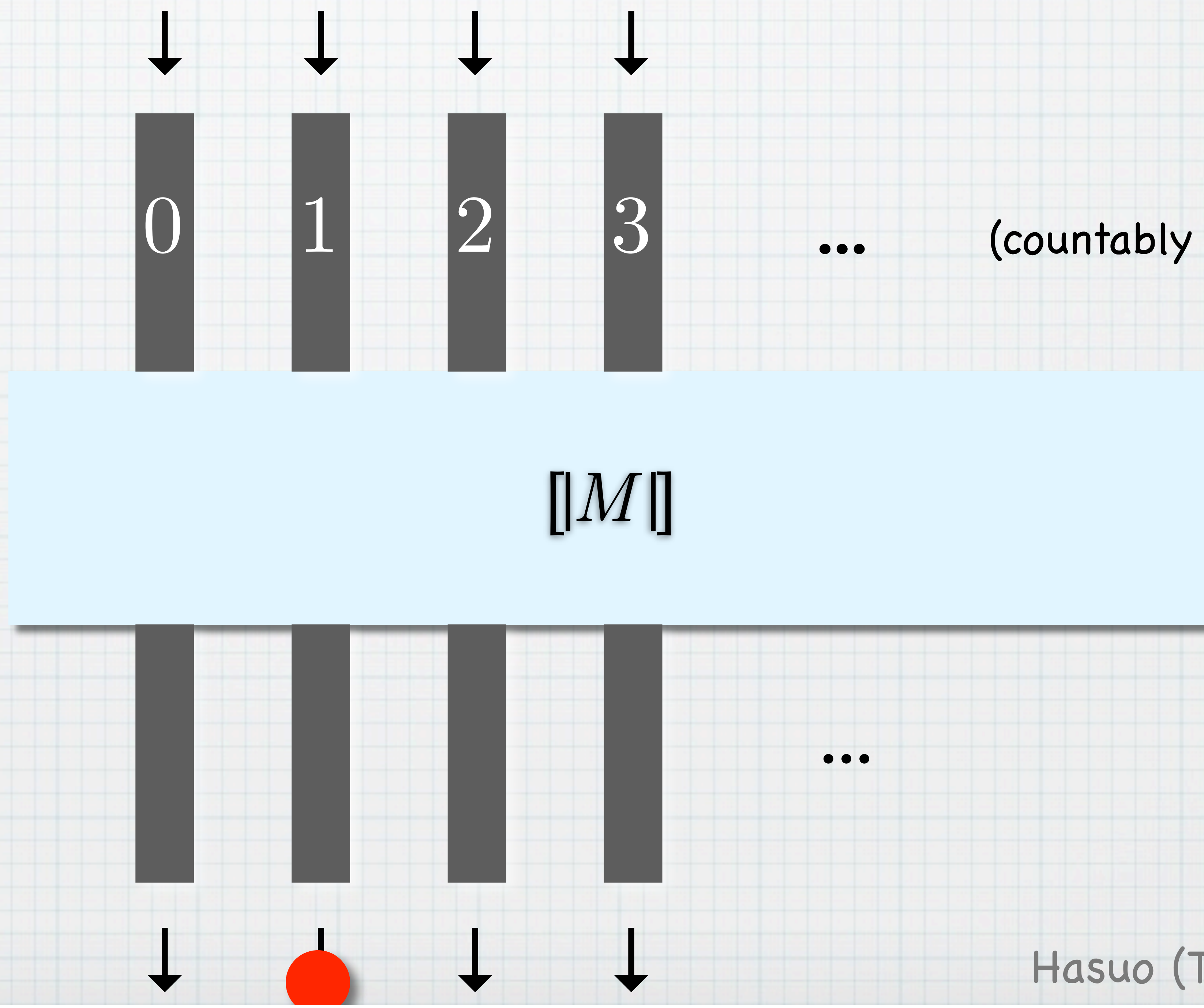


Hasuo (Tokyo)

The GoI Animation

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

$$=\ \text{“piping”}$$

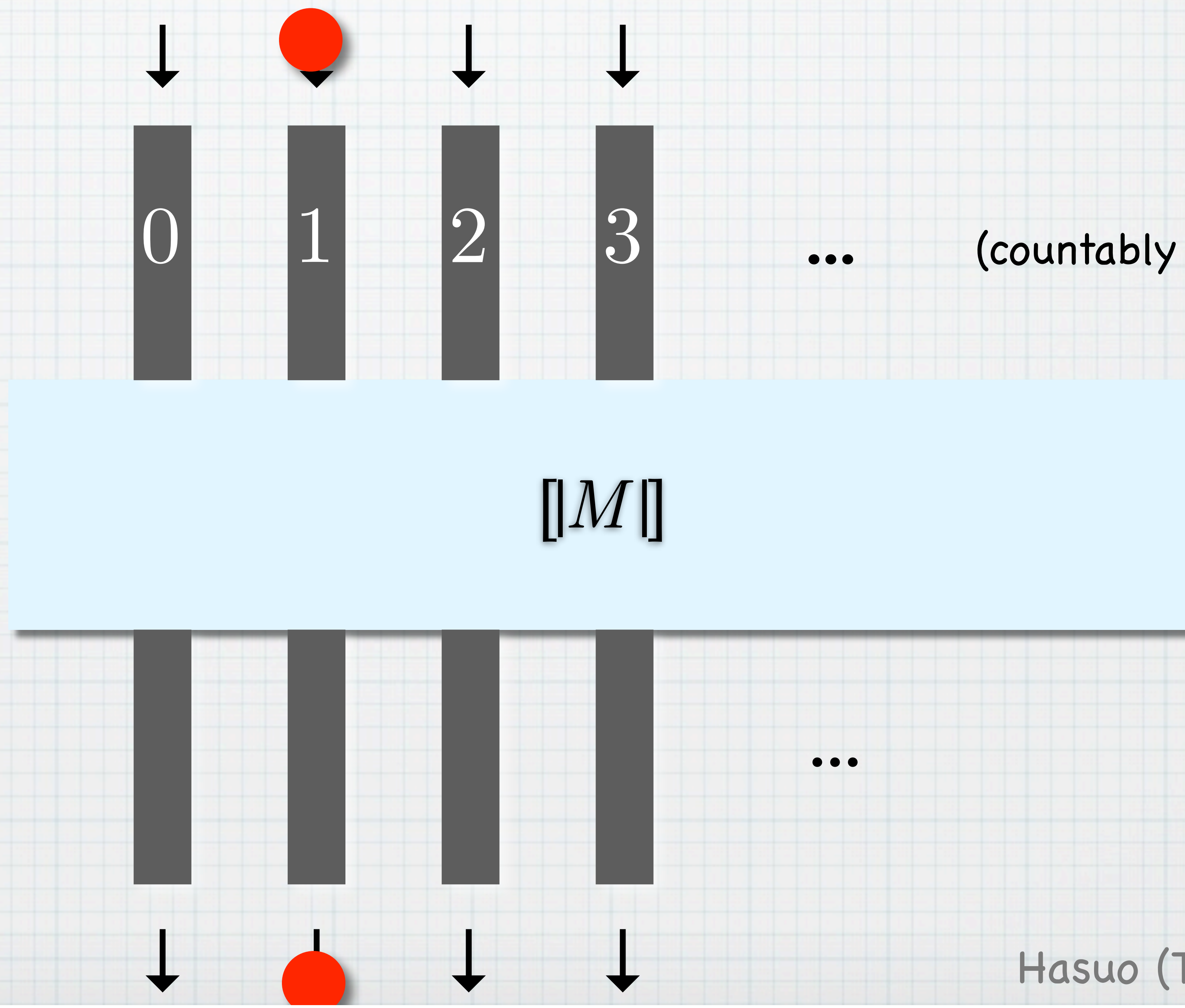


(countably many)

The GoI Animation

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

$$=\ \text{“piping”}$$

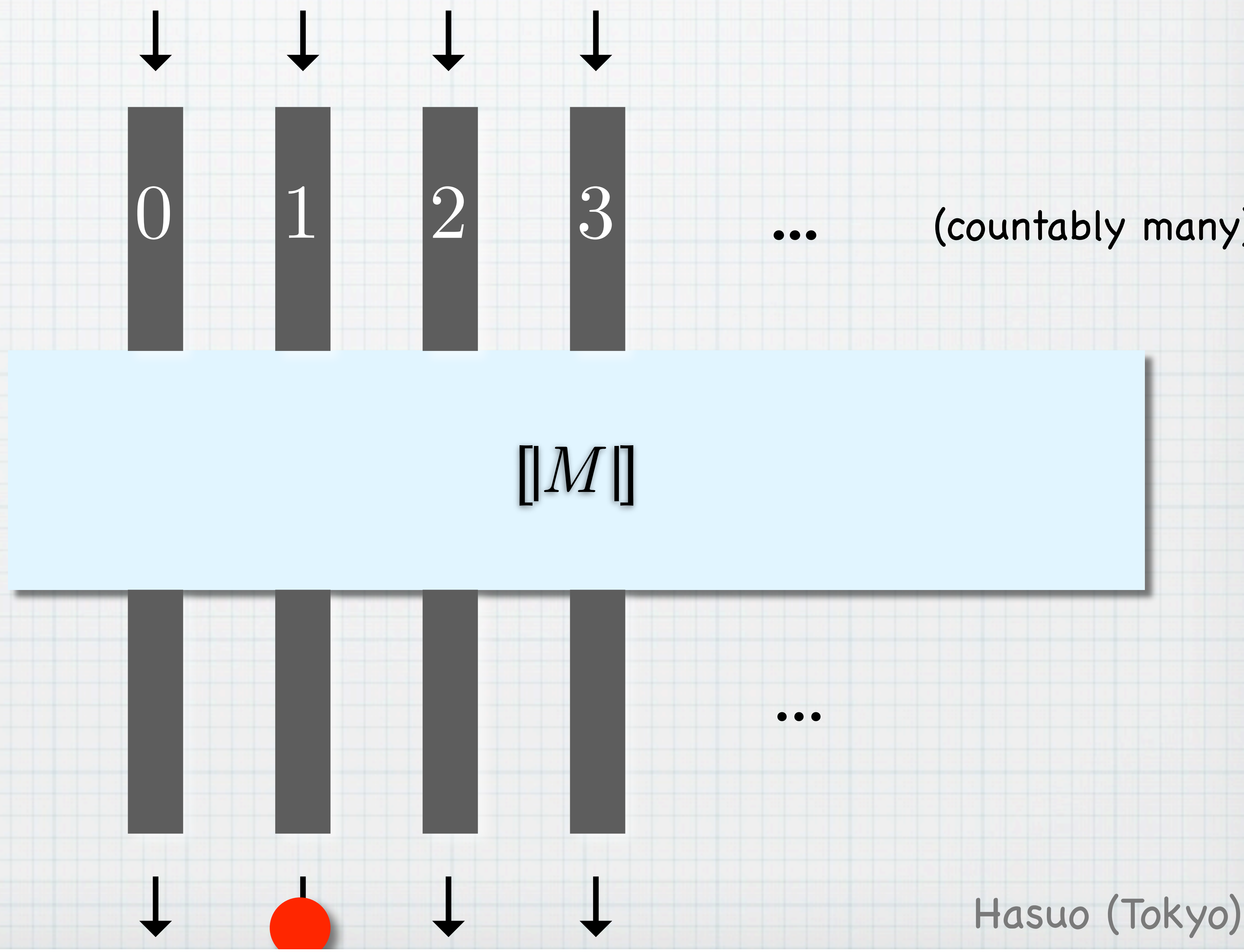


(countably many)

The GoI Animation

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

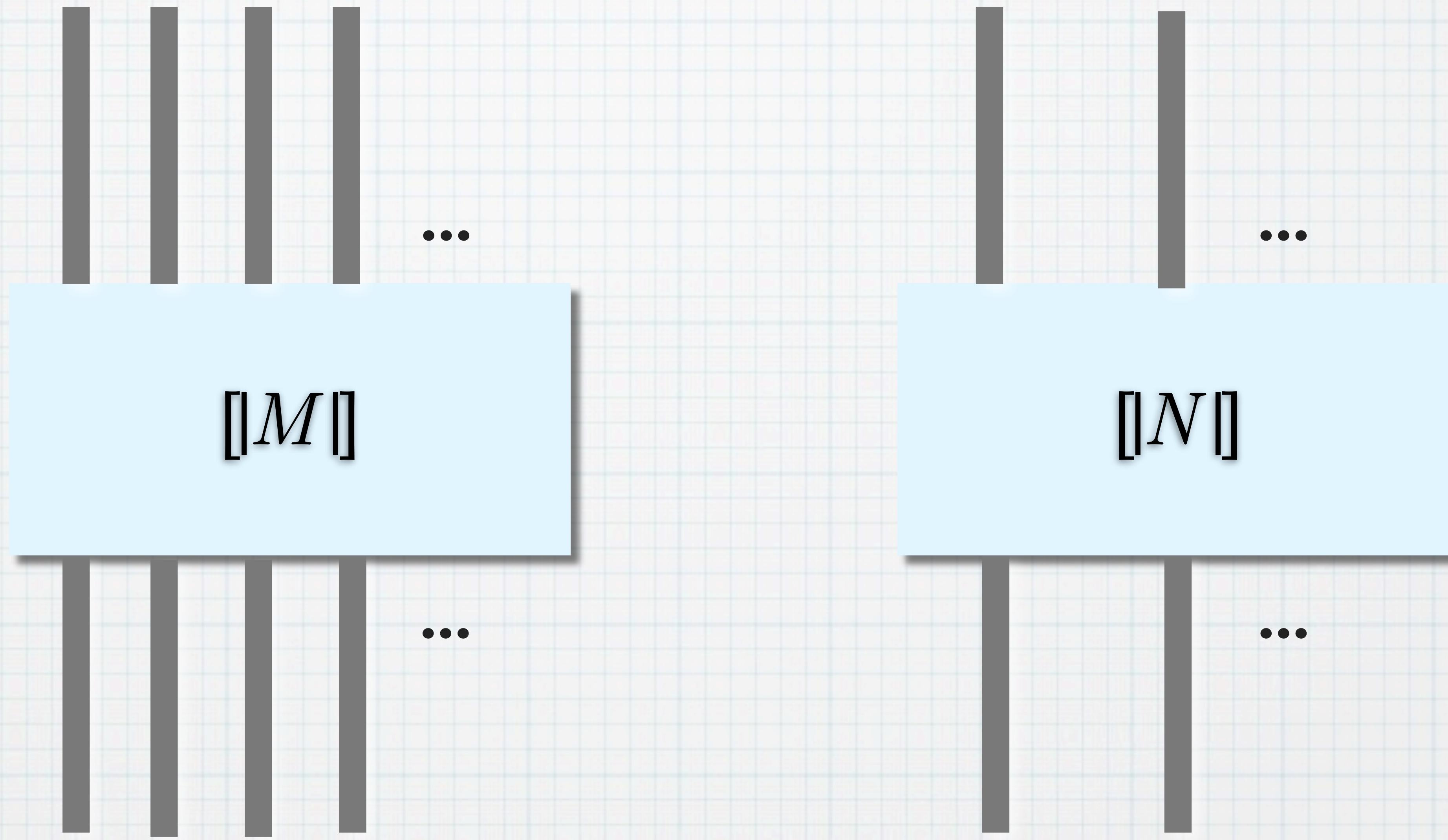
$$=\ \text{“piping”}$$



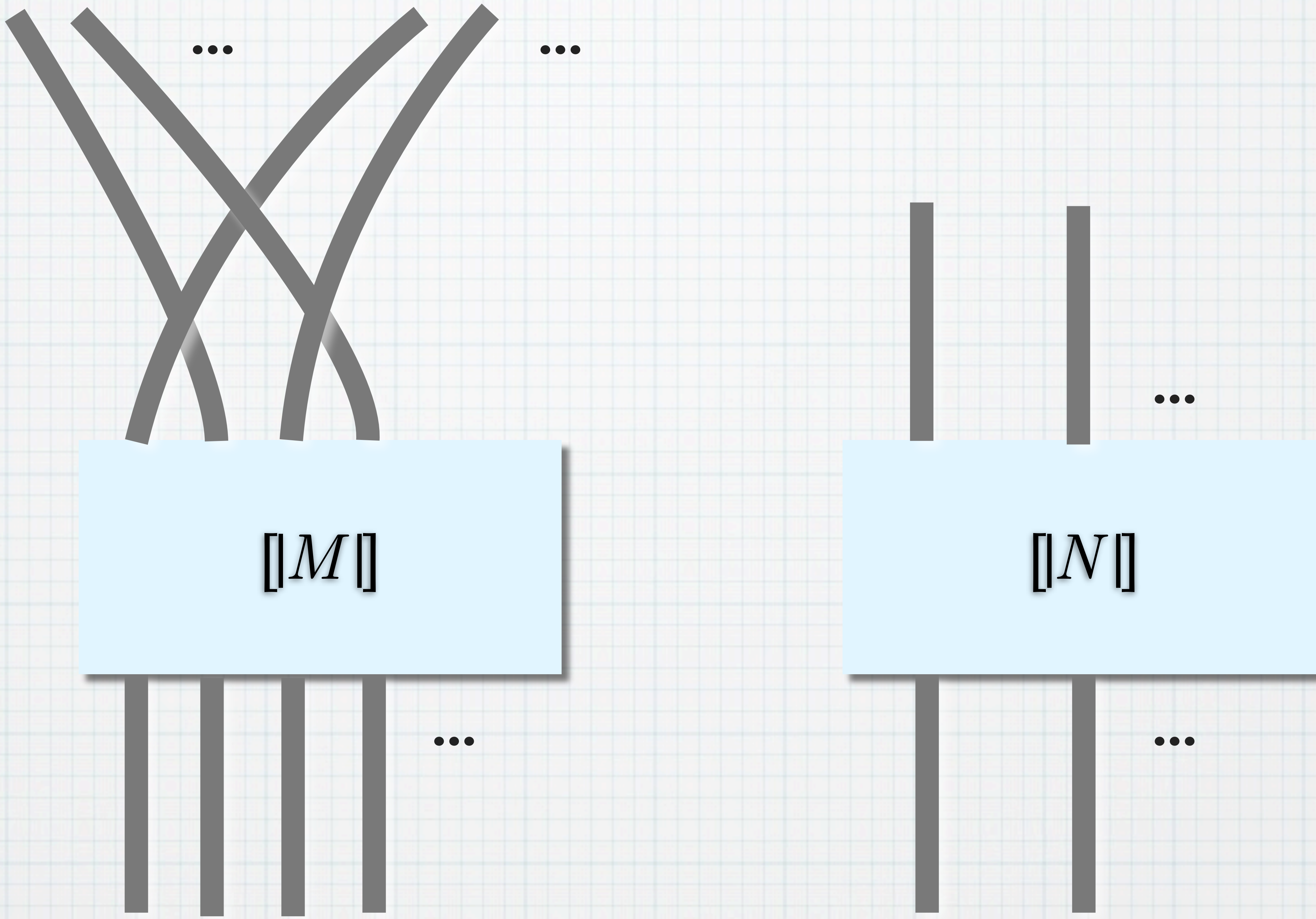
The GoI Animation

* Function application $[\![M N]\!]$
    * by "parallel composition + hiding"

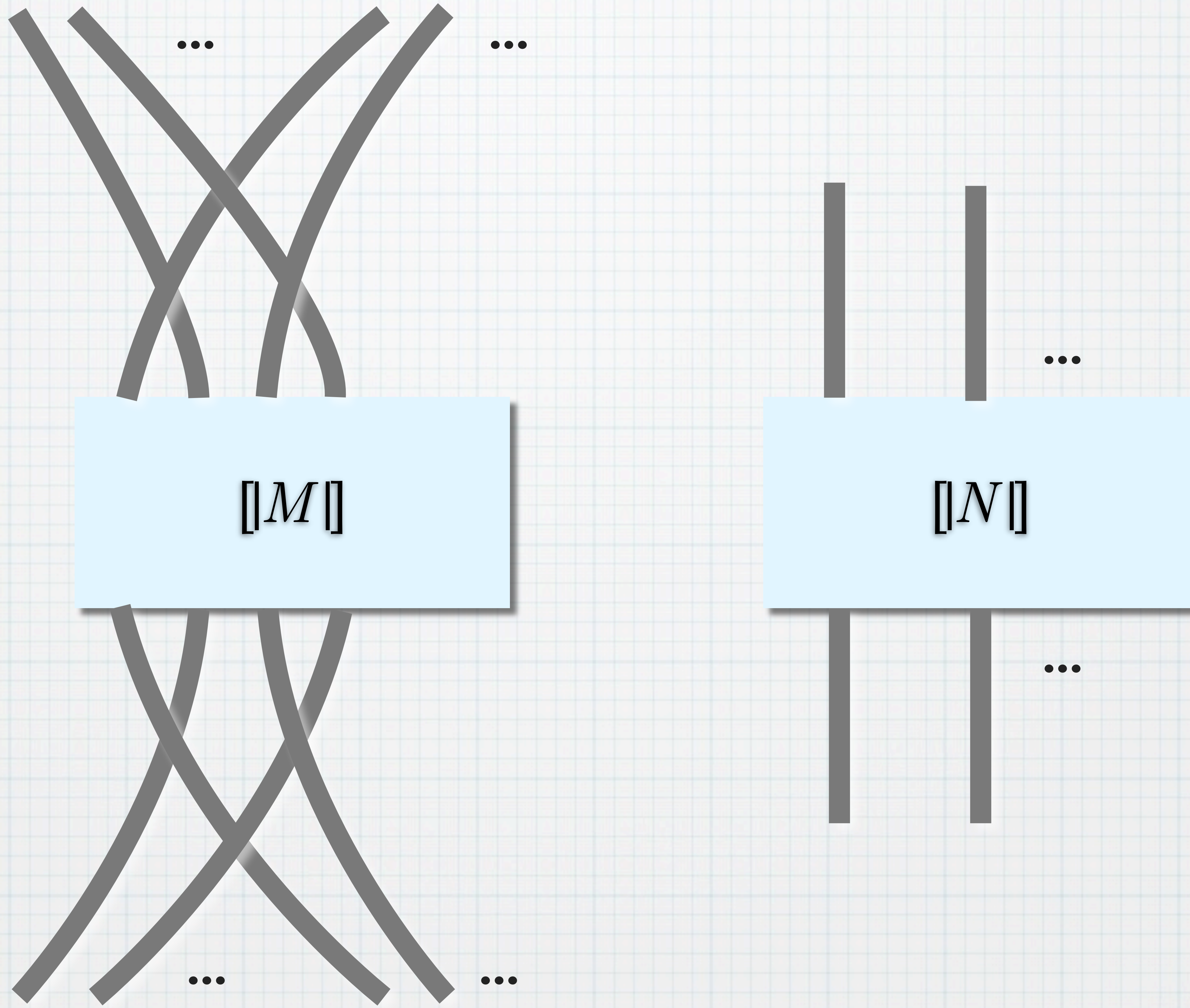
$$[\![MN]\!] =$$



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

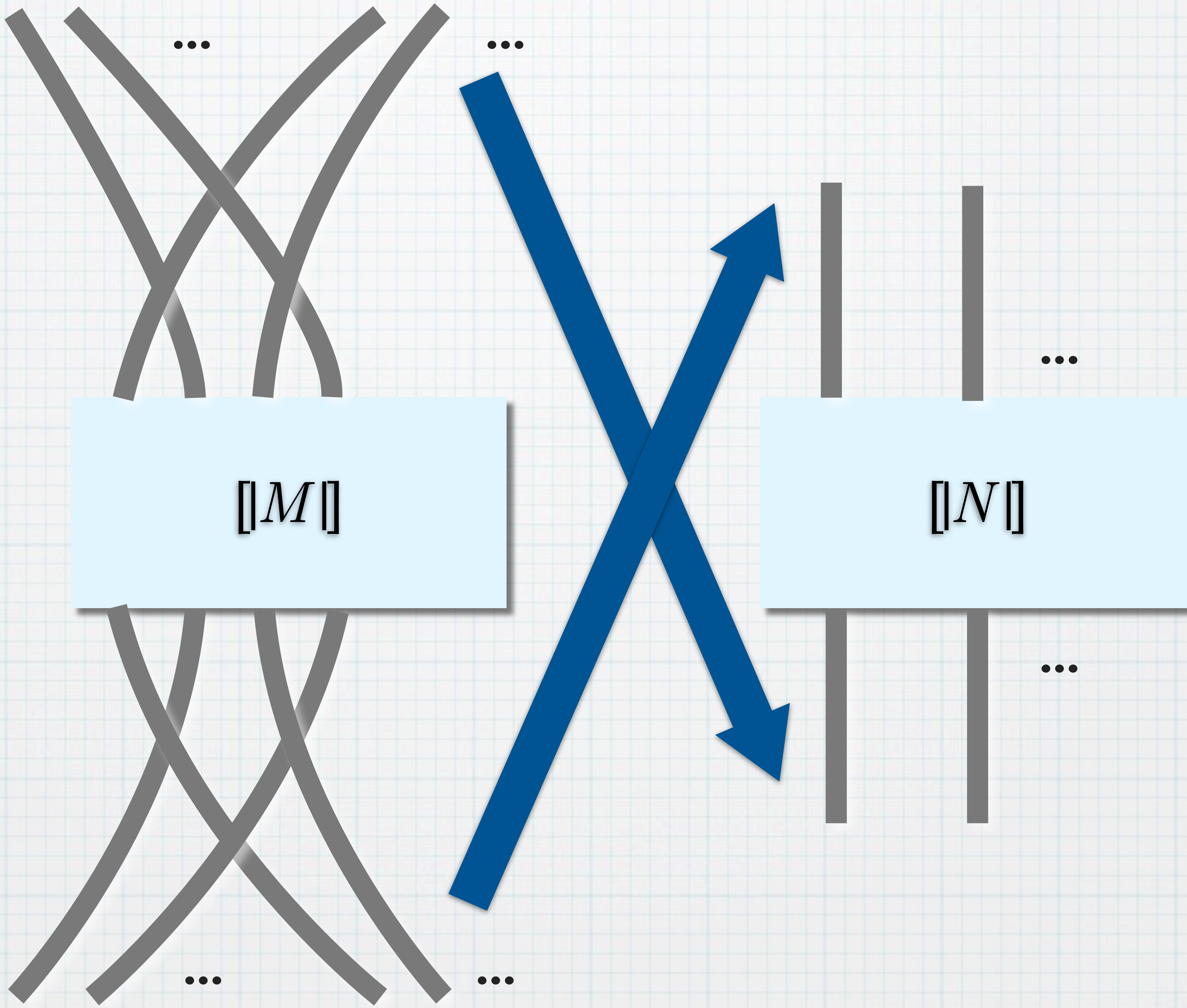


$$[\![\boldsymbol{MN}]\!] =$$

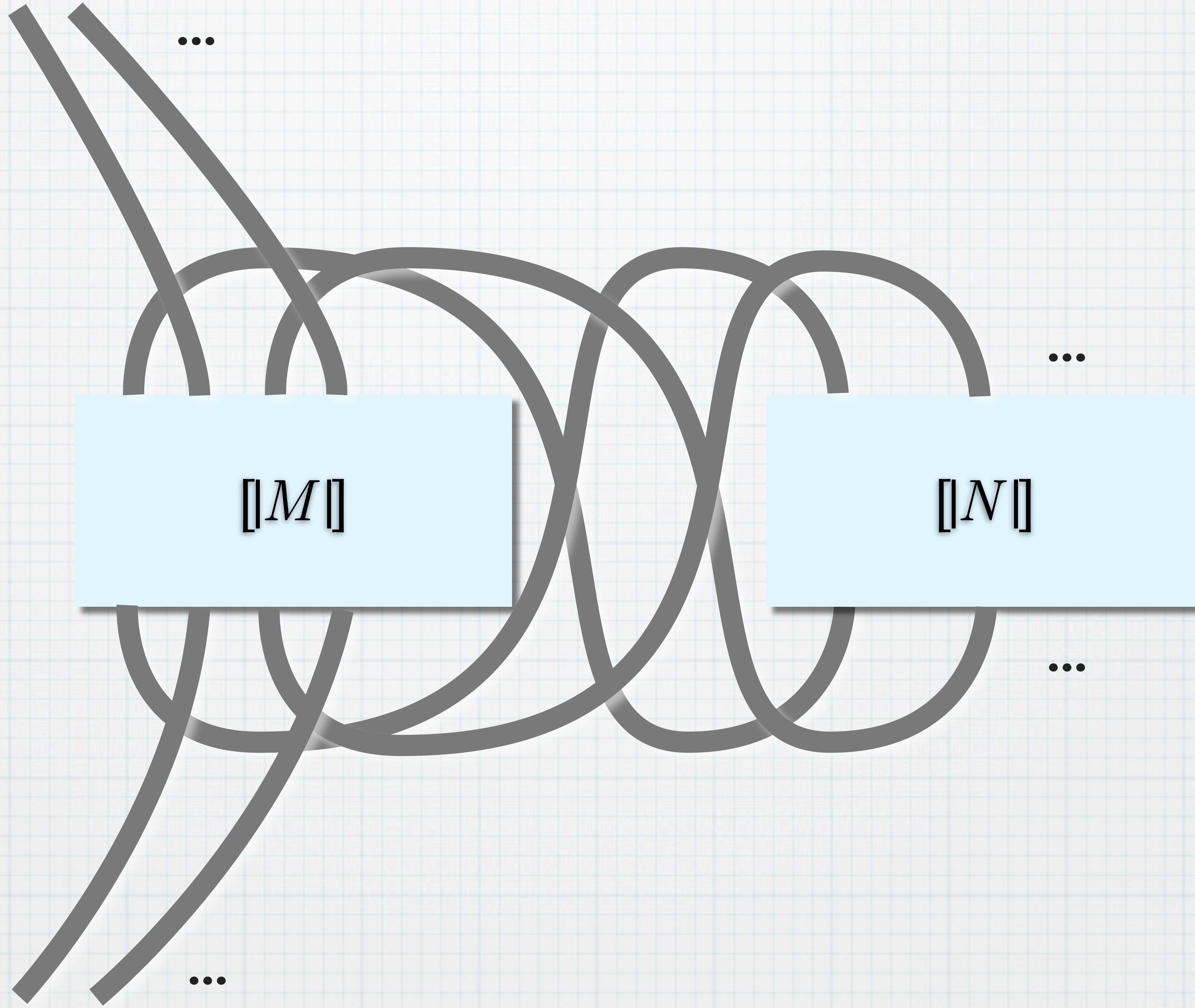


$[\![MN]\!]$

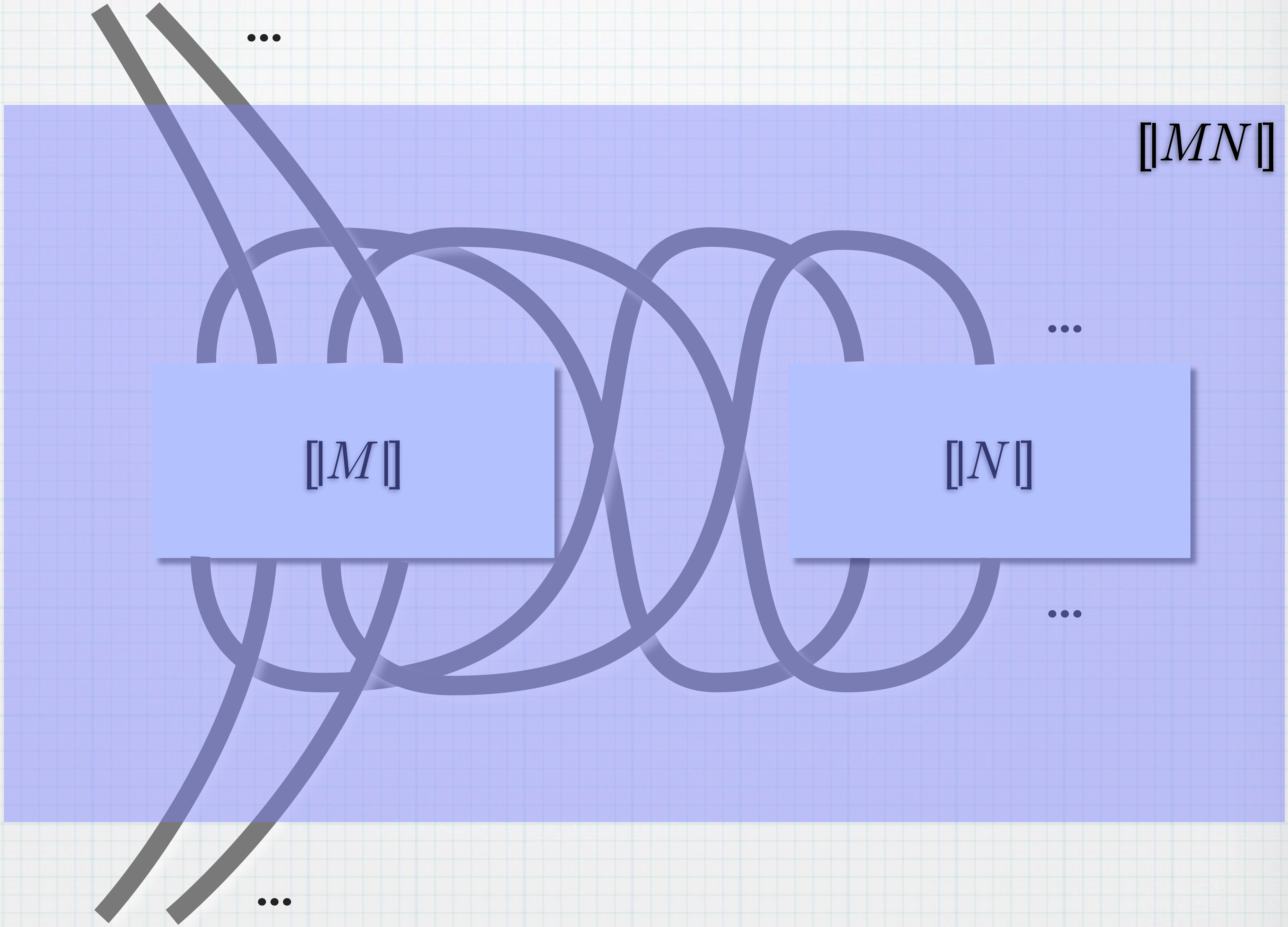
$=$



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

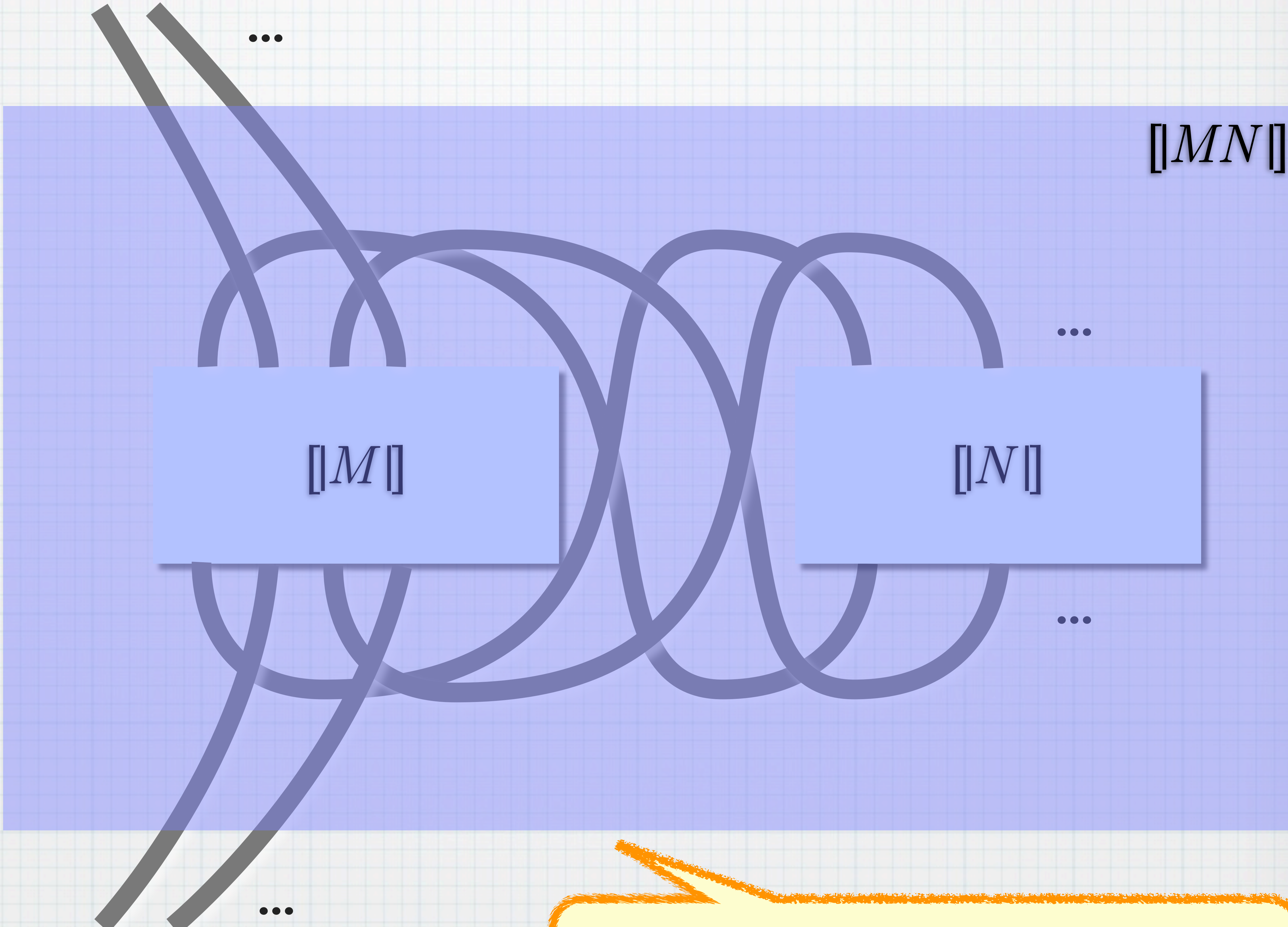


$$[\![\boldsymbol{MN}]\!] =$$



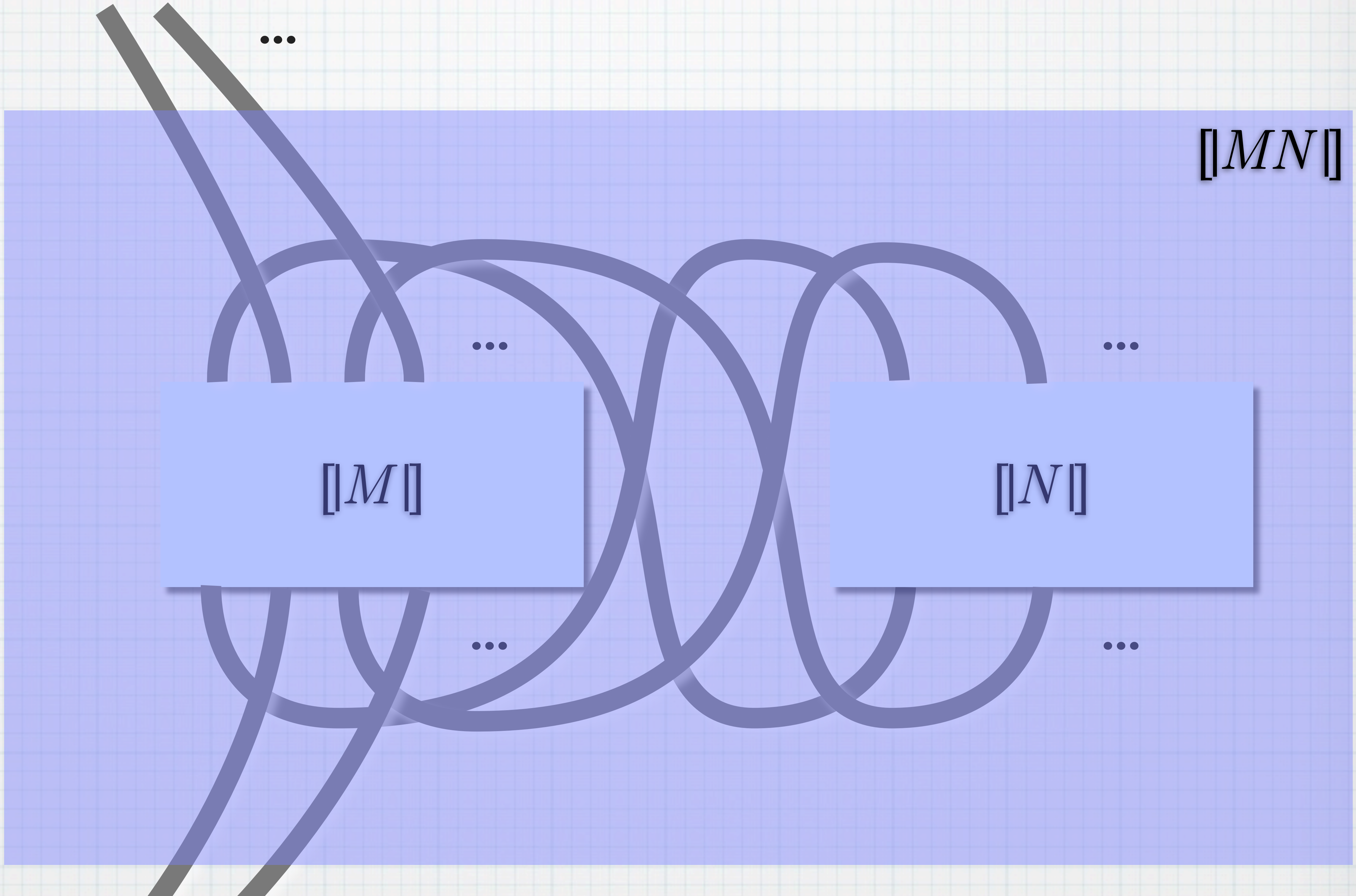
$[\![MN]\!]$

$=$



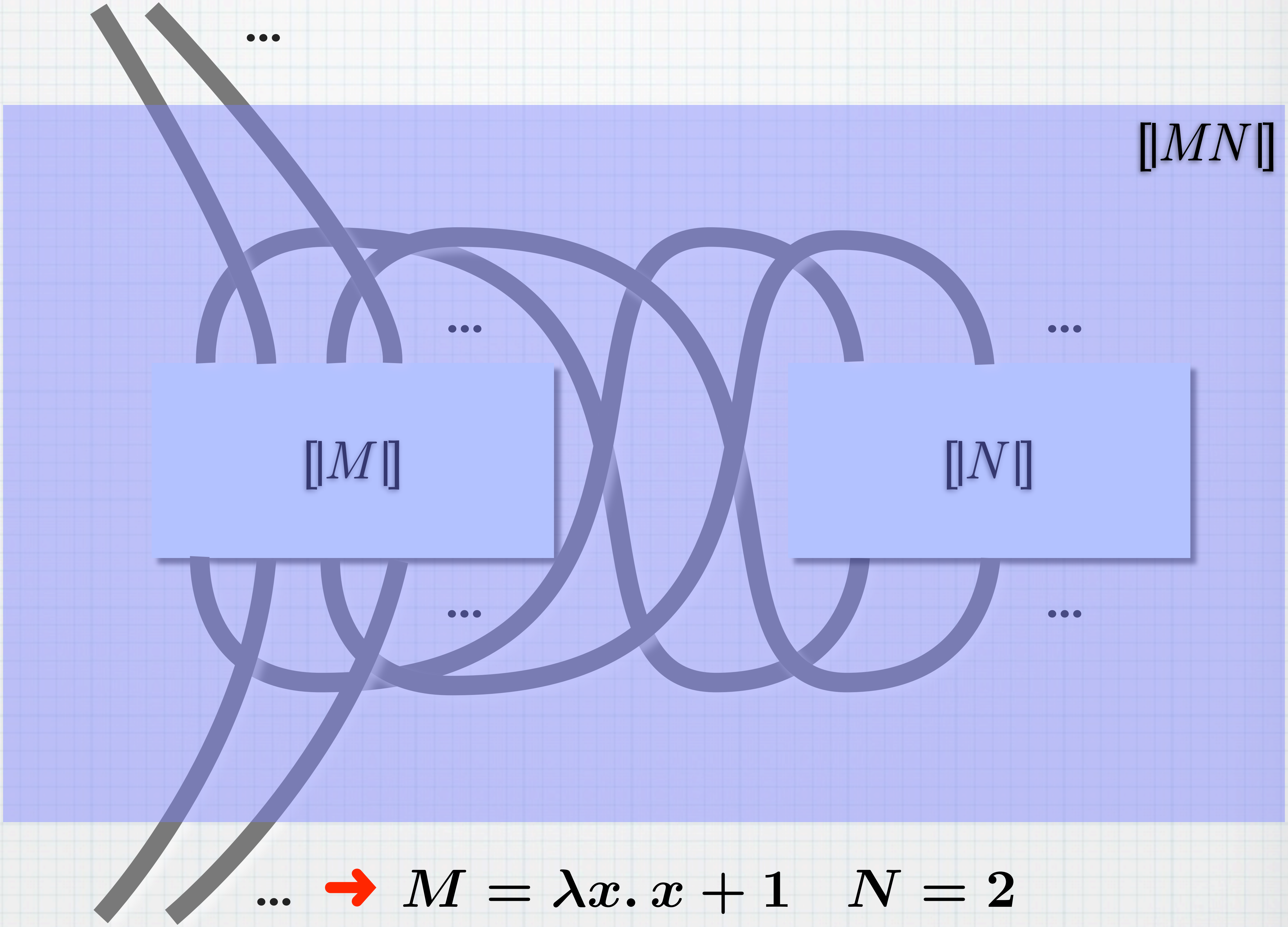
"parallel composition + hiding"
(cf. AJM games)

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



$$\boldsymbol{M = \lambda x.\, x + 1 \quad N = 2}$$
$$\boldsymbol{M = \lambda x.\, 1 \quad N = 2}$$
$$\boldsymbol{M = \lambda f.\, f 1 \quad N = \lambda x.\, (x + 1)}$$

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

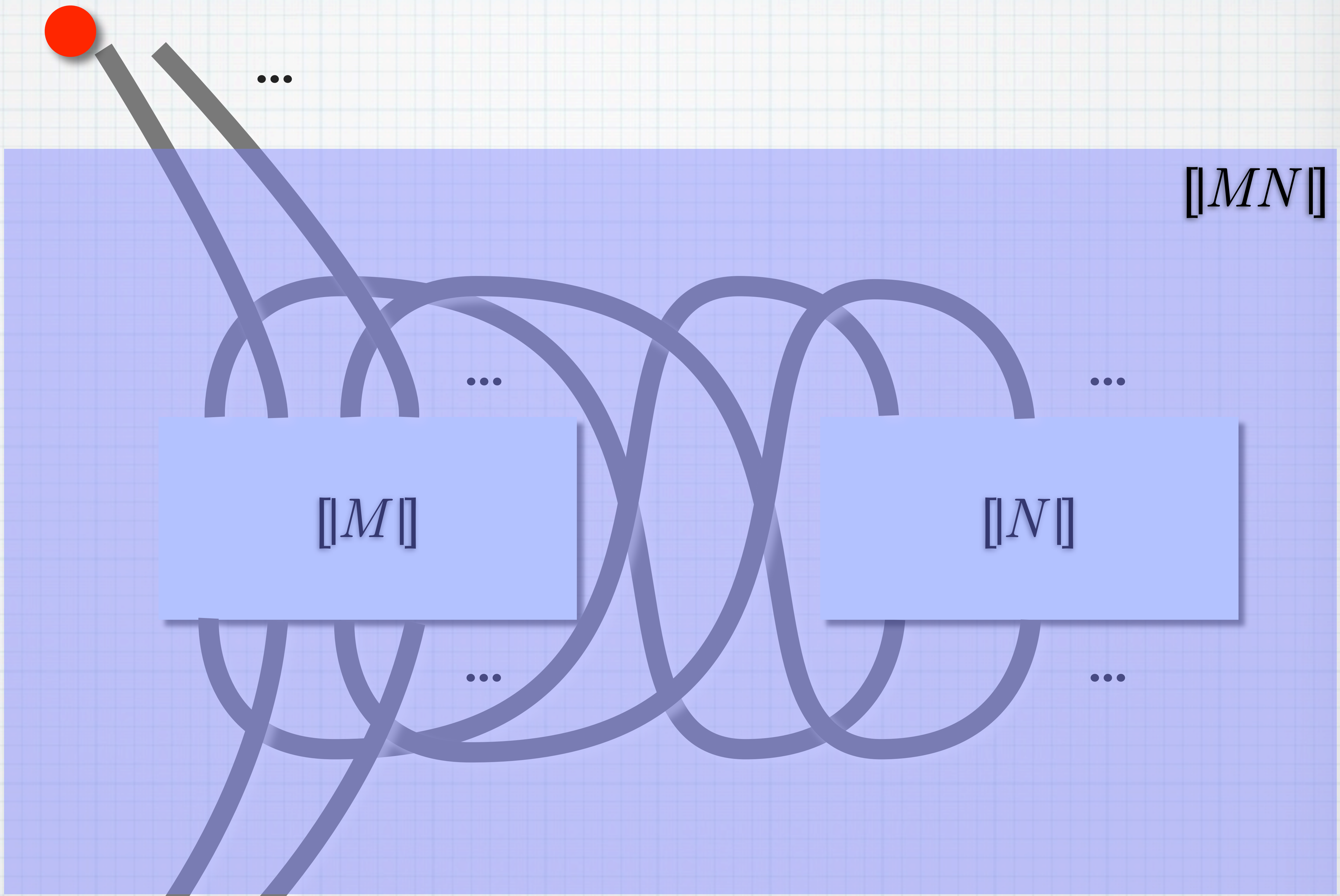


... ➜ $M = \lambda x.\, x + 1 \quad N = 2$

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

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

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



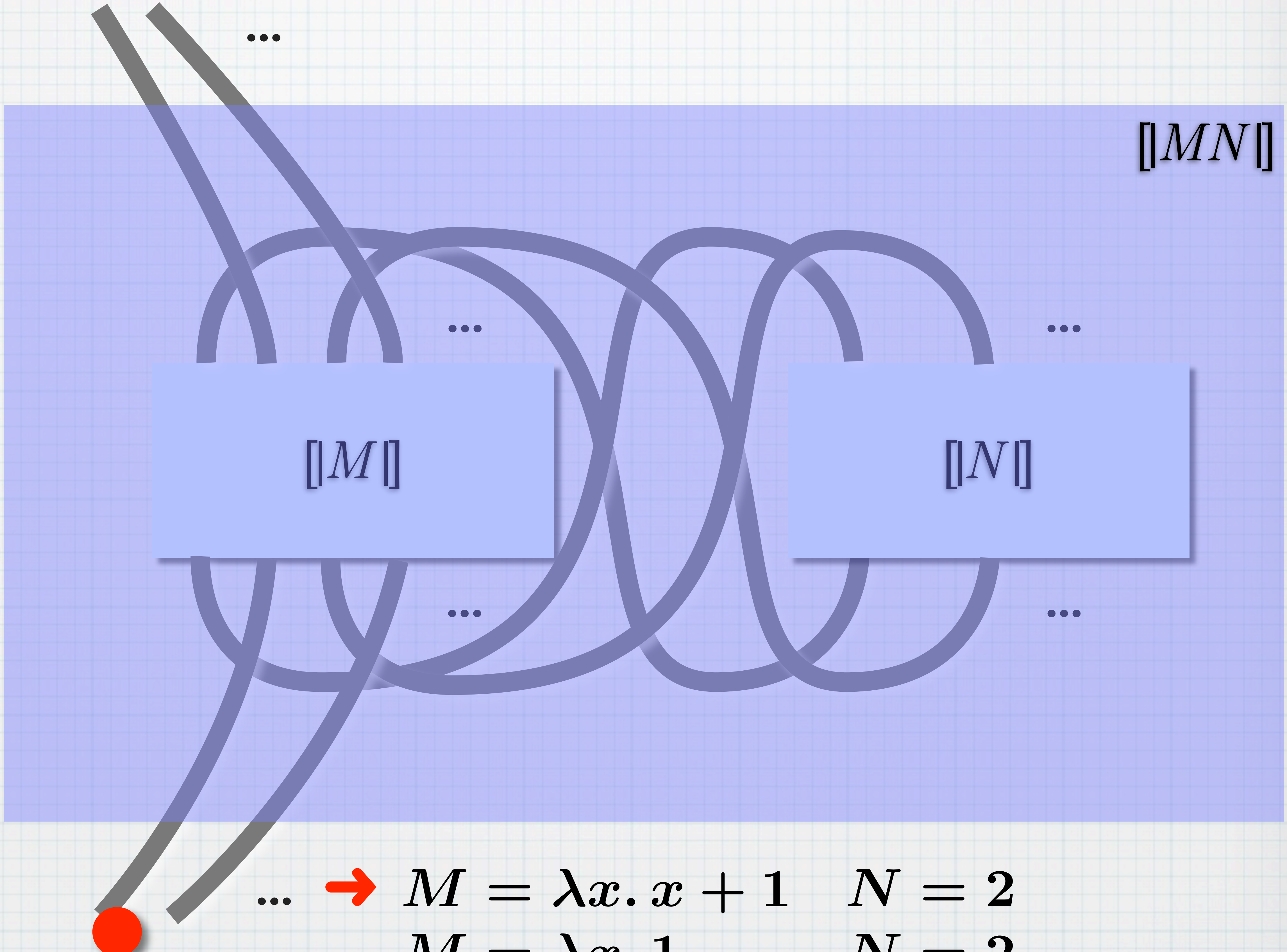
$M = \lambda x.\, x + 1 \qquad N = 2$

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

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

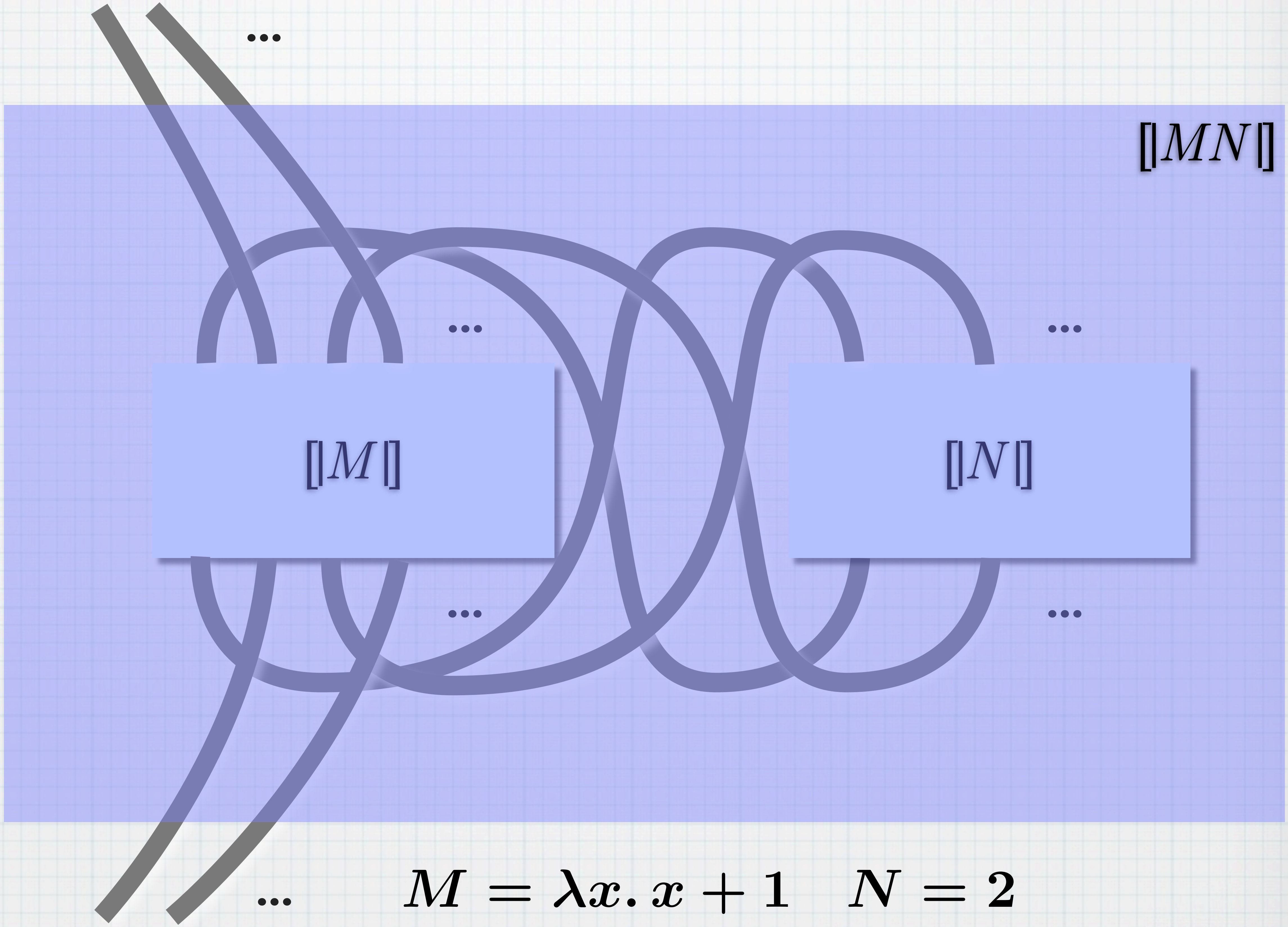
$[\![M N]\!]$

$=$



$$\to \; M = \lambda x.\, x + 1 \qquad N = 2$$
$$M = \lambda x.\, 1 \qquad N = 2$$
$$M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)$$

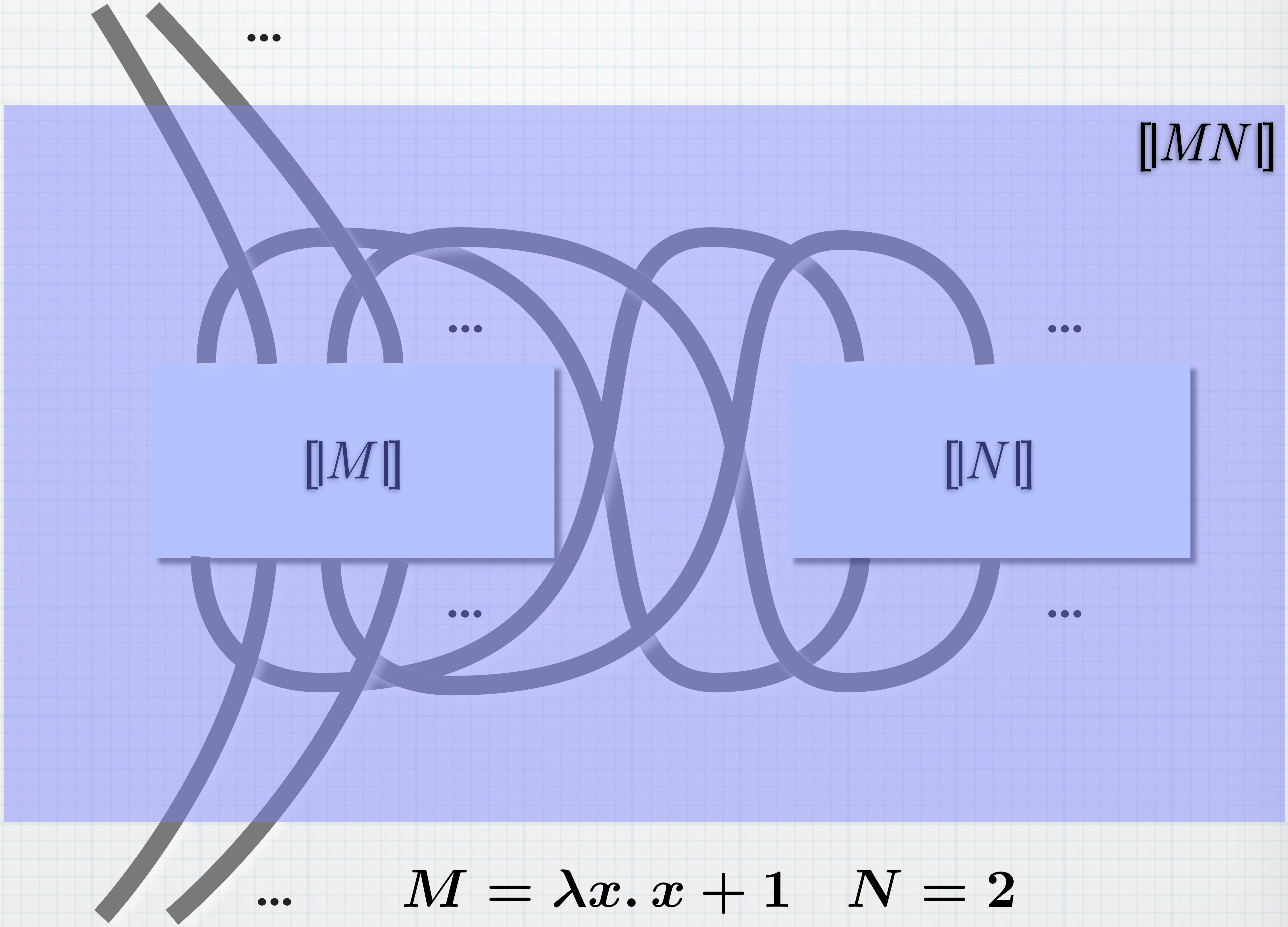
$$[\![MN]\!] =$$



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

$[\![MN]\!]$

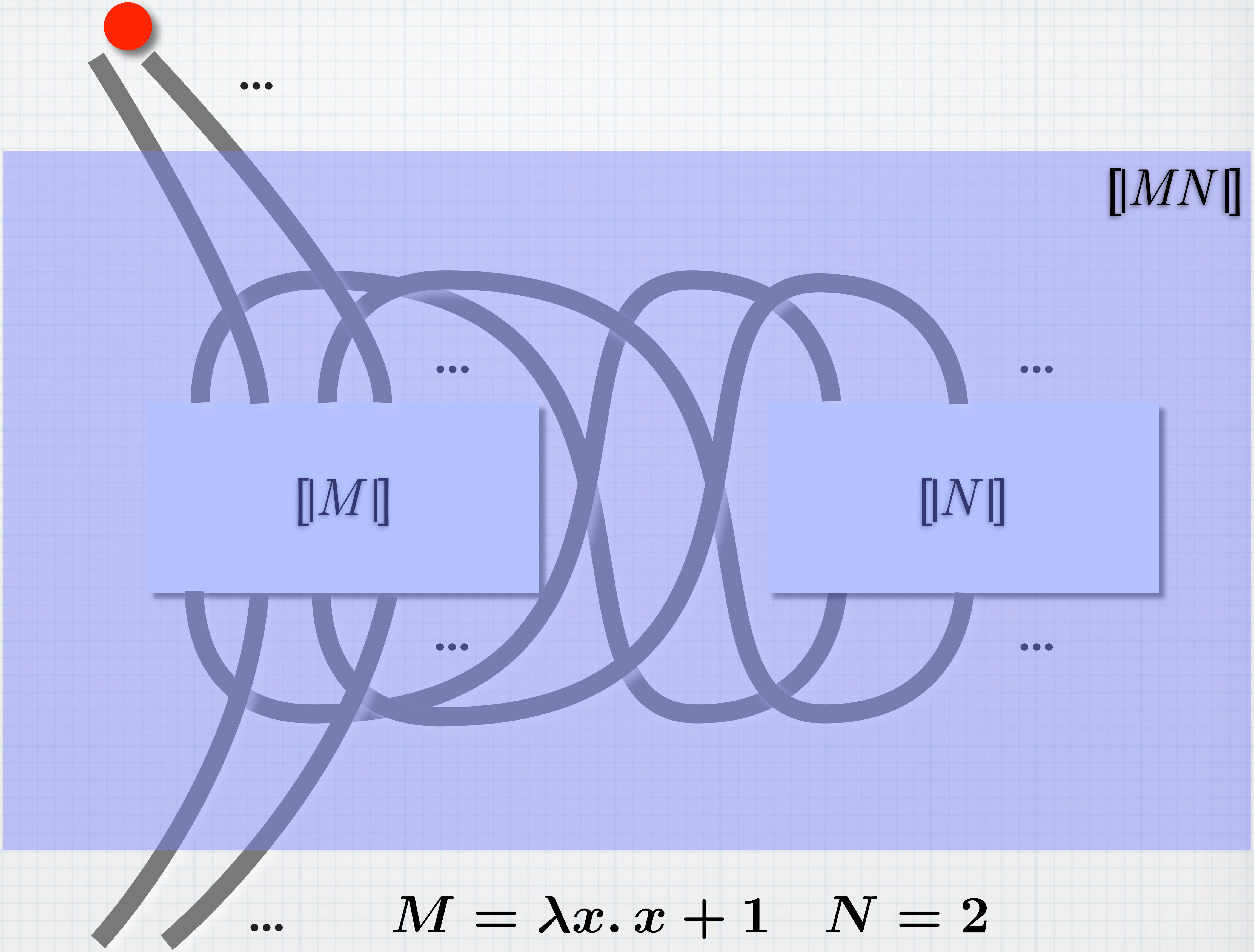
$=$



$$M = \lambda x.\, x + 1 \qquad N = 2$$
$$\to \quad M = \lambda x.\, 1 \qquad N = 2$$
$$M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)$$

$[\![MN]\!]$

$=$



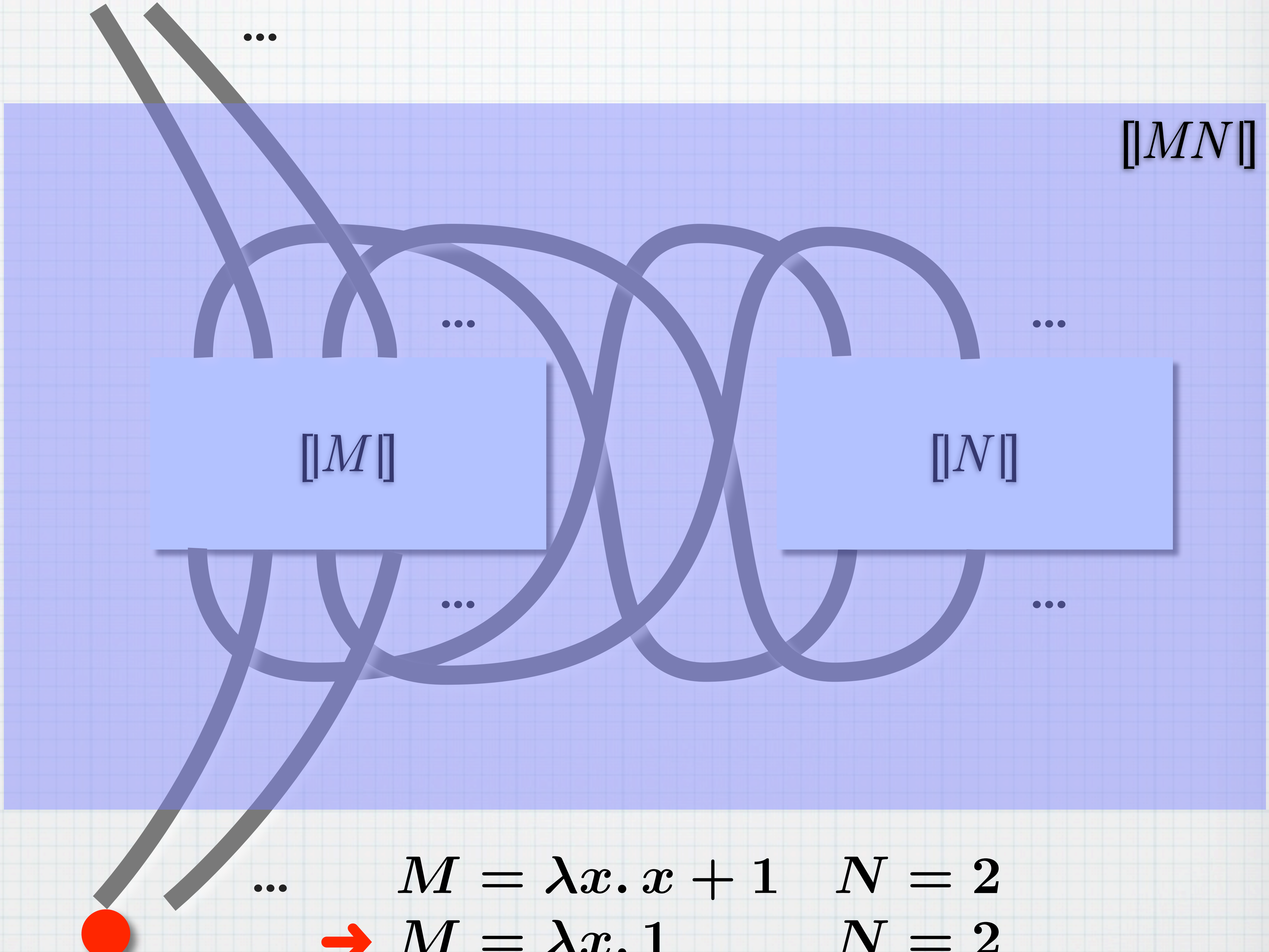
$$M = \lambda x.\, x + 1 \qquad N = 2$$

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

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

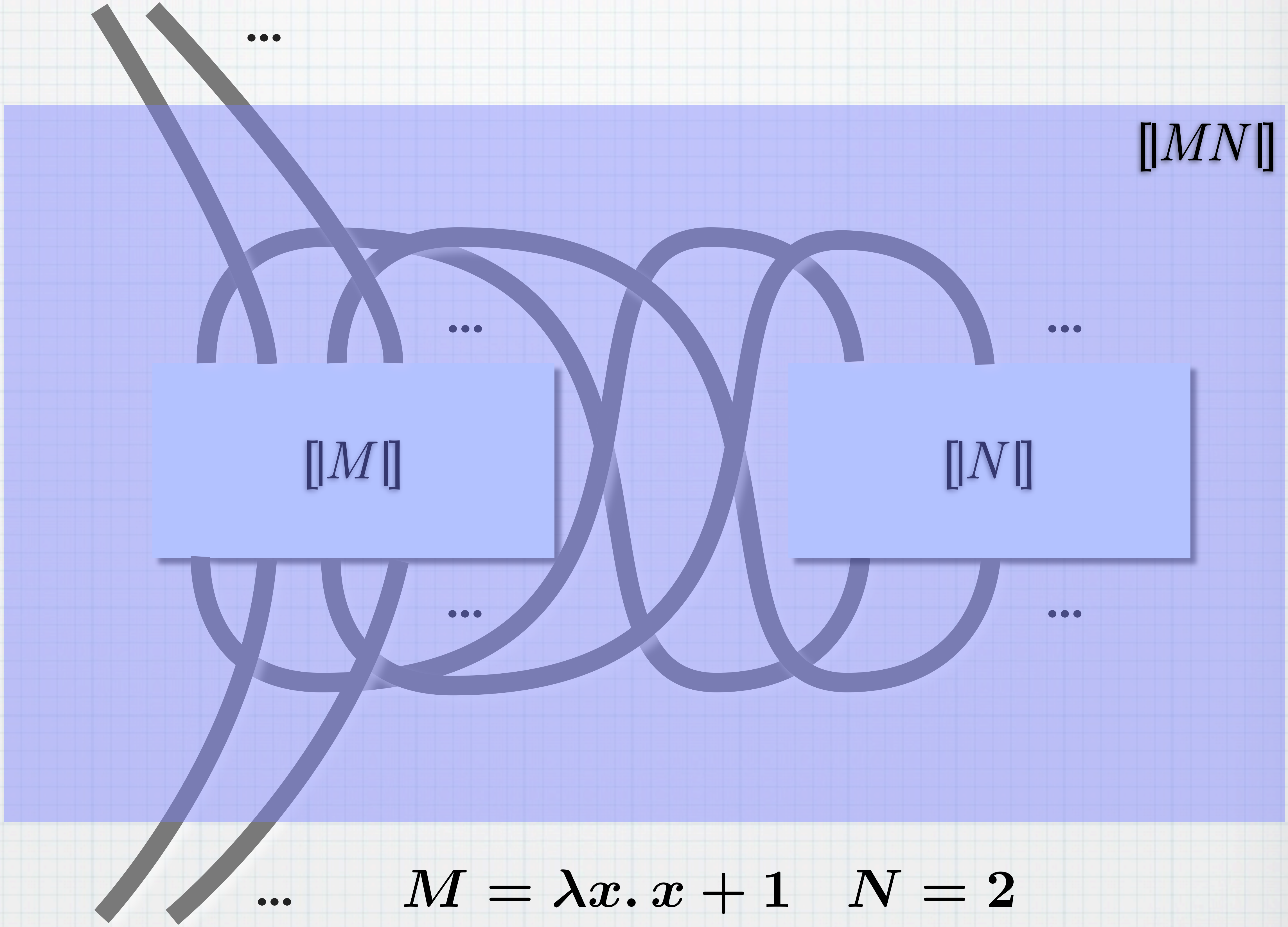
$[\![MN]\!]$

$=$



$$M = \lambda x.\, x + 1 \qquad N = 2$$
$$\rightarrow M = \lambda x.\, 1 \qquad N = 2$$
$$M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)$$

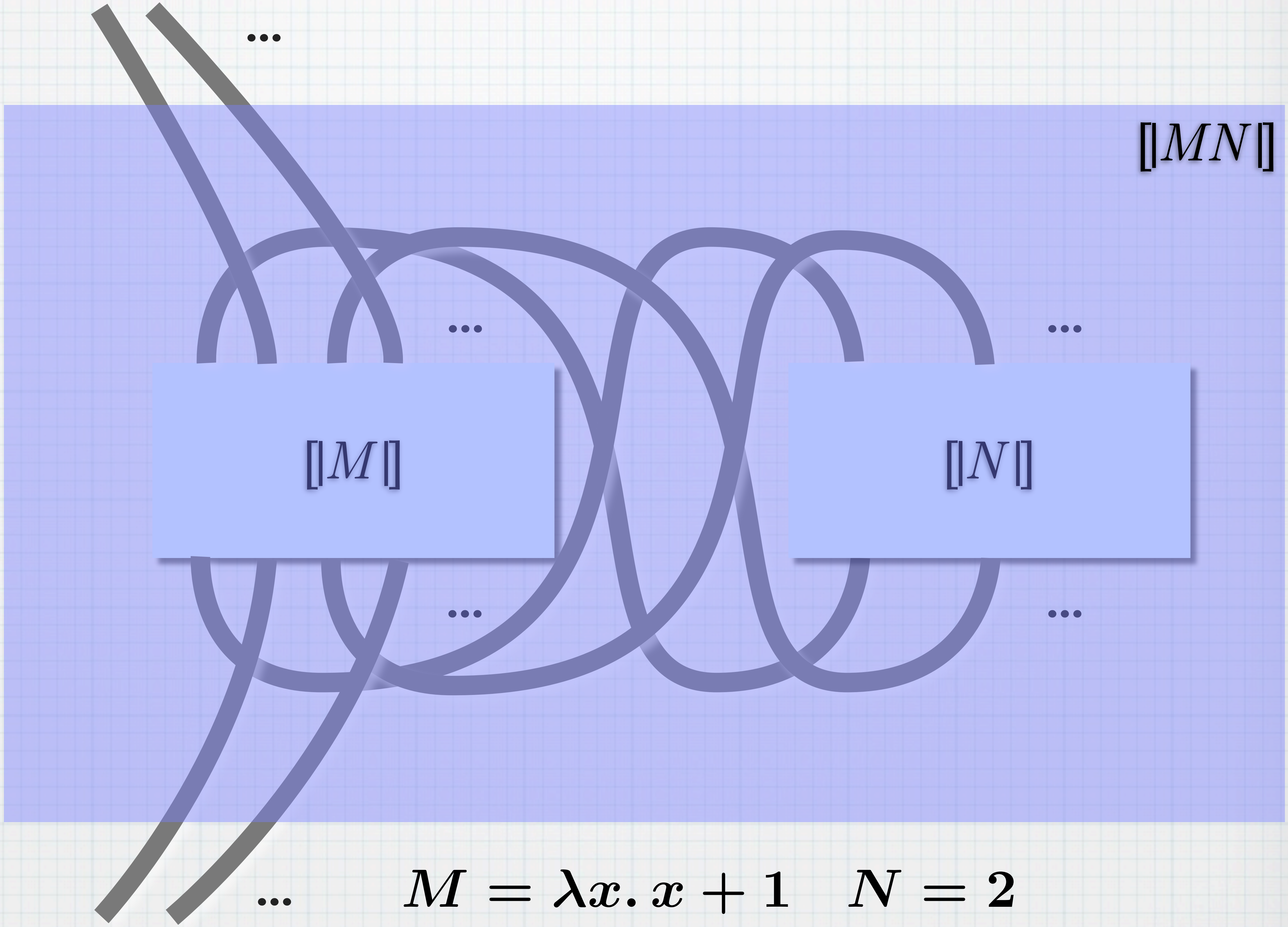
$[\![MN]\!]$
$=$



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

$[\![MN]\!]$

$=$

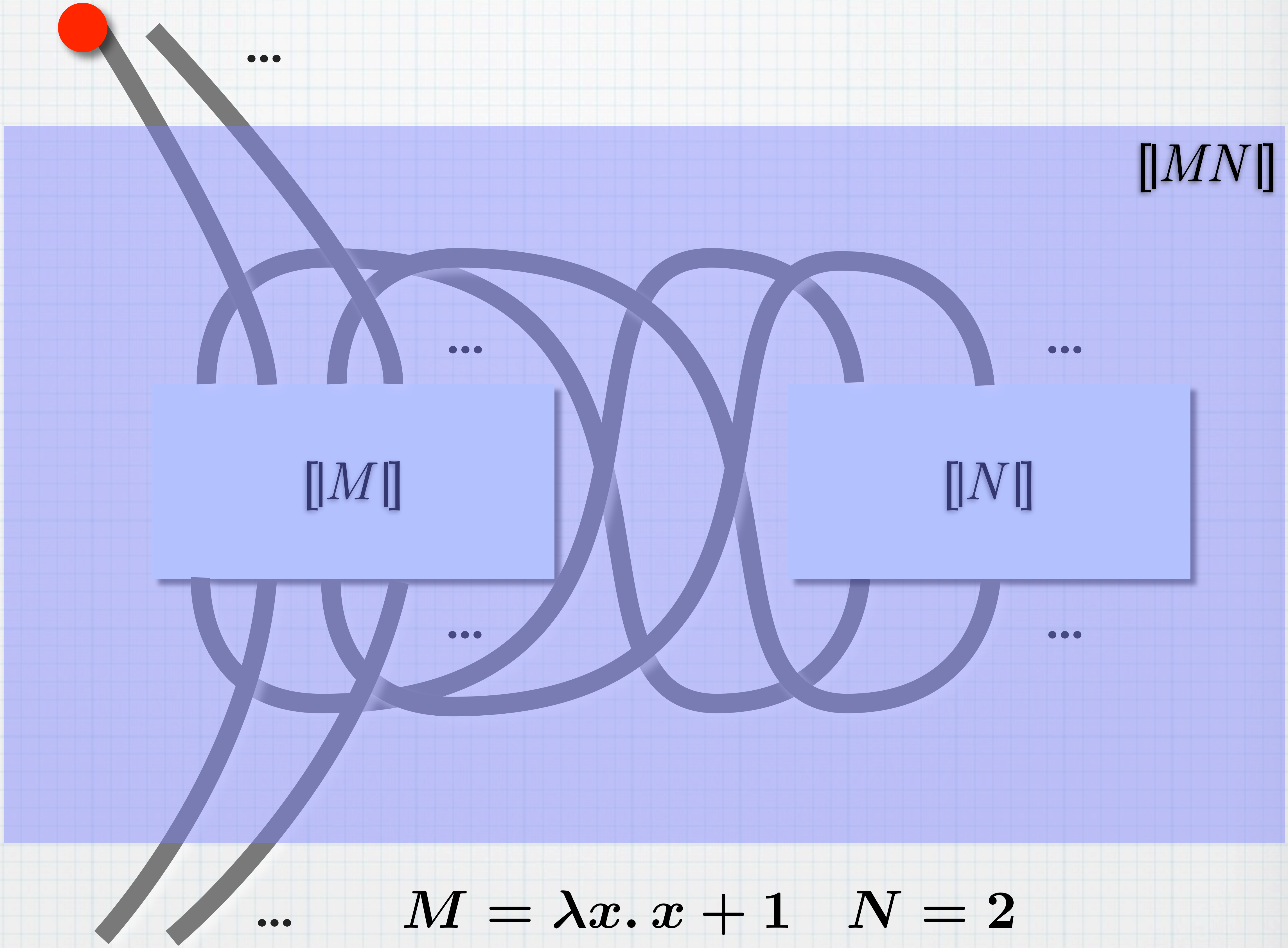


$$M = \lambda x.\, x + 1 \qquad N = 2$$

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

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

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

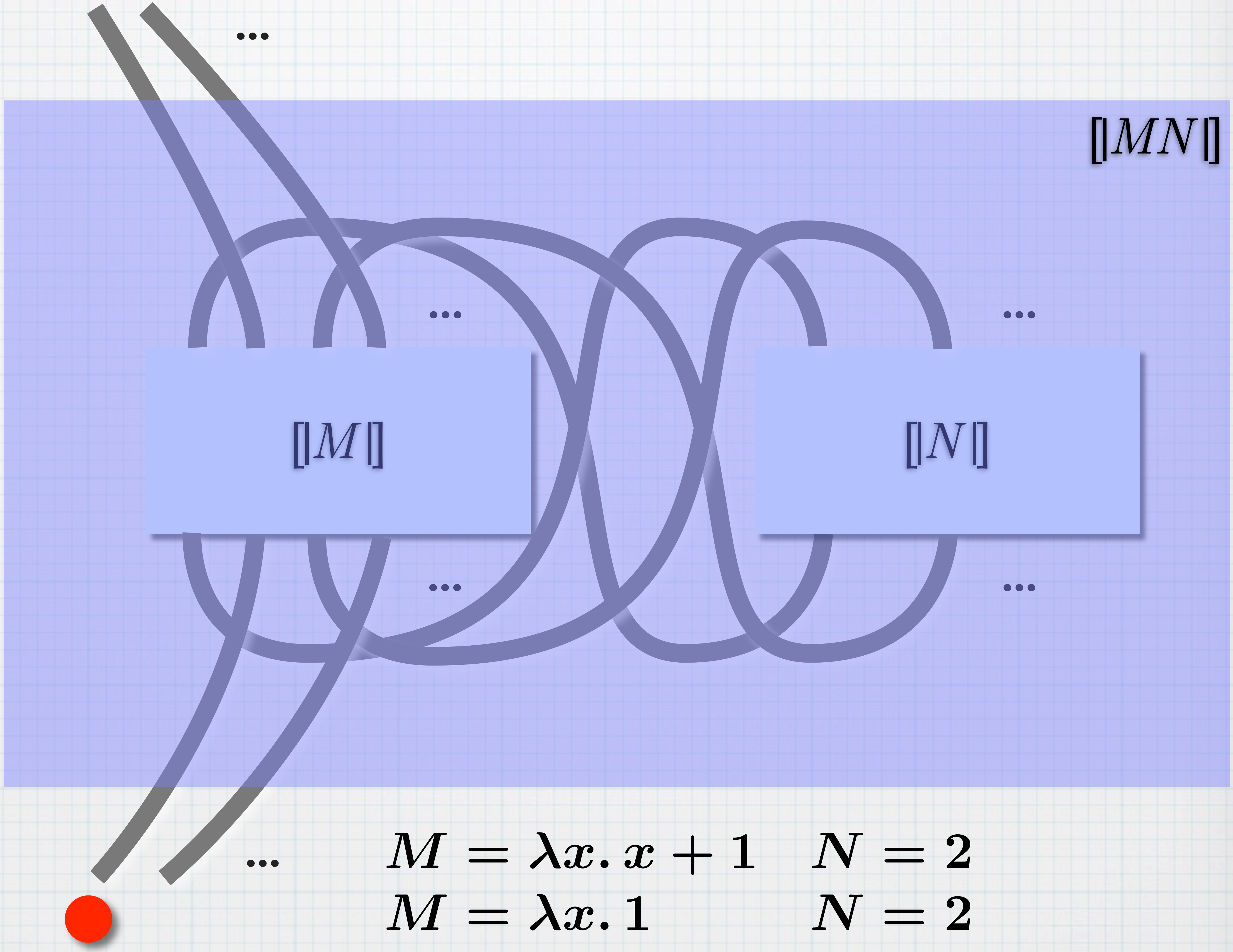


$$\boldsymbol{M = \lambda x.\, x + 1} \qquad \boldsymbol{N = 2}$$

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

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

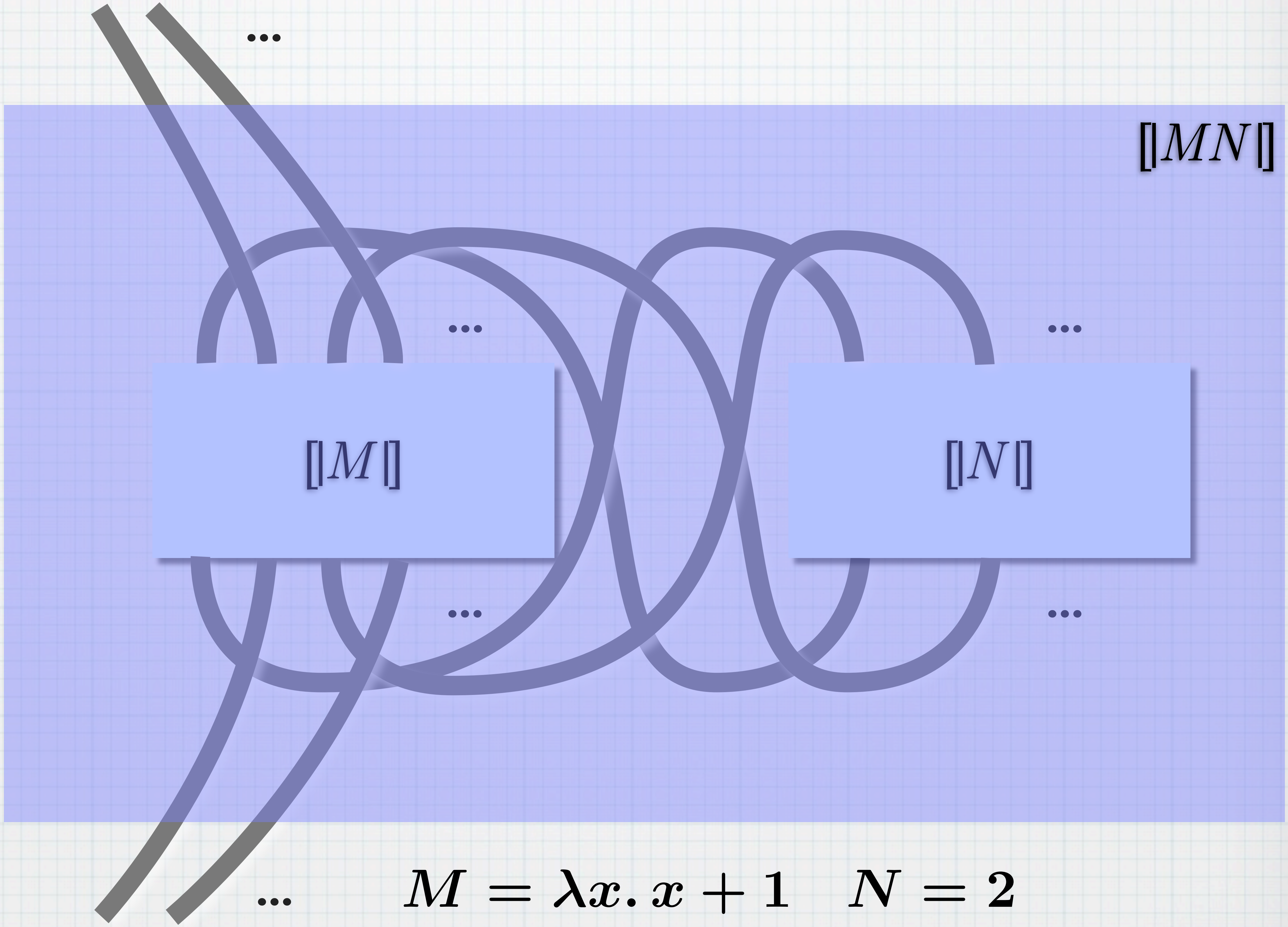
$$[\![MN]\!] =$$



$$M = \lambda x.\, x + 1 \qquad N = 2$$
$$M = \lambda x.\, 1 \qquad N = 2$$
$$\rightarrow M = \lambda f.\, f1 \qquad N = \lambda x.\, (x+1)$$

$[\![MN]\!]$

$=$

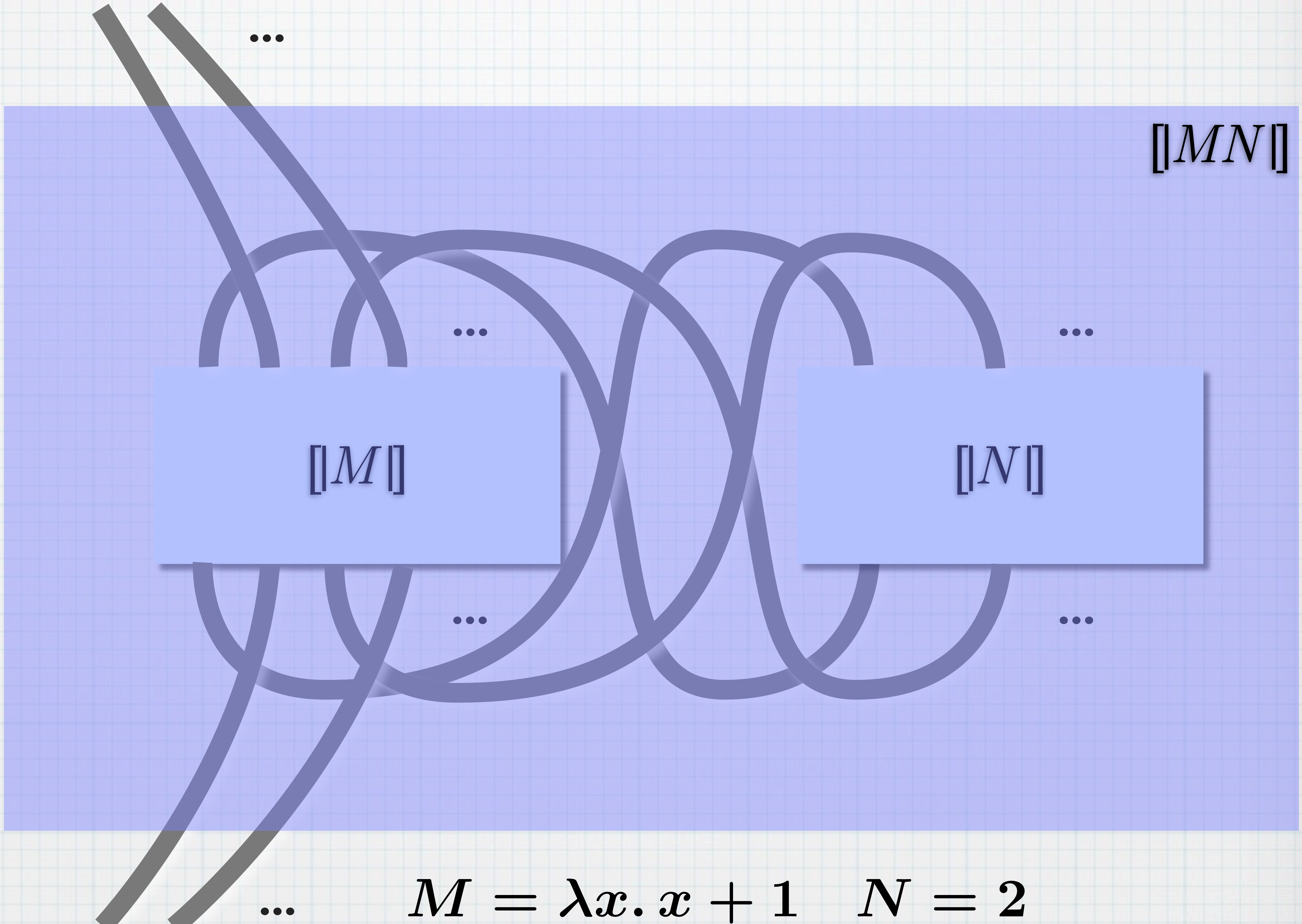


$M = \lambda x.\, x+1 \quad N = 2$

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

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

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

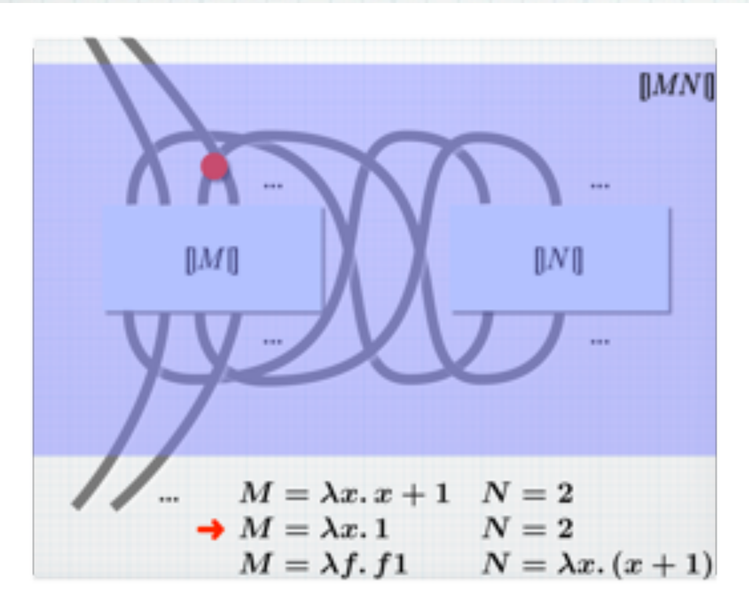


$$\boldsymbol{M = \lambda x.\, x + 1 \qquad N = 2}$$
$$\boldsymbol{M = \lambda x.\, 1 \qquad N = 2}$$
$$\boldsymbol{M = \lambda f.\, f 1 \qquad N = \lambda x.\, (x + 1)}$$

Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

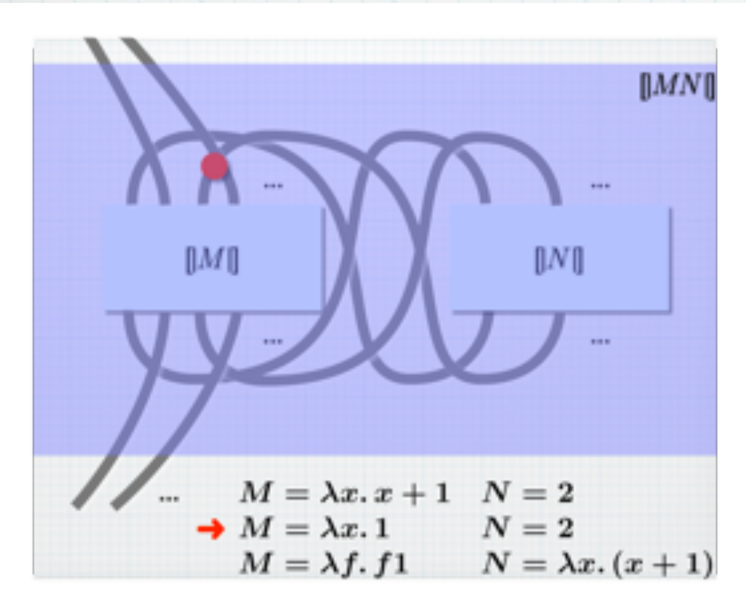
"GoI Animation"

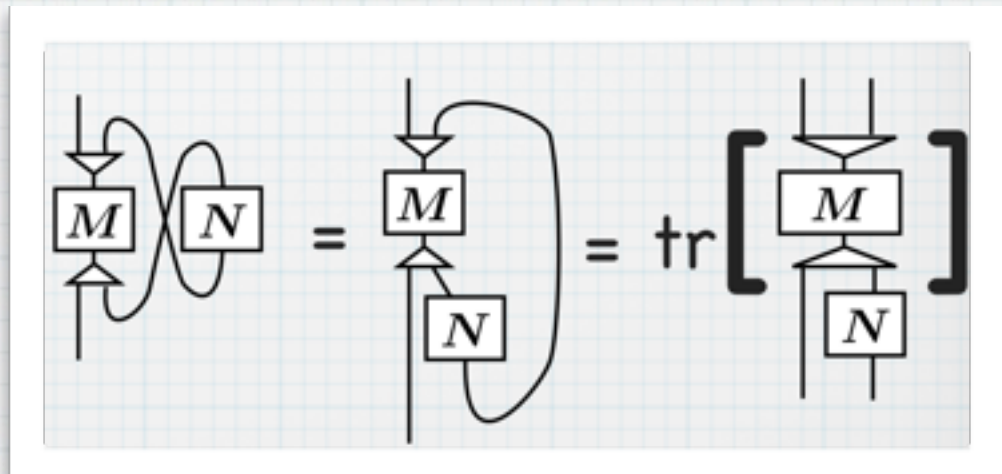


Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

"GoI Animation"





Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]

Hasuo (Tokyo)

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**, Math. Str. Comp. Sci, 2002
  * Especially its technical report version (Oxford CL), since it's a bit more detailed
* See also:
  * IH and Naohiko Hoshino. **Semantics of Higher-Order Quantum Computation via Geometry of Interaction.** Extended ver. of [LICS'11], to appear in Annals Pure & Applied Logic. arxiv.org/abs/1605.05079

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]

Categorical GoI [AHS02]

Linear combinatory algebra

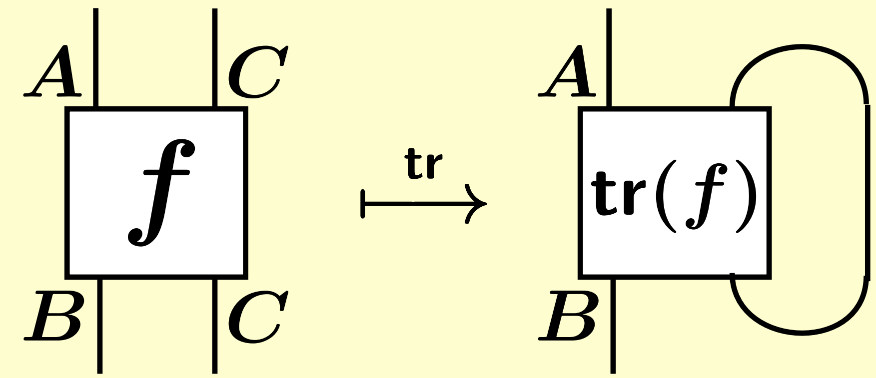
Realizability

Linear category

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]



Categorical GoI [AHS02]

Linear combinatory algebra

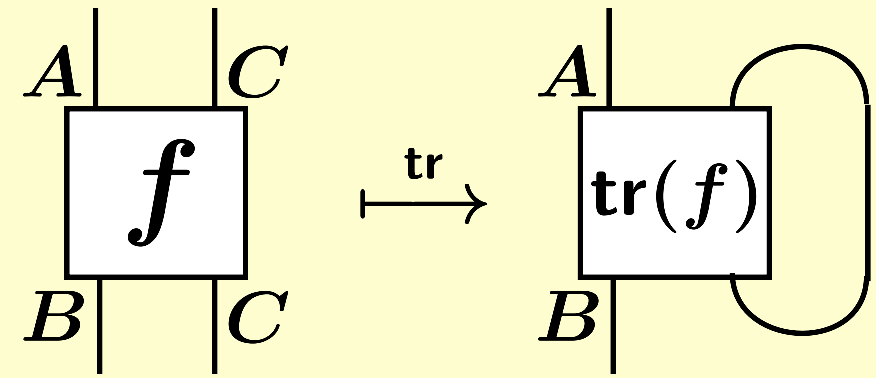
Realizability

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

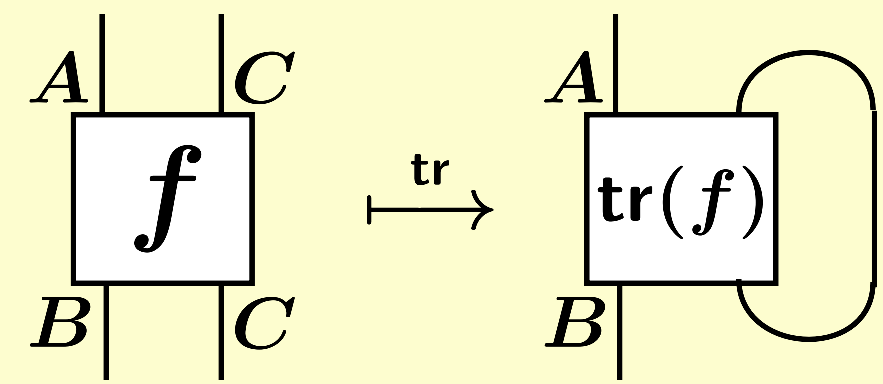
Realizability

Linear category

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]



⬇ Categorical GoI [AHS02]

* Applicative str. + combinators
* Model of untyped calculus

Linear combinatory algebra

⬇ Realizability

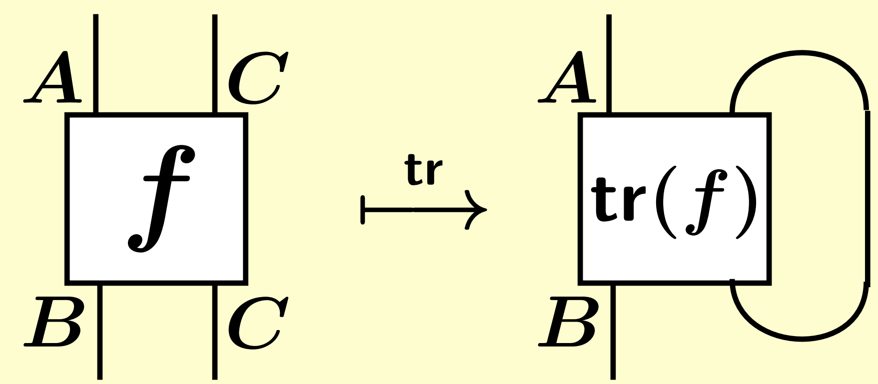
Linear category

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"

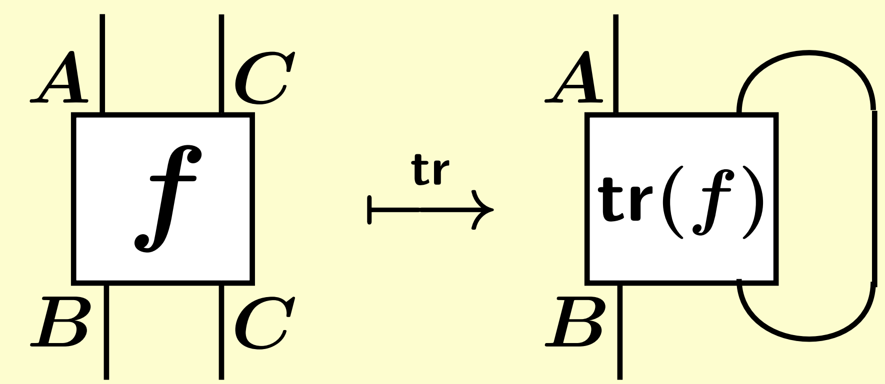
Linear category

Model of typed calculus

The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]



Categorical GoI [AHS02]

Linear combinatory algebra

* Applicative str. + combinators
* Model of untyped calculus

Realizability

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

Linear category

Model of typed calculus

Hasuo (Tokyo)

Linear Combinatory Algebra (LCA)

Defn. (LCA)
A *linear combinatory algebra (LCA)* is a set A equipped with

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

$$\cdot \;:\; A^2 \longrightarrow A$$

- a unary operator

$$! \;:\; A \longrightarrow A$$

- (*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.

Hasuo (Tokyo)

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.

Hasuo (Tokyo)

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.

* Model of
untyped linear λ

Hasuo (Tokyo)

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

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

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

* Model of
untyped linear λ

* $a \in A \quad \approx$
closed linear λ-term

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.

* Model of untyped linear λ

* $a \in A$ $\approx$ closed linear λ-term

* No **S** or **K** (linear!)

* Combinatory completeness: e.g.

$$\lambda xyz.\, zxy$$

designates an elem. of A

Hasuo (Tokyo)

What we use (ingredient)

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).

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

  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.

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

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.

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

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

GoI situation

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

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

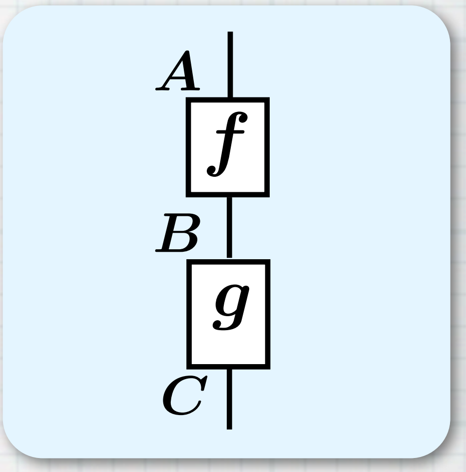
  Here K_I is the constant functor into the monoidal unit I;
- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

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

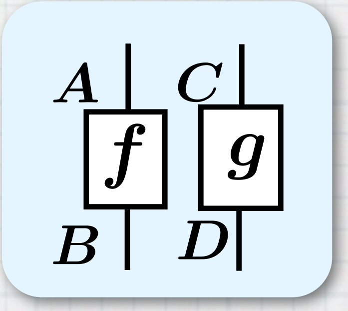
* Monoidal category $(\mathbb{C}, \otimes, I)$

* String diagrams

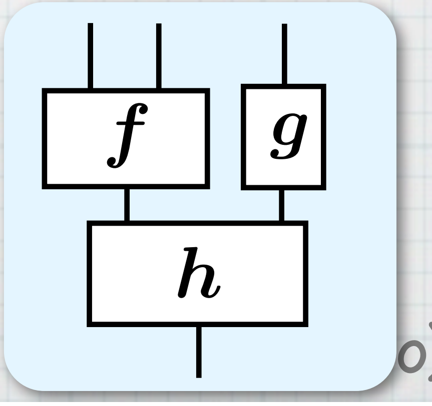
$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{g \circ f} C}$$



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



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



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).

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

  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.

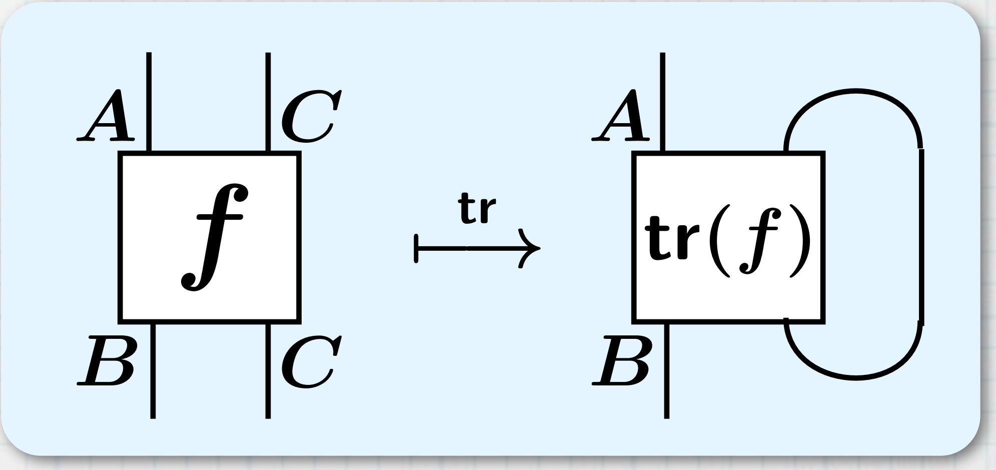
$$\begin{array}{c} j : U \otimes U \lhd U : k \\ I \lhd U \\ u : FU \lhd U : v \end{array}$$

* **Traced** monoidal category

* "feedback"

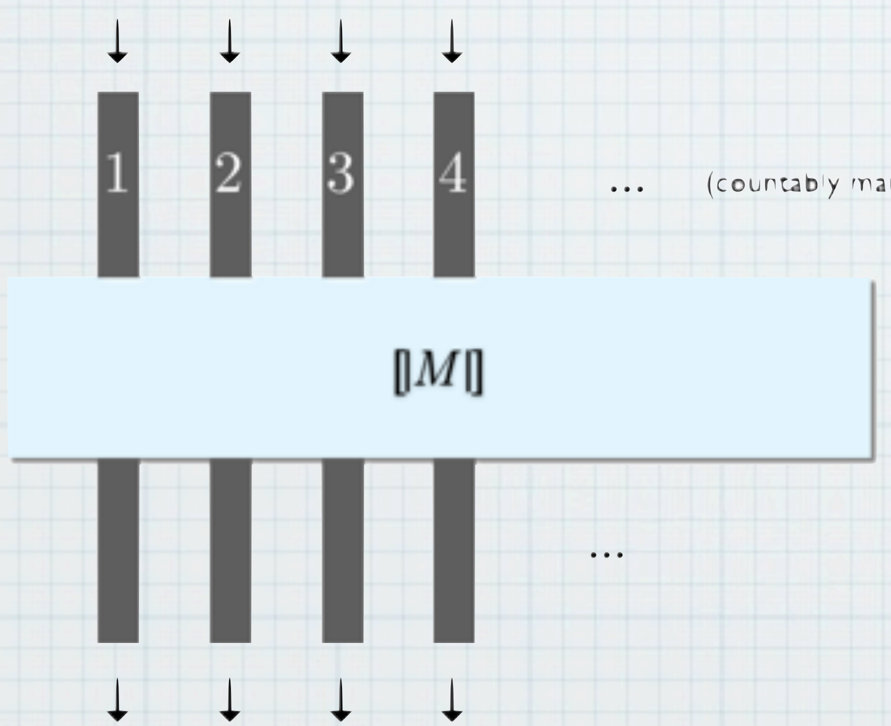
$$\frac{A \otimes C \xrightarrow{f} B \otimes C}{A \xrightarrow{\mathrm{tr}(f)} B}$$

that is

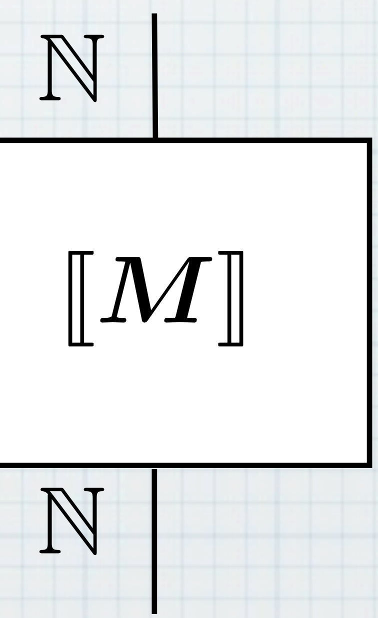


String Diagram vs. "Pipe Diagram"

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



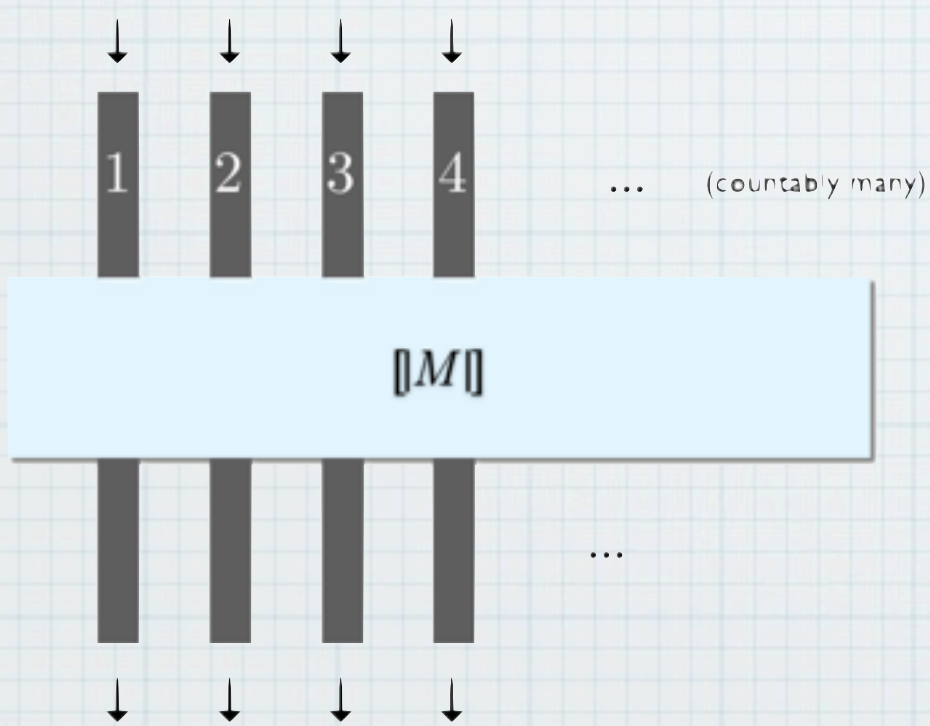
Pipe diagram



String diagram

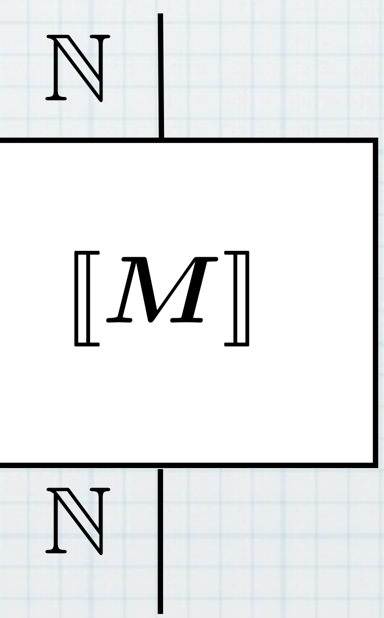
String Diagram vs. "Pipe Diagram"

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



Pipe diagram

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

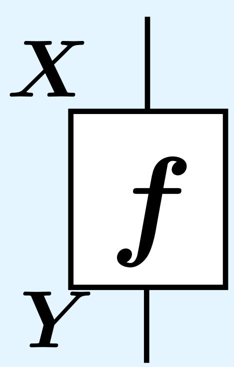


String diagram

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

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

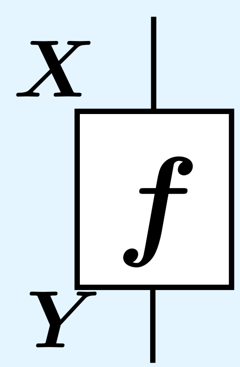
$$\frac{X \to Y \ \text{ in } \mathbf{Pfn}}{X \rightharpoonup Y, \ \text{partial function}}$$



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

* Category $\mathbf{Pfn}$ of **partial functions**
  * __Obj.__ A set X
  * __Arr.__ A partial function

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



* is traced symmetric monoidal

Traced Sym. Monoidal Category

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

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

  How?

*

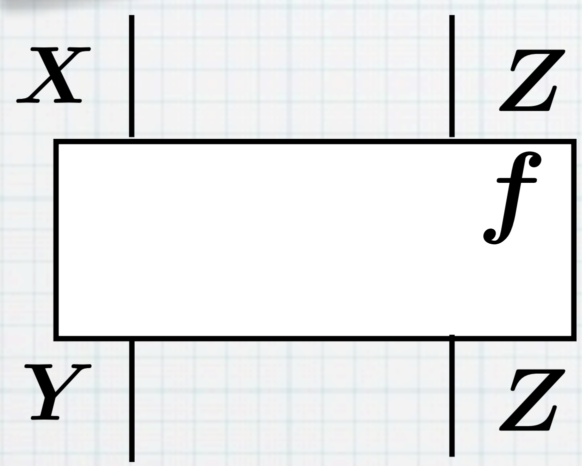
Hasuo (Tokyo)

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?

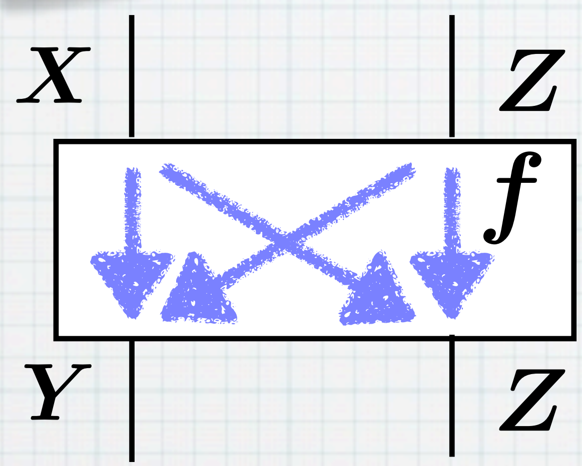
- 

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?

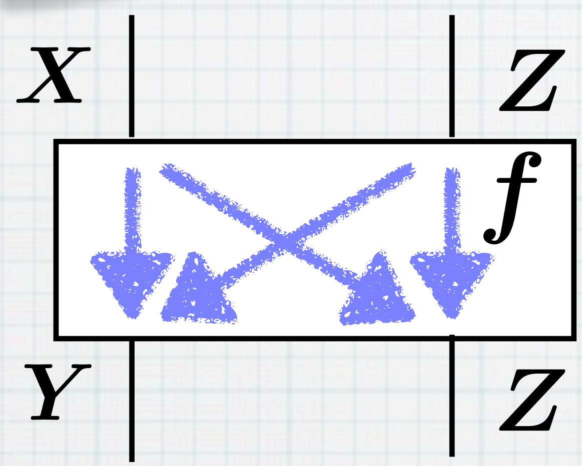
* 

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?

* 

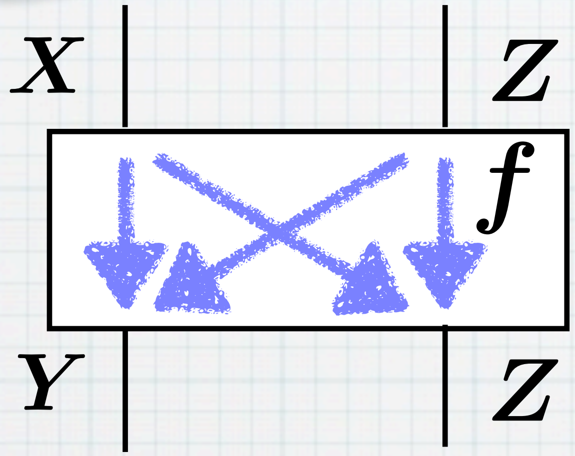
$$f_{XY}(x) := \begin{cases} f(x) & \text{if } f(x) \in Y \\ \perp & \text{o.w.} \end{cases}$$

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

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

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

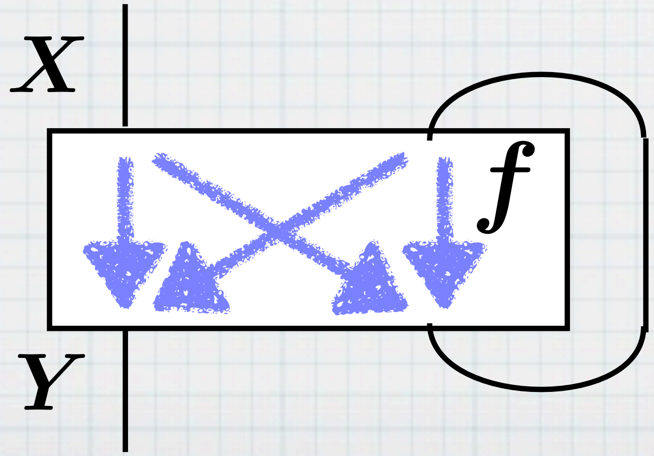
How?

* 

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

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

* Trace operator:

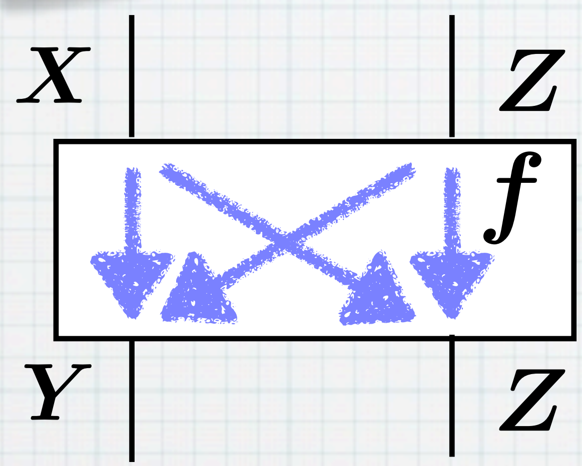


Traced Sym. Monoidal Category

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

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

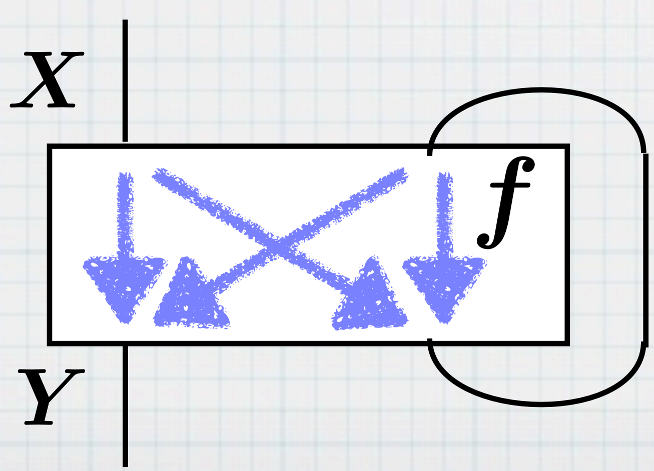
How?

* 

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

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

* Trace operator:



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

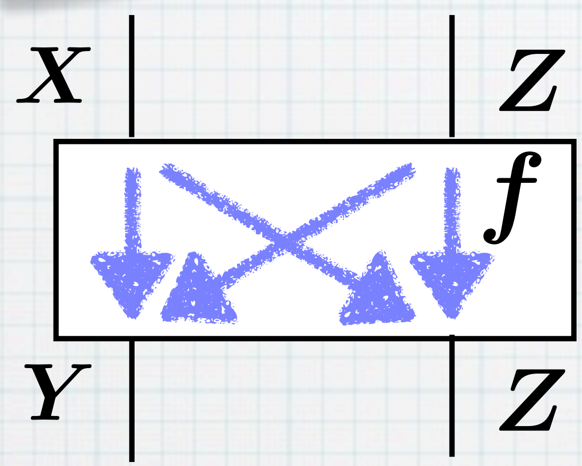
(Tokyo)

Traced Sym. Monoidal Category

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

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

How?

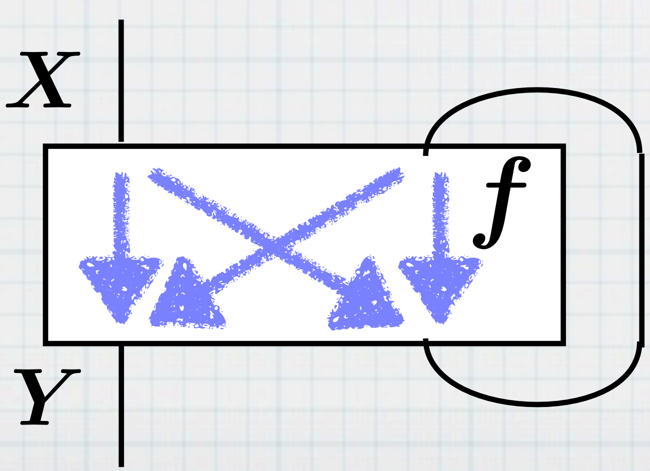
*


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

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

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

* Trace operator:



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

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

(Tokyo)

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).

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

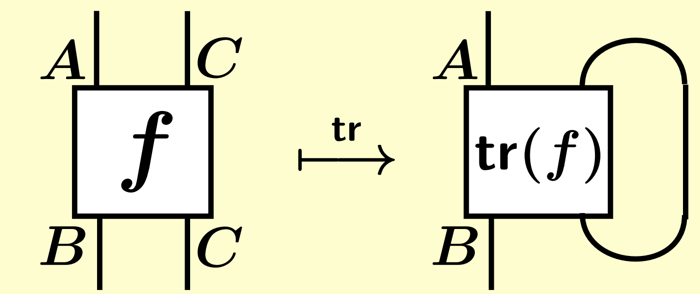
Comultiplication, Dereliction, Contraction, Weakening (respectively)

  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.

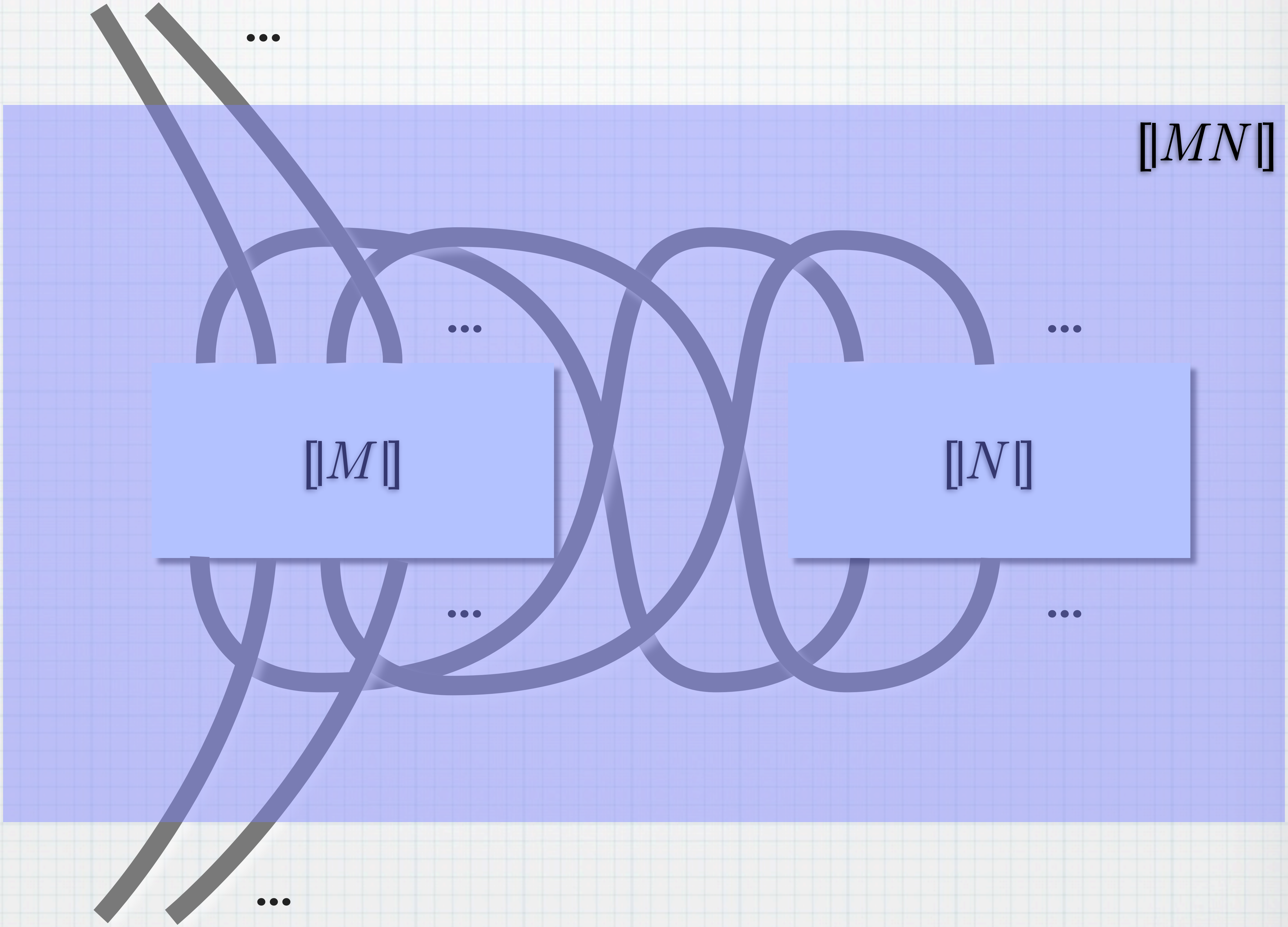
$$\begin{array}{rclcl} j & : & U \otimes U \lhd U & : & k \\ & & I \lhd U & & \\ u & : & FU \lhd U & : & v \end{array}$$

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

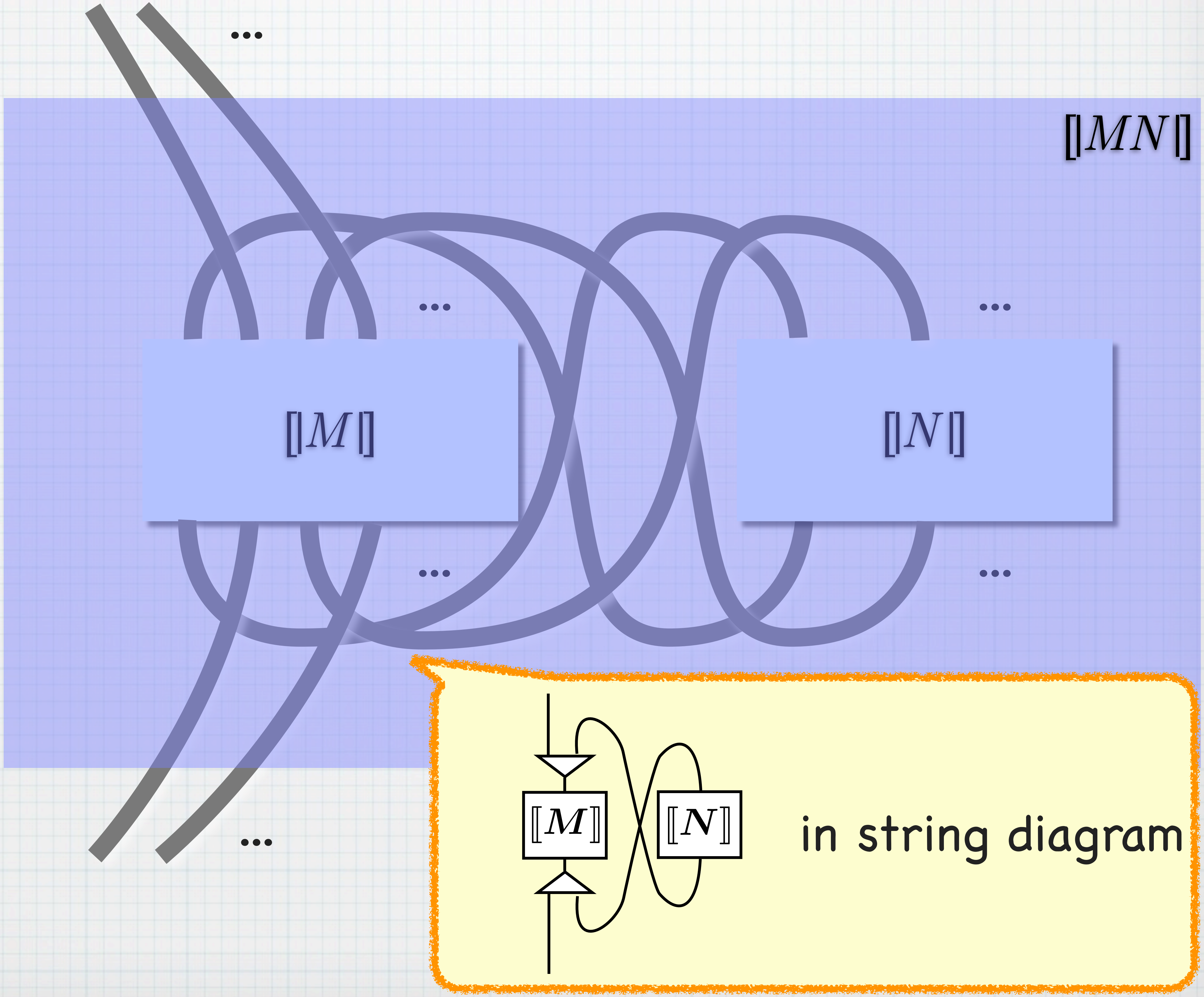


  * Why for GoI?

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



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



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).

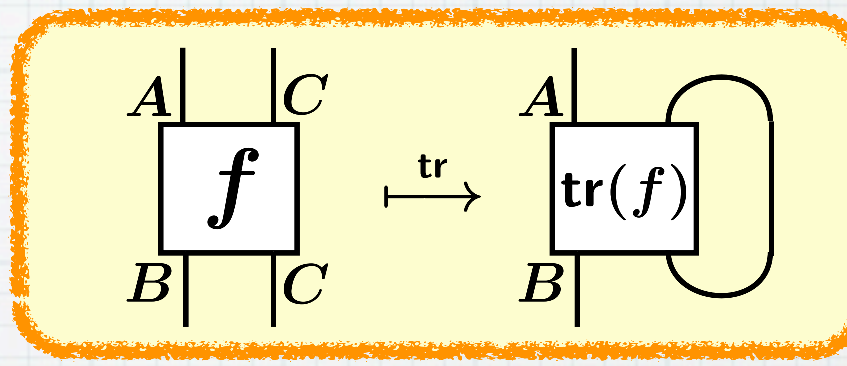
$$
\begin{array}{rcll}
e : \boldsymbol{FF} \lhd \boldsymbol{F} : e' & & & \text{Comultiplication} \\
d : \mathbf{id} \lhd \boldsymbol{F} : d' & & & \text{Dereliction} \\
c : \boldsymbol{F} \otimes \boldsymbol{F} \lhd \boldsymbol{F} : c' & & & \text{Contraction} \\
w : \boldsymbol{K_I} \lhd \boldsymbol{F}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

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

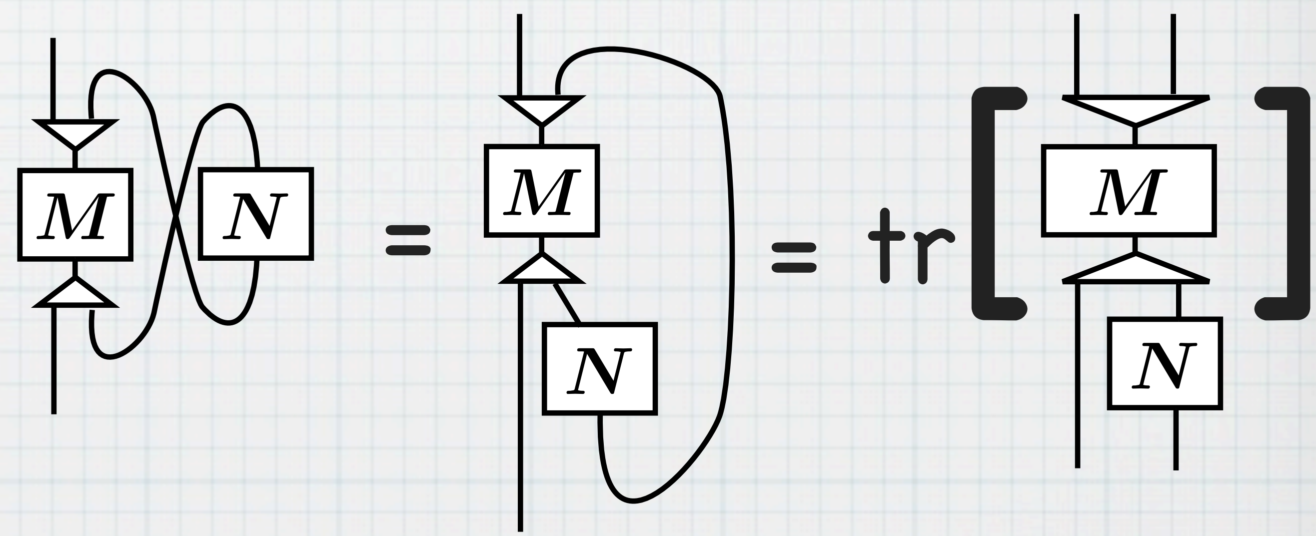
- $\boldsymbol{U} \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

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

- 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

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

"embedding"

"projection"

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

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

Hasuo (Tokyo)

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).

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

  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.

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

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

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).

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

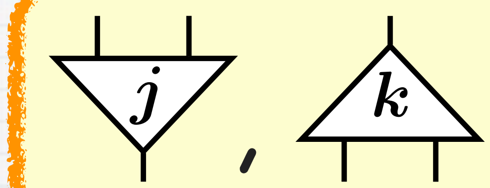
Here $\boldsymbol{K_I}$ is the constant functor into the monoidal unit $\boldsymbol{I}$;

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

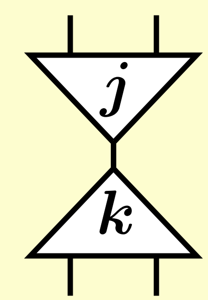
$$\begin{array}{c} \boldsymbol{j} : \boldsymbol{U} \otimes \boldsymbol{U} \lhd \boldsymbol{U} : \boldsymbol{k} \\ \boldsymbol{I} \lhd \boldsymbol{U} \\ \boldsymbol{u} : \boldsymbol{F}\boldsymbol{U} \lhd \boldsymbol{U} : \boldsymbol{v} \end{array}$$

* The reflexive object U

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



with



= id

GoI situation

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

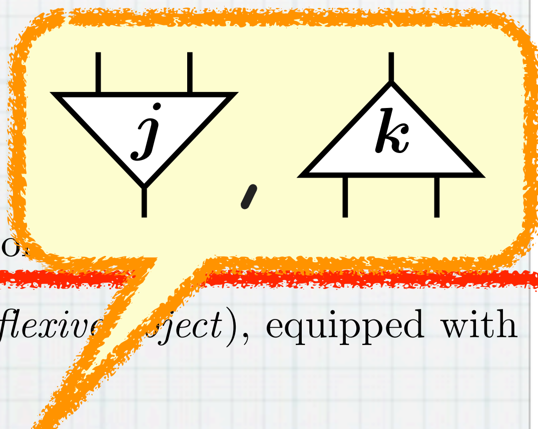
- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

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

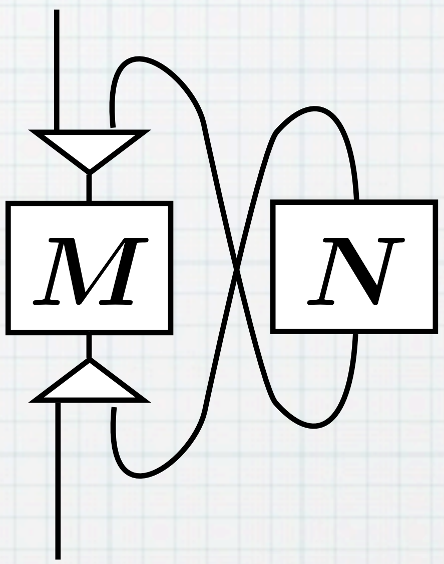
Here $\boldsymbol{K_I}$ is the constant functo

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

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



- The reflexive object U
  - Why for GoI?



  - Example in **Pfn**:

GoI situation

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

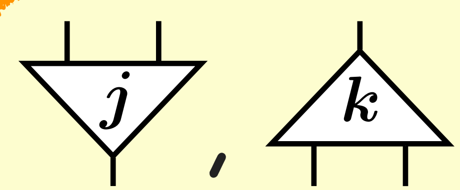
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

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

Here K_I is the constant functo

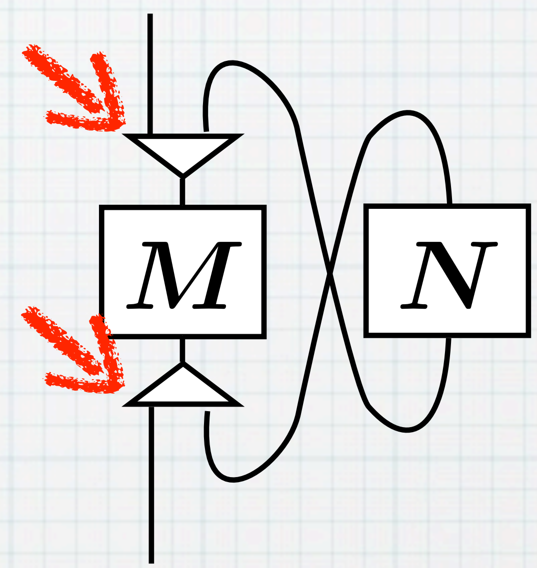
- $U \in \mathbb{C}$ is an object (called *reflexive* *ject*), equipped with the following retractions.

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



* The reflexive object U

* Why for GoI?



* Example in **Pfn**:

GoI situation

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

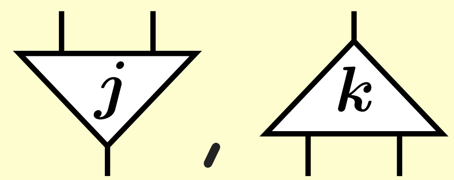
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

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

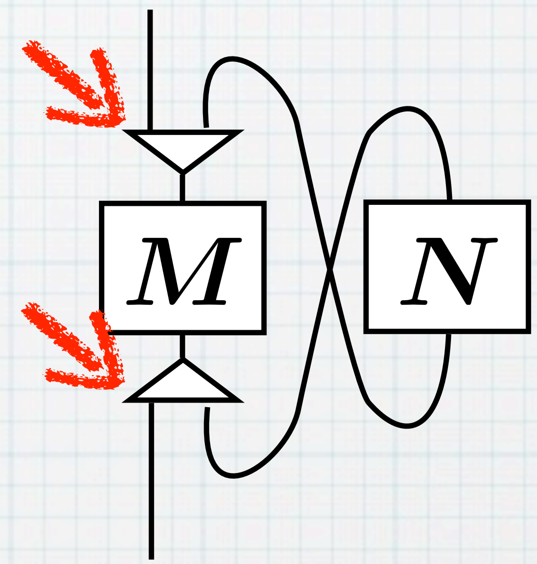
Here K_I is the constant functo

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

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



- The reflexive object U
- Why for GoI?



- Example in **Pfn**:

$$\mathbb{N} \in \mathbf{Pfn}, \text{ with}$$
$$\mathbb{N} + \mathbb{N} \cong \mathbb{N},$$
$$\mathbb{N} \cdot \mathbb{N} \cong \mathbb{N}$$

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).

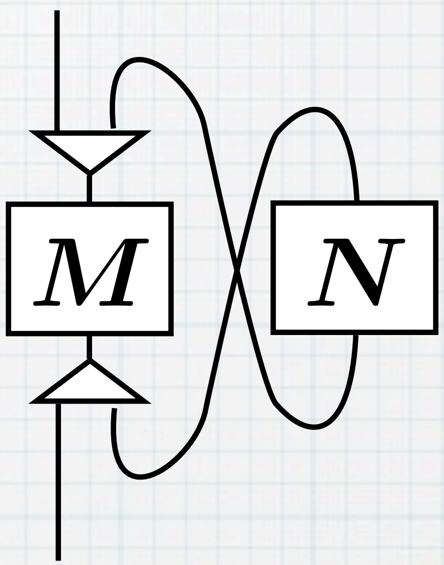
$$\begin{array}{rcl} e : FF \lhd F : e' & & \text{Comultiplication} \\ d : \mathrm{id} \lhd F : d' & & \text{Dereliction} \\ c : F \otimes F \lhd F : c' & & \text{Contraction} \\ w : K_I \lhd F : w' & & \text{Weakening} \end{array}$$

  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.

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

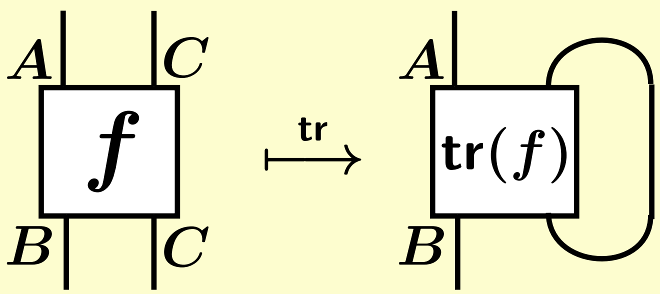
- Categorical axiomatics of the "GoI animation"



- Example:

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

Hasuo (Tokyo)



tuation: Summary

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

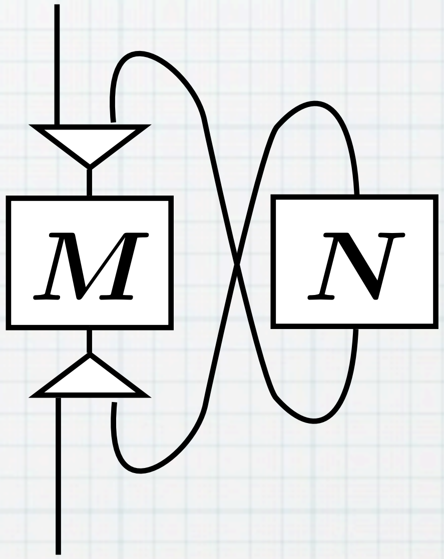
- $\mathbb{C} = (\mathbb{C}, \otimes, \boldsymbol{I})$ is a traced symmetric monoidal category (TSMC);
- $\boldsymbol{F} : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

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

  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.

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

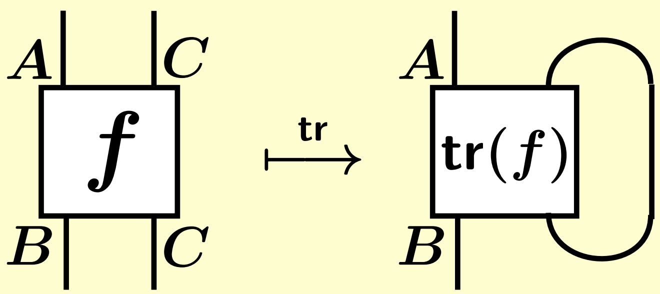
- Categorical axiomatics of the "GoI animation"



- Example:

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

Hasuo (Tokyo)



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):

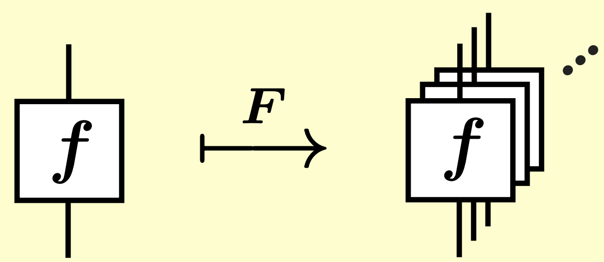
$$e : FF \lhd F : e'$$
$$d : \mathrm{id} \lhd F : d' \qquad \text{De}$$
$$c : F \otimes F \lhd F : c' \qquad \text{Co}$$
$$w : K_I \lhd F : w' \qquad \text{We}$$

Here K_I is the constant functor into th

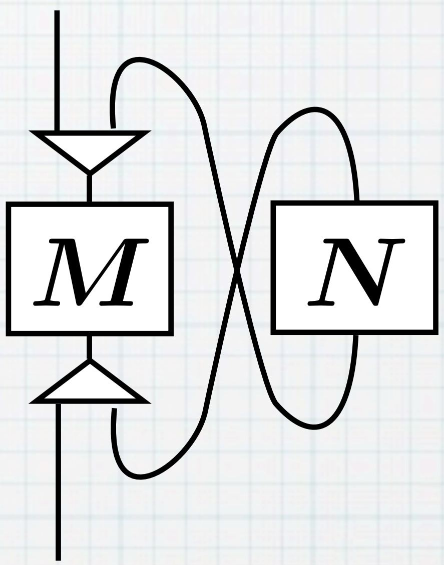
- $U \in \mathbb{C}$ is an object (called *reflexive obj*
the following retractions.

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

For ! , via



- Categorical axiomatics of the "GoI animation"

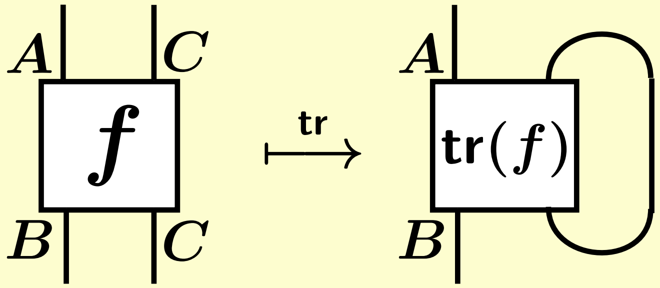


- Example:

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

Hasuo (Tokyo)

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):

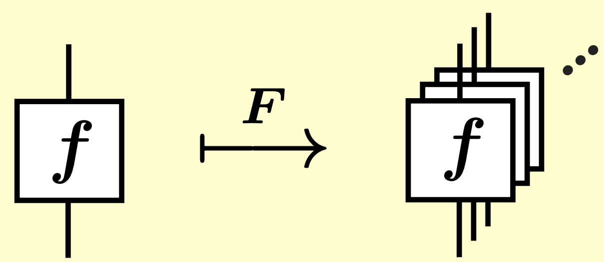
$$e : FF \lhd F : e'$$
$$d : \mathrm{id} \lhd F : d' \qquad \text{De}$$
$$c : F \otimes F \lhd F : c' \qquad \text{Co}$$
$$w : K_I \lhd F : w' \qquad \text{We}$$

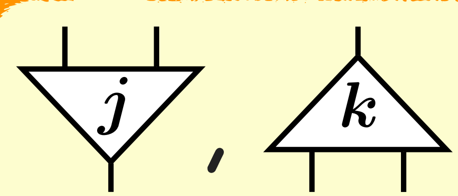
Here K_I is the constant functor into th

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

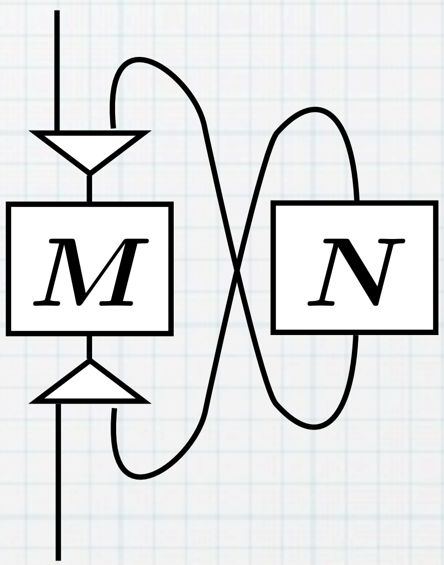
$$j : U \otimes U \lhd U : k$$
$$I \lhd U$$
$$u : FU \lhd U : v$$

For ! , via





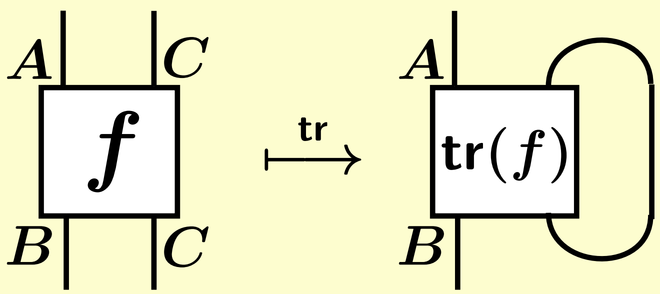
- Categorical axiomatics of the "GoI animation"



- 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 lowing retractions (which are monoidal natural transformations):

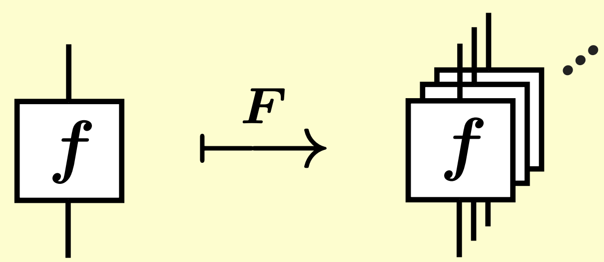
$$e : FF \lhd F : e'$$
$$d : \mathrm{id} \lhd F : d' \qquad \text{De}$$
$$c : F \otimes F \lhd F : c' \qquad \text{Co}$$
$$w : K_I \lhd F : w' \qquad \text{We}$$

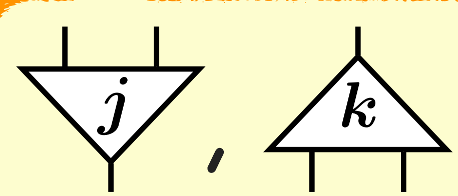
Here K_I is the constant functor into t

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

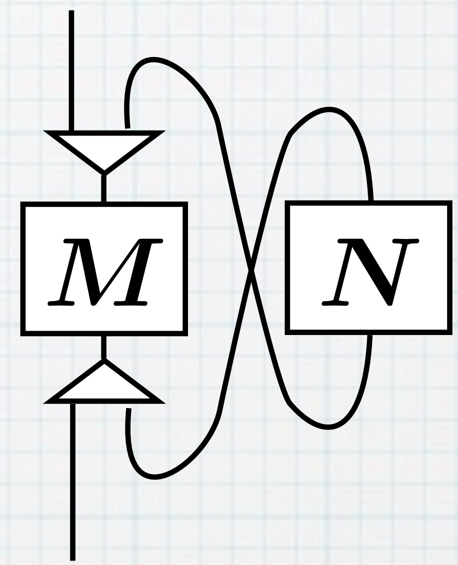
$$j : U \otimes U \lhd U : k$$
$$I \lhd U$$
$$u : FU \lhd U : v$$

For ! , via





* Categorical axiomatics of the "GoI animation"



* Example:

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

Hasuo (Tokyo)

Friday, June 24, 16

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, ...

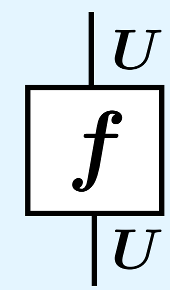
Categorical GoI: Constr. of an LCA

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, ...



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

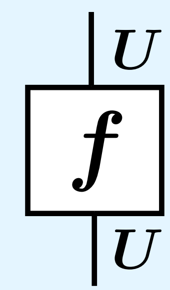
Categorical GoI: Constr. of an LCA

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

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

carries a canonical LCA structure.

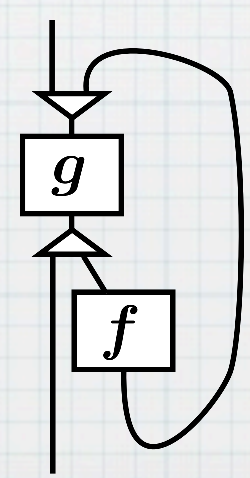
- Applicative str. $\cdot$
- ! operator
- Combinators B, C, I, ...

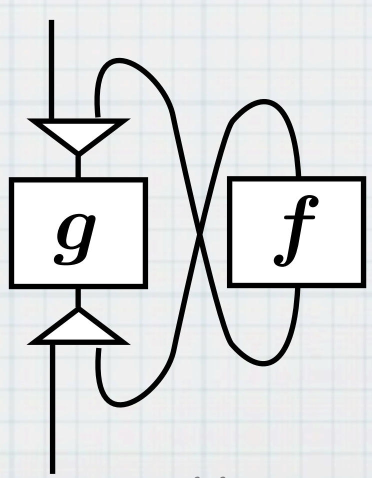


$\in \mathbb{C}(U, U)$

- $g \cdot f$

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

$=$ 

$=$ 

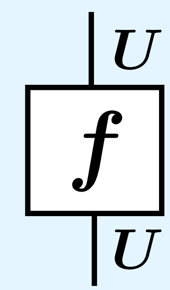
Categorical GoI: Constr. of an LCA

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.

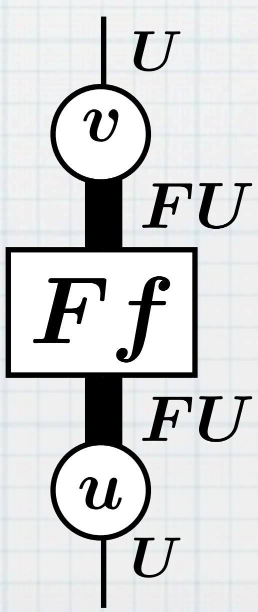
* Applicative str. $\cdot$
* ! operator
* Combinators B, C, I, ...



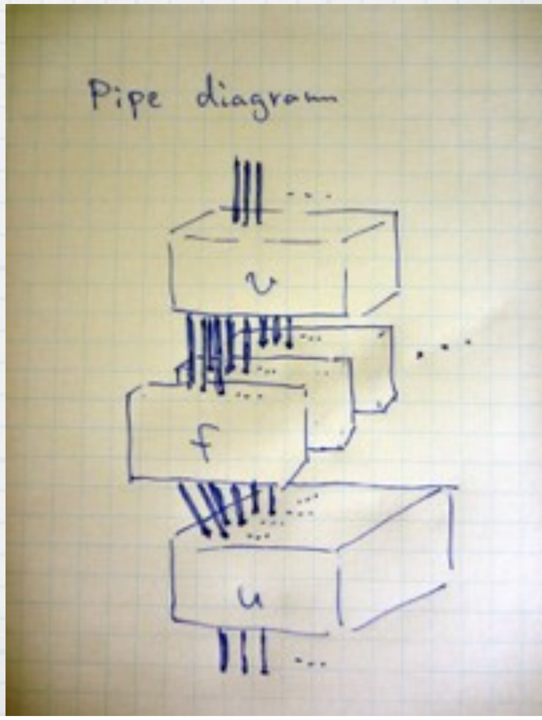
$\in \mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$

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

$=$



$=$



Categorical GoI: Constr. of an LCA

* Combinator $Bxyz = x(yz)$

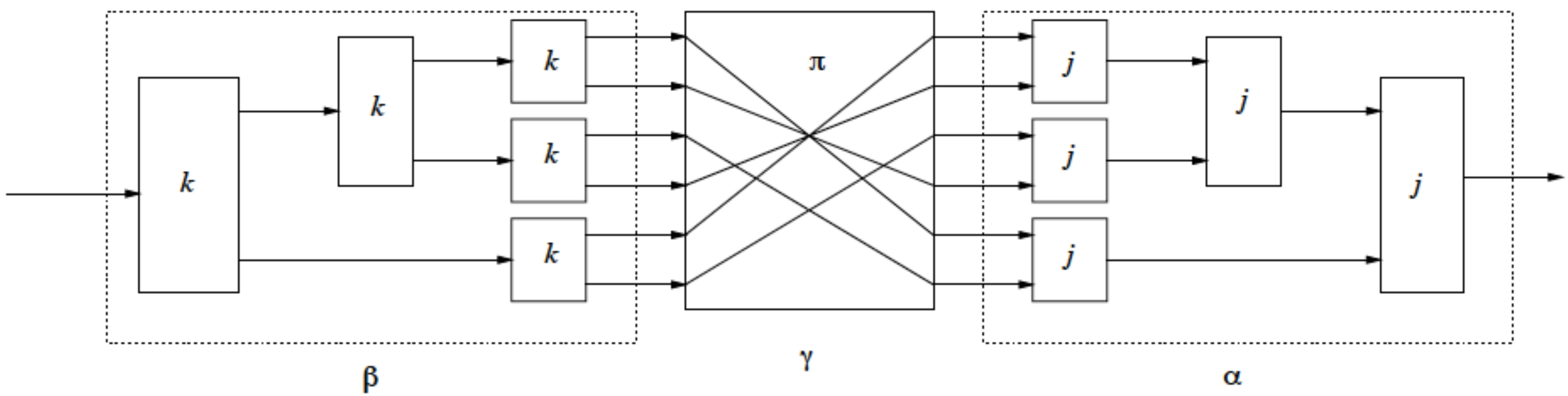
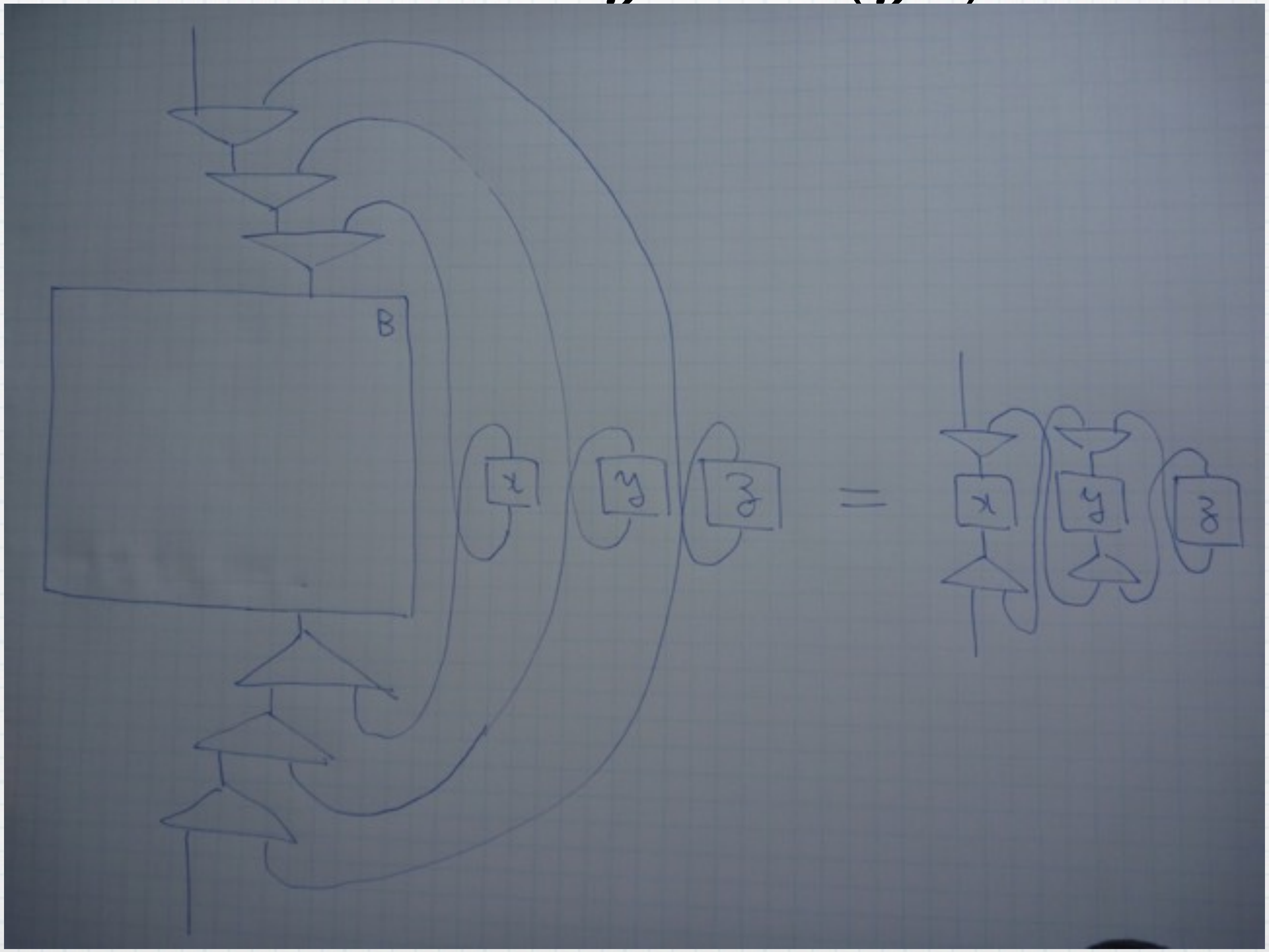


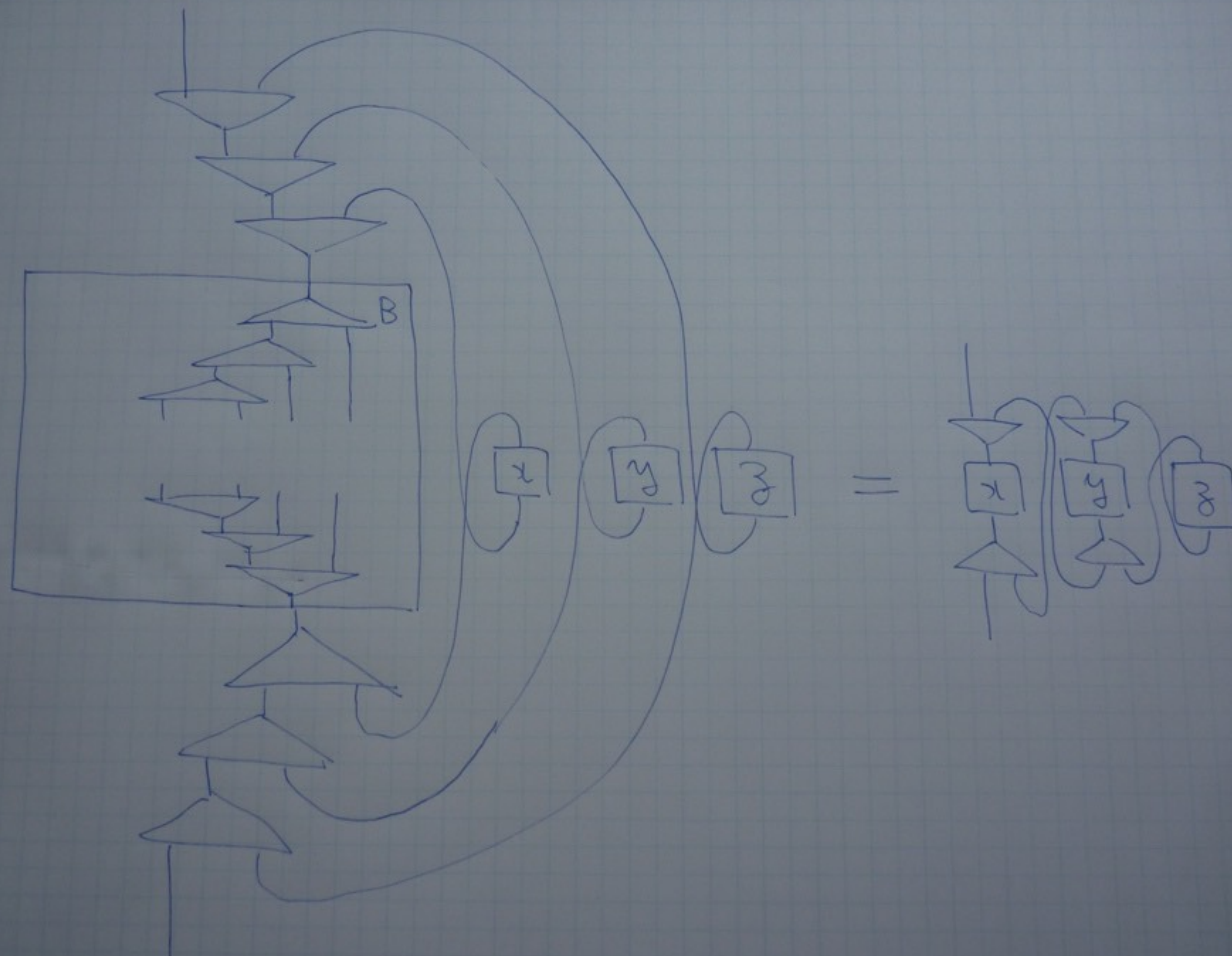
Figure 7: Composition Combinator B

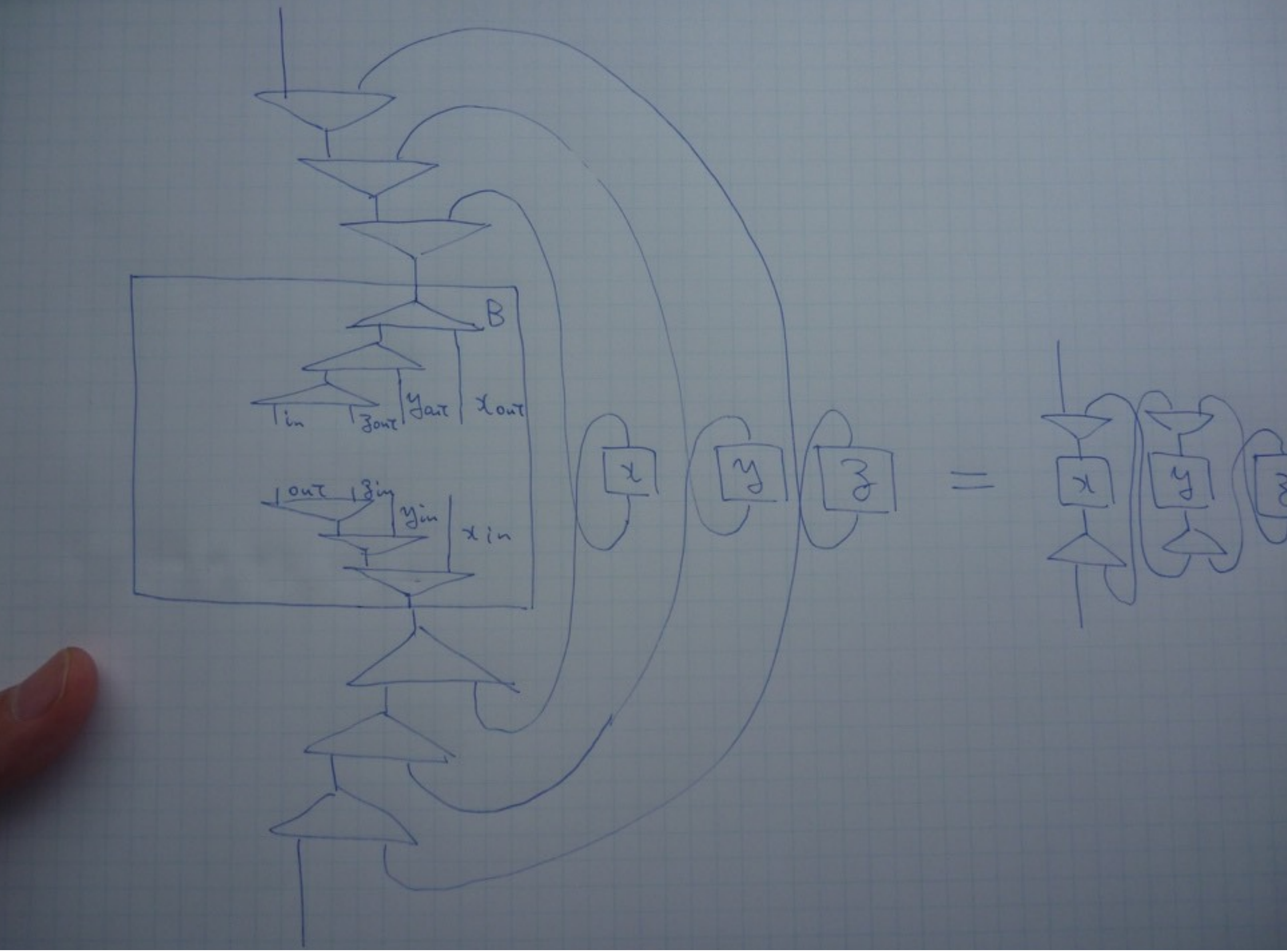
from [AHS02]

Categorical GoI: Constr. of an LCA

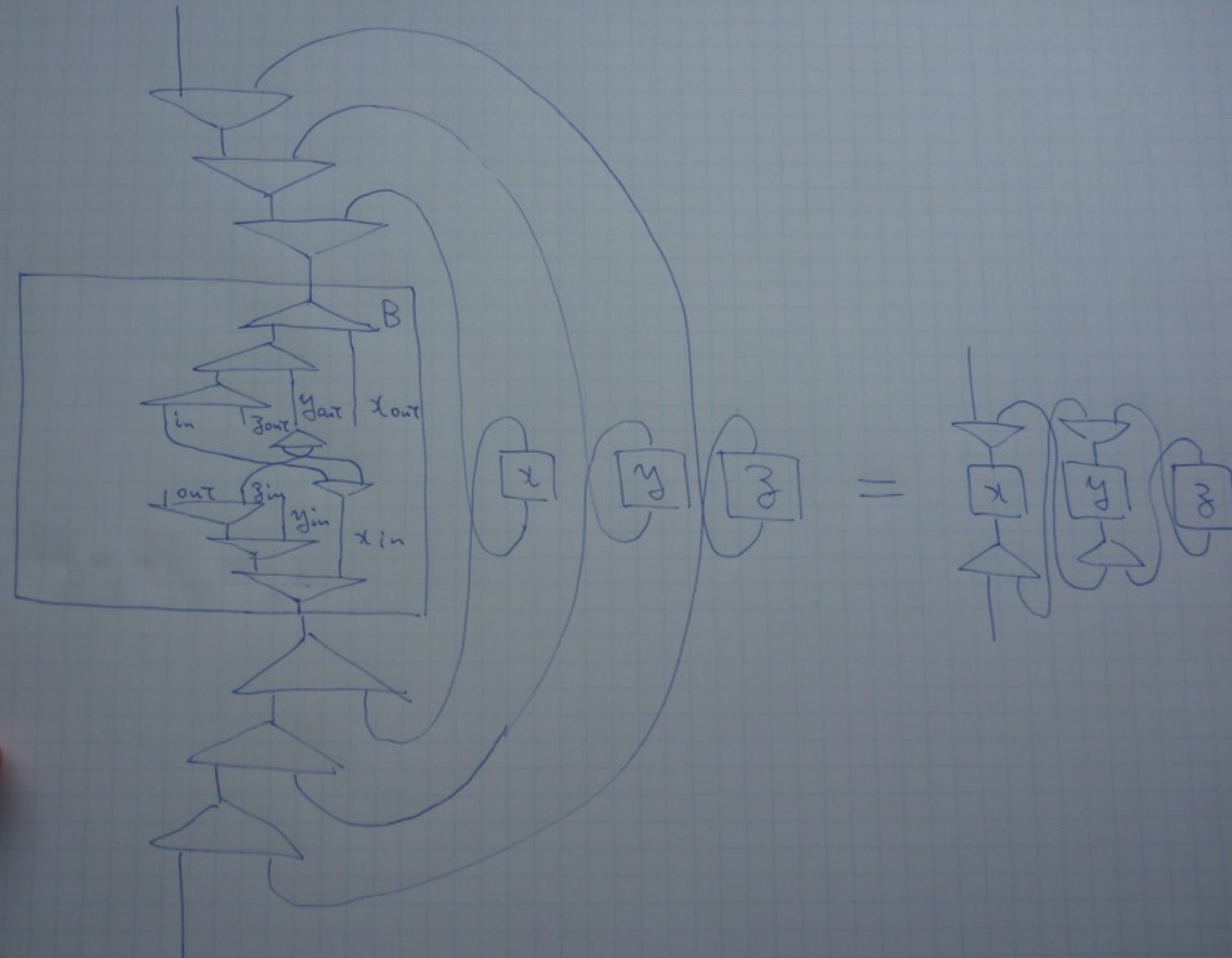
* Combinator $Bxyz = x(yz)$

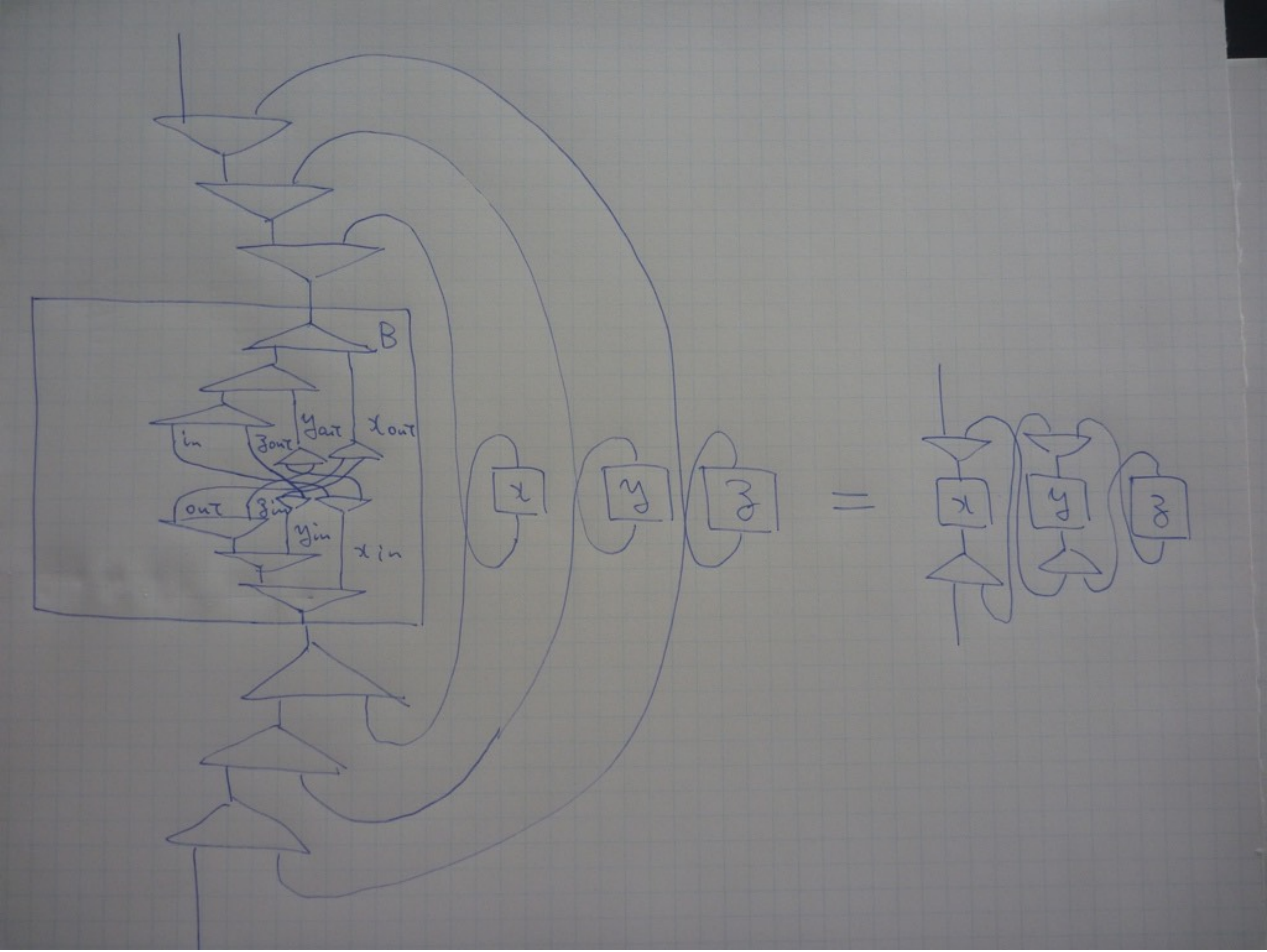



B
x
y
z
=
x
y
z



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


B
in
z_{out}
y_{out}
x_{out}
out
z_{in}
y_{in}
x_{in}
x
y
z
=
x
y
z



B
y_{out}
x_{out}
in
z_{out}
out
z_{in}
y_{in}
x_{in}
x
y
z
=
x
y
z

Categorical GoI: Constr. of an LCA

- Combinator $Bxyz = x(yz)$

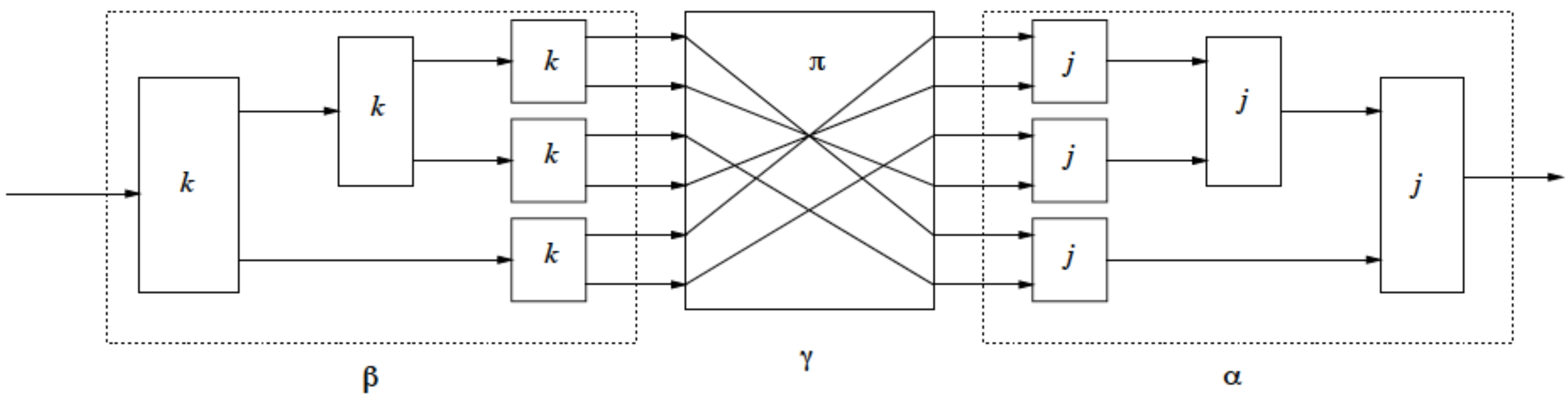


Figure 7: Composition Combinator B

from [AHS02]

Categorical GoI: Constr. of an LCA

* Combinator $Bxyz = x(yz)$

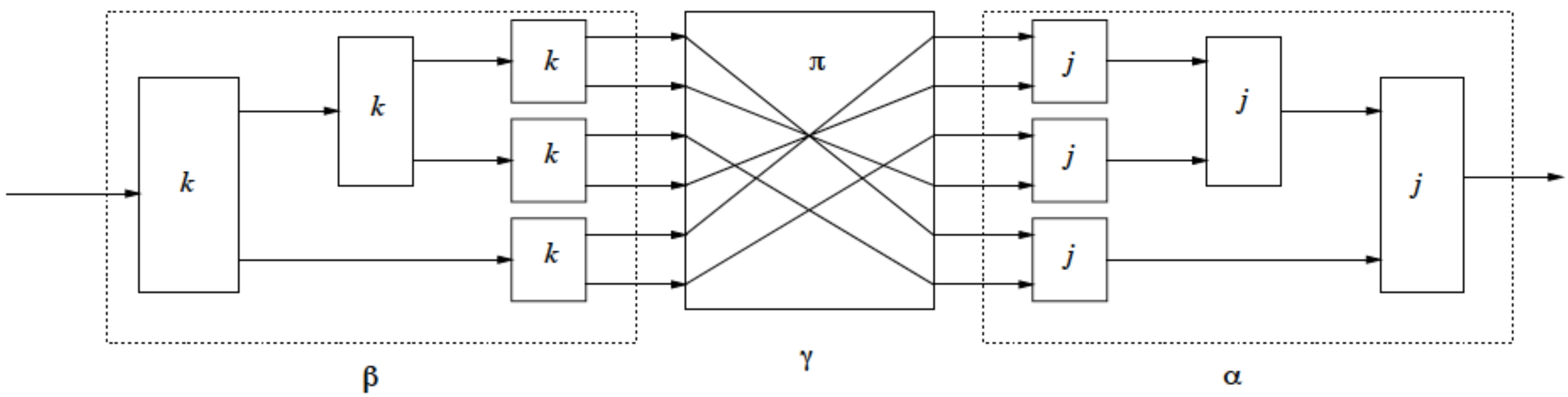


Figure 7: Composition Combinator B

from [AHS02]

Nice dynamic interpretation of (linear) computation!!

Summary: Categorical GoI

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

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

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

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

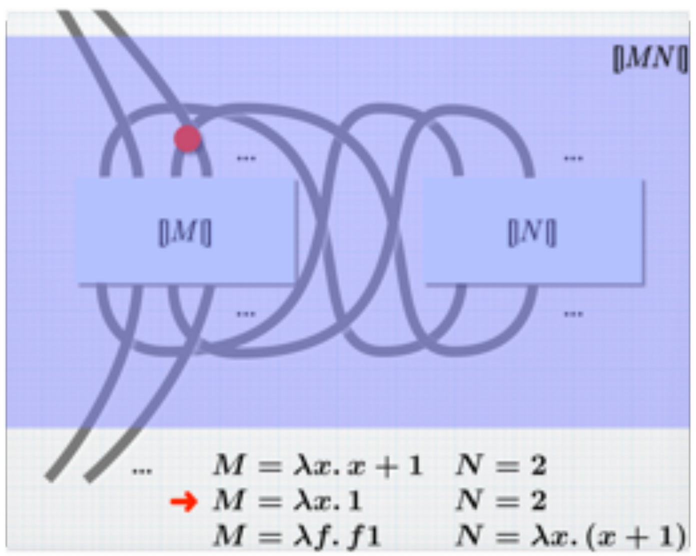
carries a canonical LCA structure.

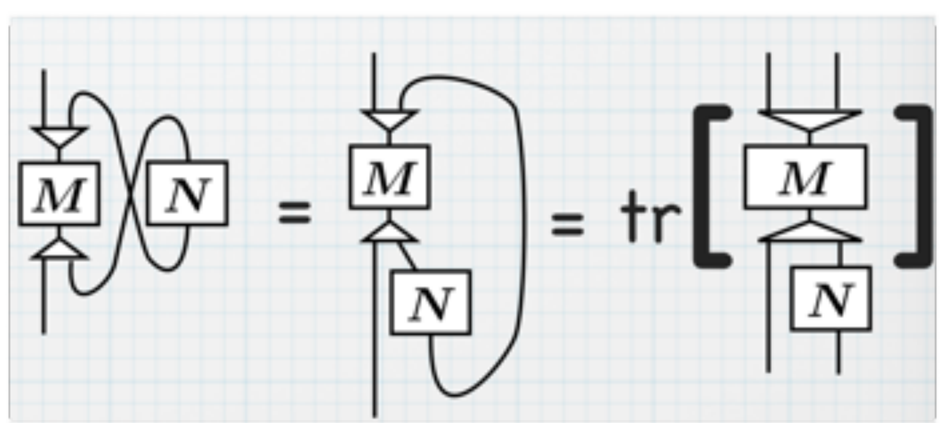
Hasuo (Tokyo)

Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

"GoI Animation"





Categorical GoI

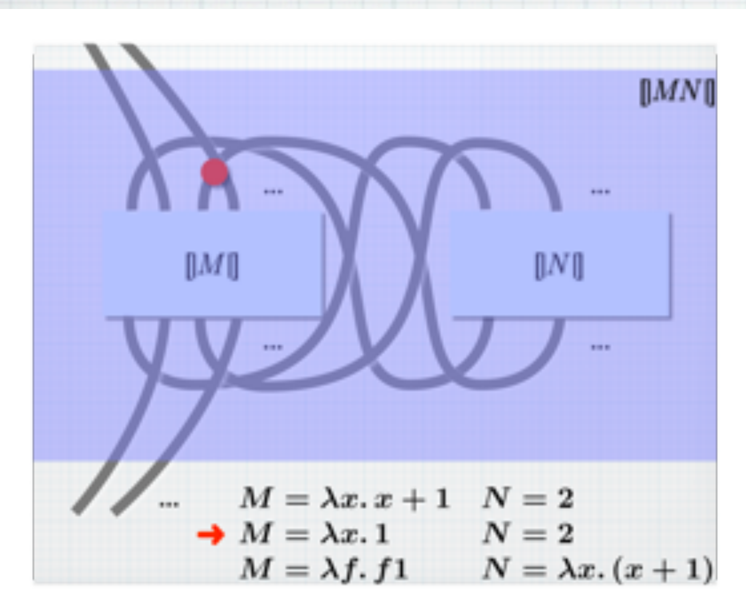
[Abramsky, Haghverdi & Scott, MSCS'02]

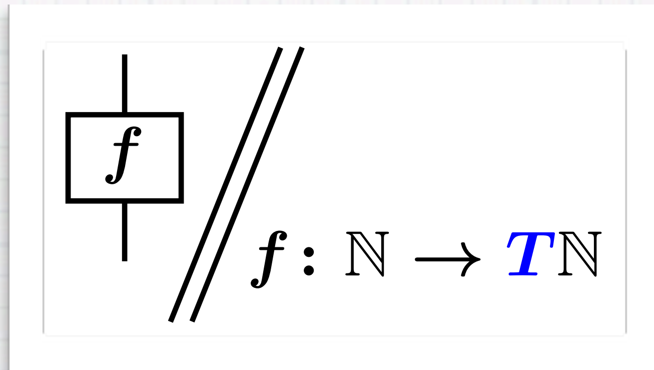
Hasuo (Tokyo)

Outline

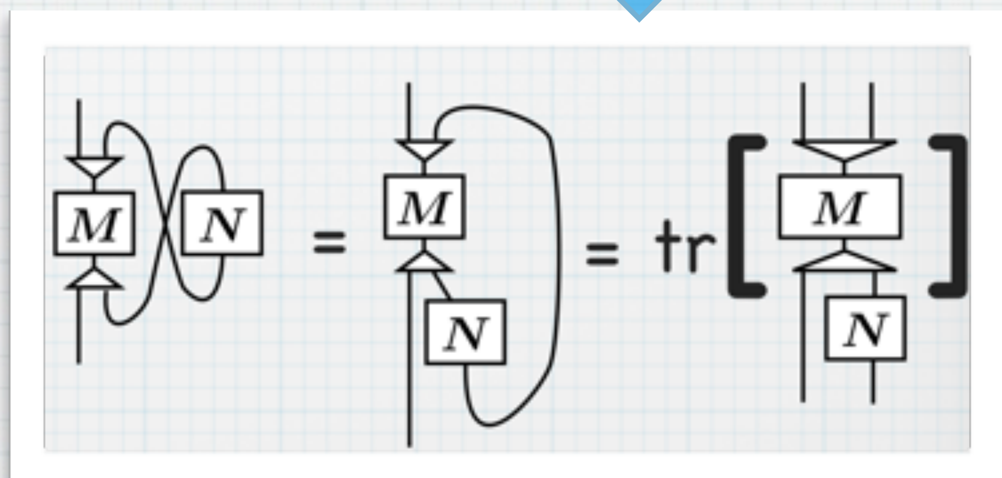
Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

"GoI Animation"





GoI w/ T-branching
[IH & Hoshino, LICS'11]

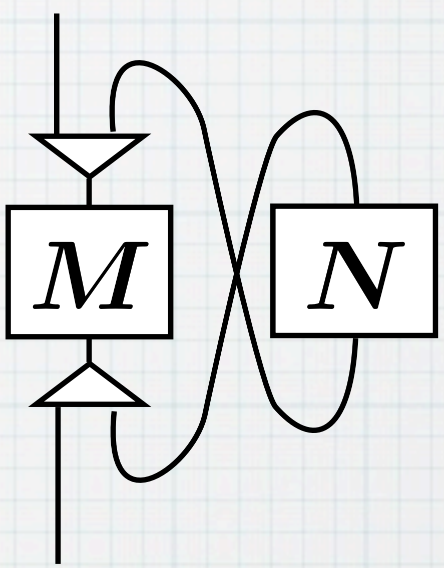


Categorical GoI
[Abramsky, Haghverdi & Scott, MSCS'02]

Hasuo (Tokyo)

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

* Strategy: find a TSMC!

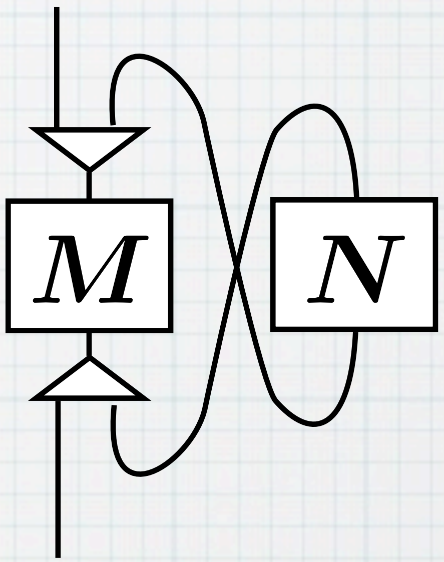


* "Wave-style" examples
  * $\otimes$ is Cartesian product(-like)
    * in which case,

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

Why Categorical Generalization?: Examples Other Than $\mathbf{Pfn}$ [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]

* $\mathbf{Pfn}$ (partial functions)

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

* $\mathbf{Rel}$ (relations)

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

* $\mathbf{DSRel}$

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

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

Why Catego
Examples [AHS02]

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

* **Pfn** (partial functions)

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

* **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**

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

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

Why Catego… Examples Other Than Pfn [AHS02]

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

* **Pfn** (partial functions)

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

(Potential) non-termination

* **Rel** (relations)

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

Non-determinism

* **DSRel**

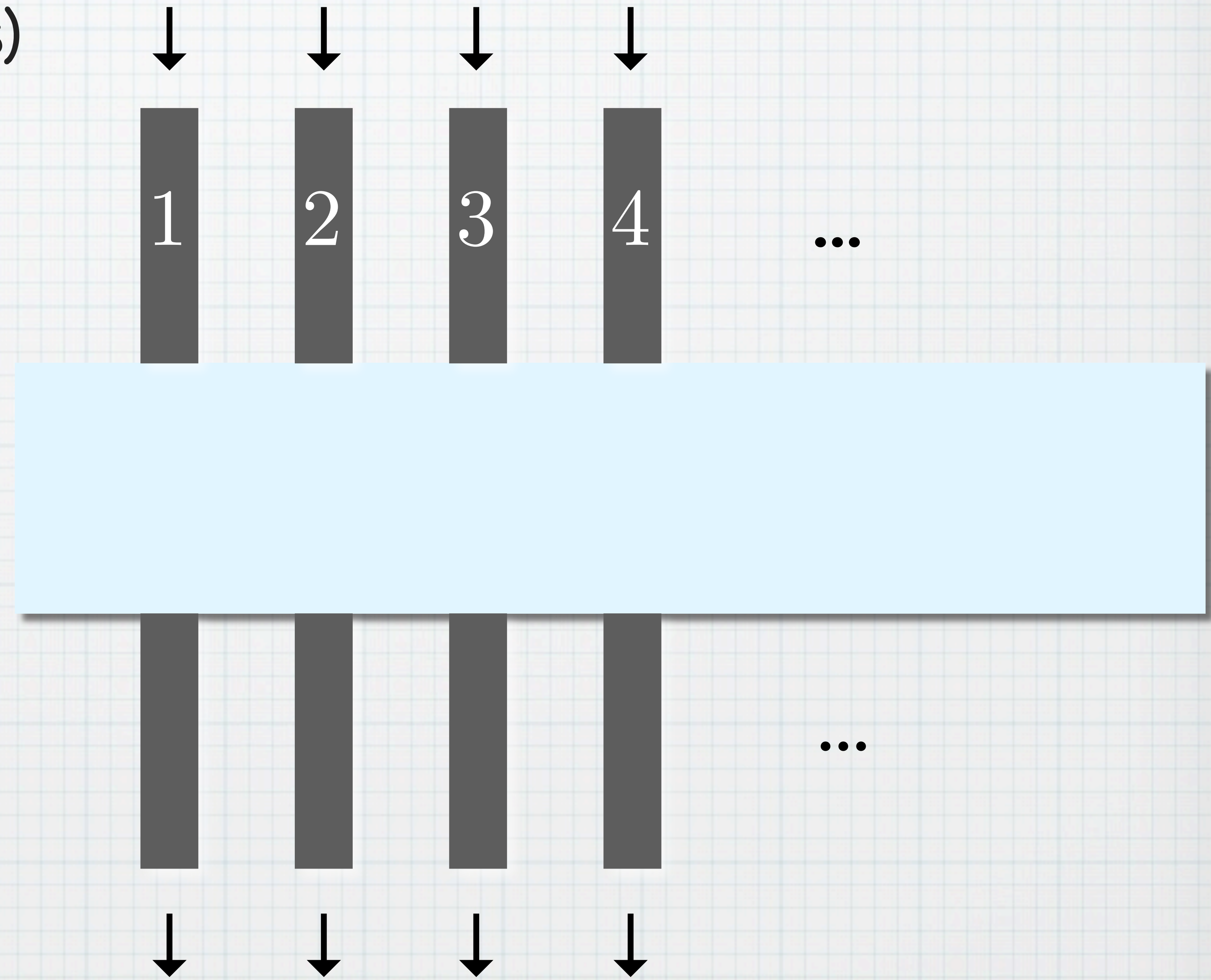
$$\dfrac{X \to Y \text{ in } \mathbf{DSRel}}{X \to \mathcal{D}Y \text{ in } \mathbf{Sets}}$$

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

Probabilistic branching

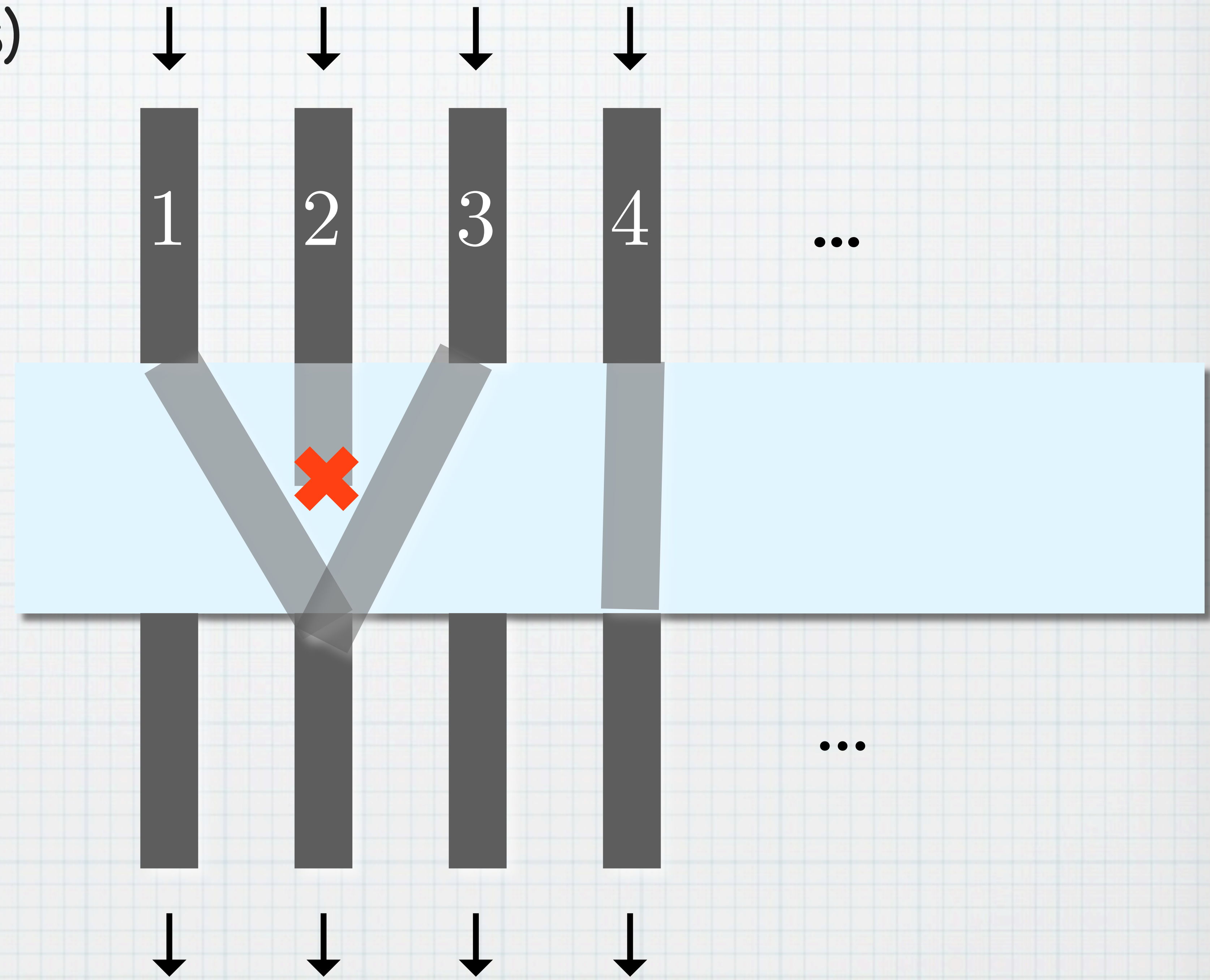
Different Branching in The GoI Animation

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



Different Branching in The GoI Animation

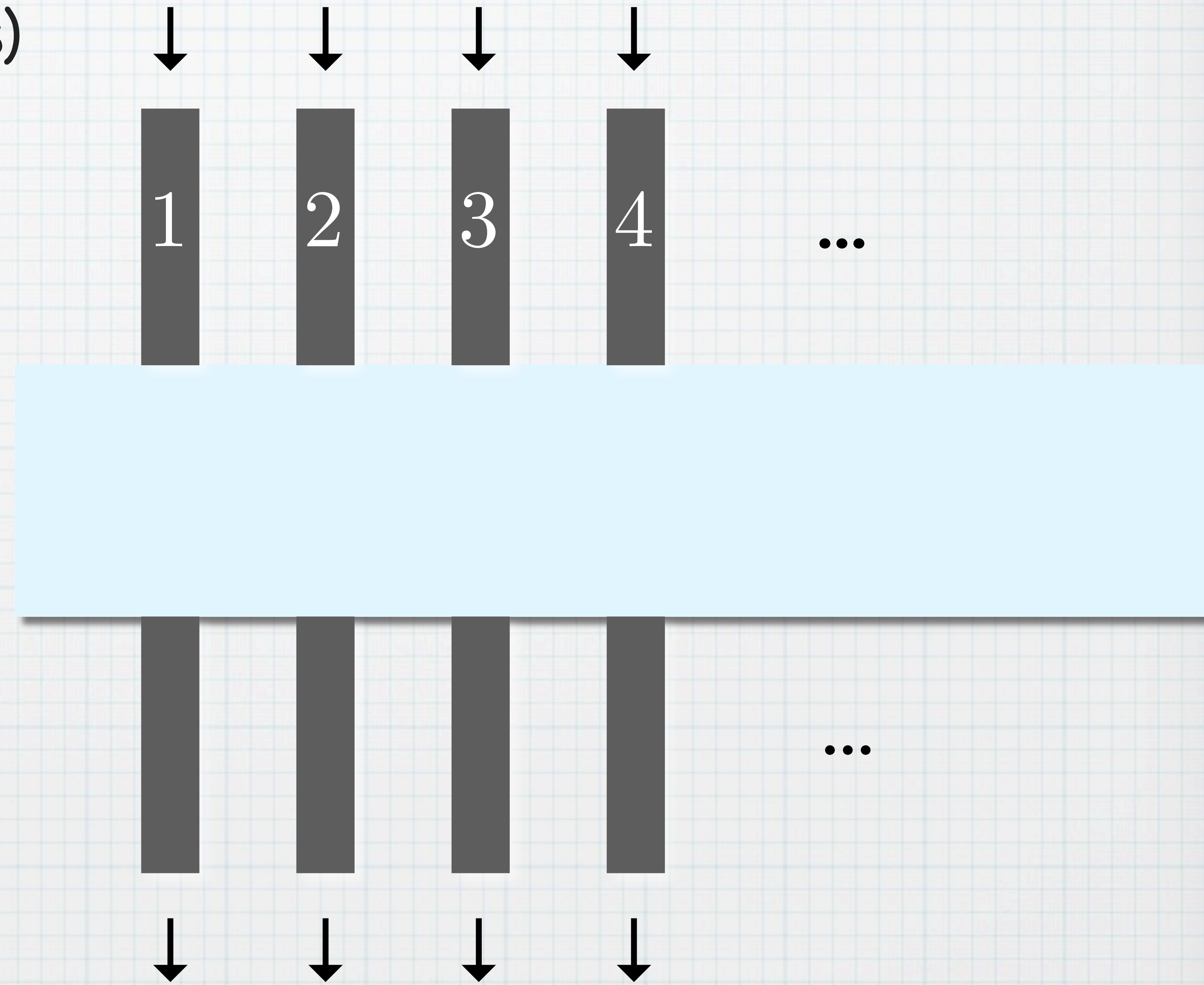
- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Hasuo (Tokyo)

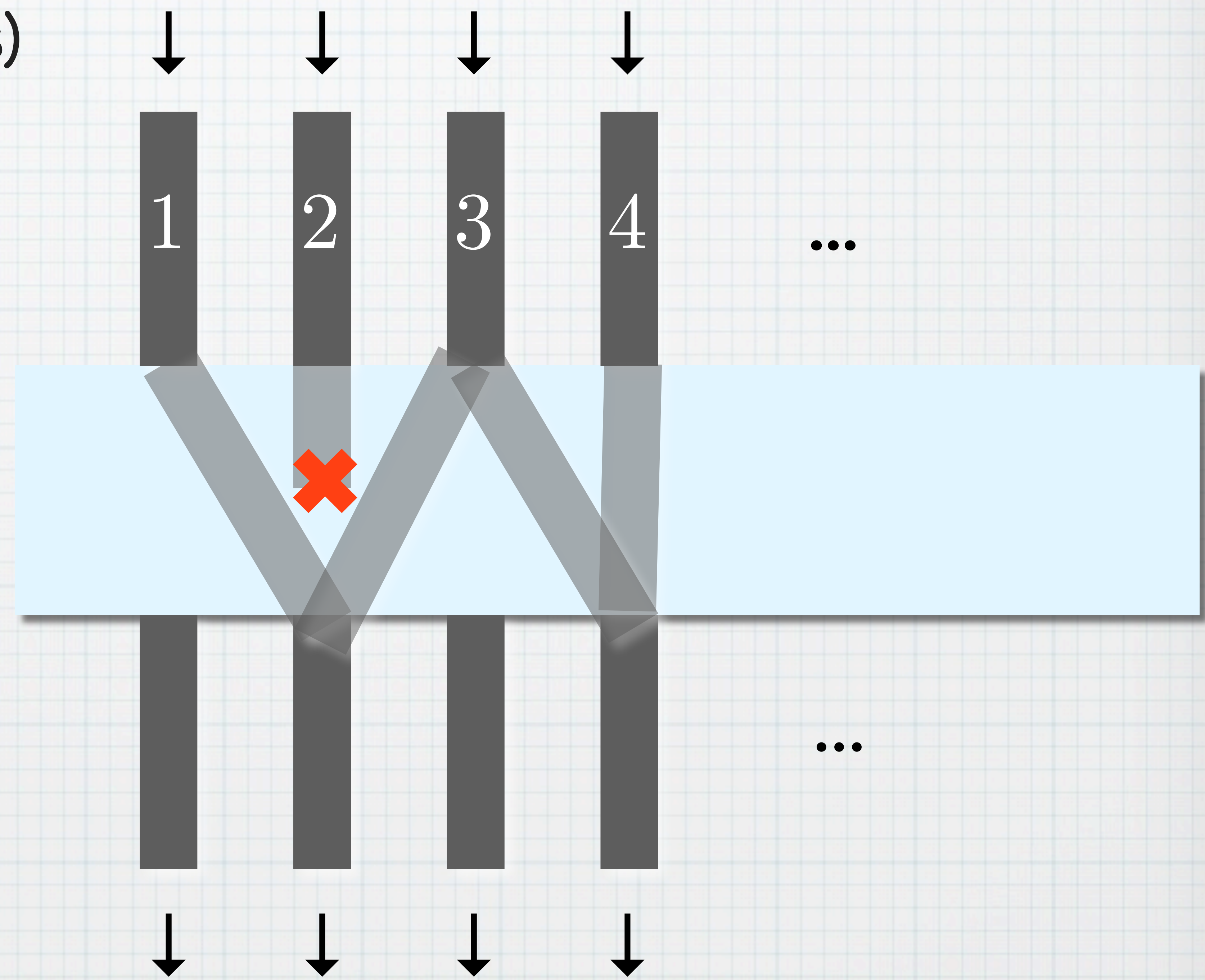
Different Branching in The GoI Animation

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



Different Branching in The GoI Animation

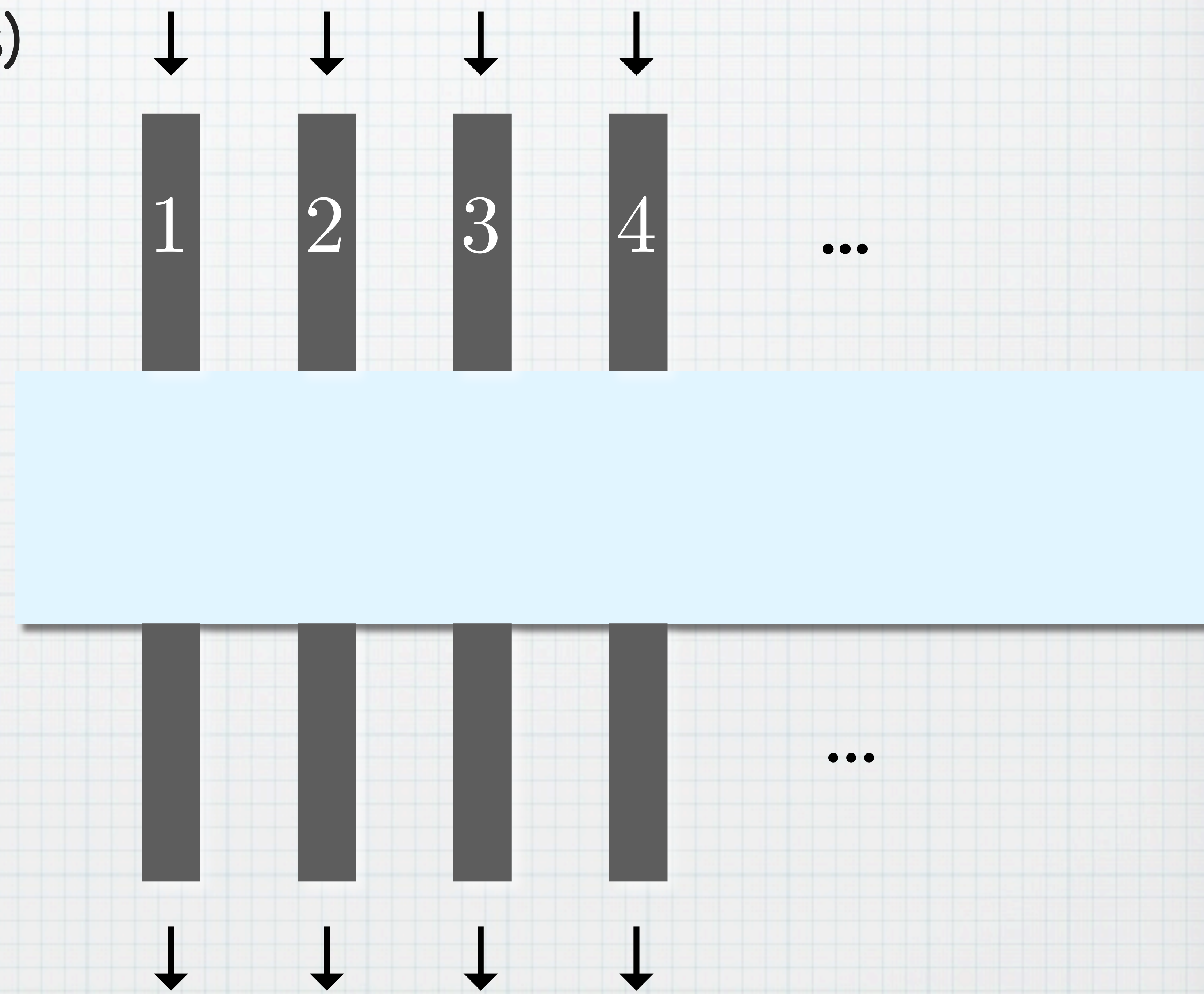
- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Hasuo (Tokyo)

Different Branching in The GoI Animation

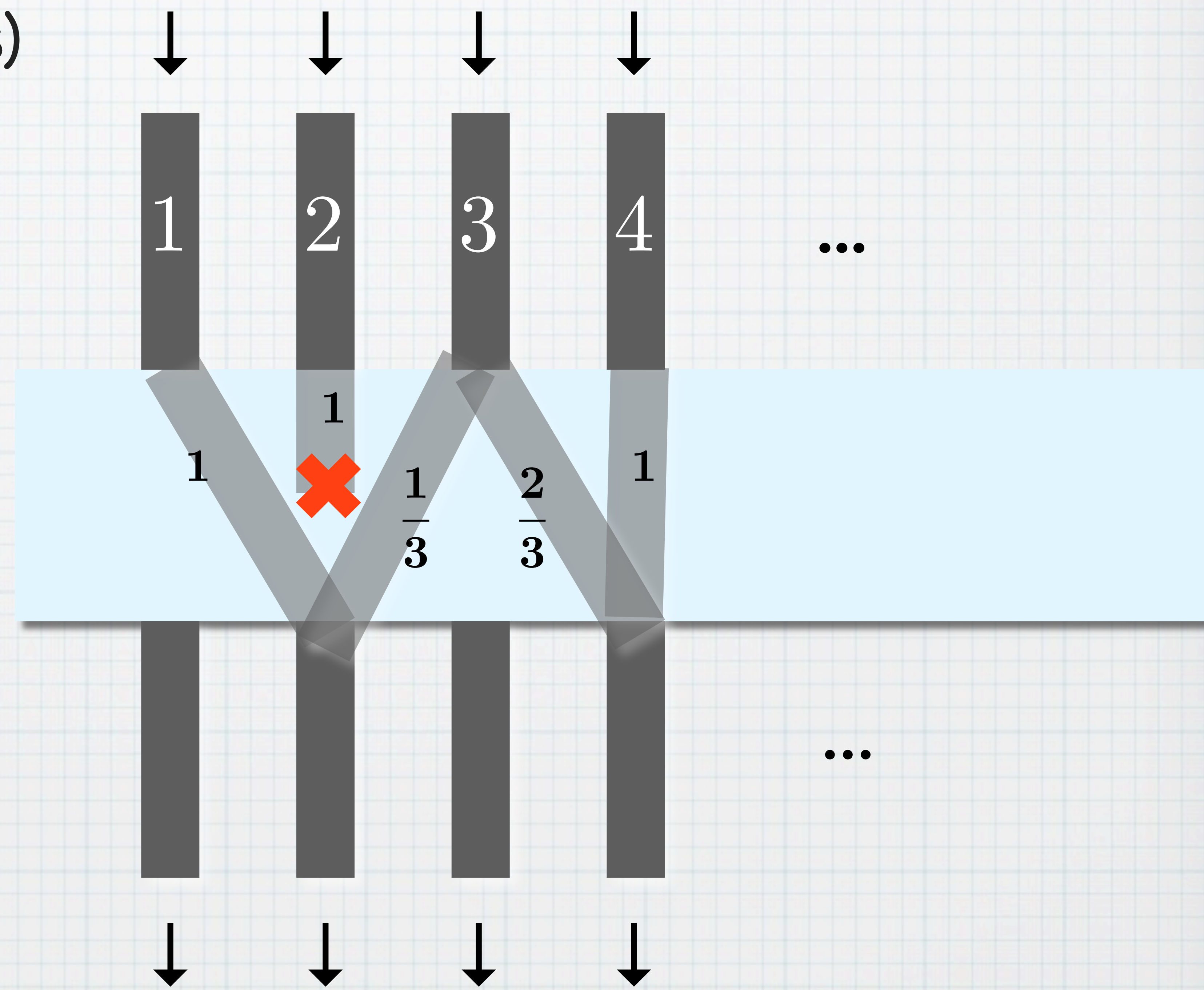
* **Pfn** (partial functions)
  * Pipes can be stuck
* **Rel** (relations)
  * Pipes can branch
* **DSRel**
  * Pipes can branch probabilistically



Hasuo (Tokyo)

Different Branching in The GoI Animation

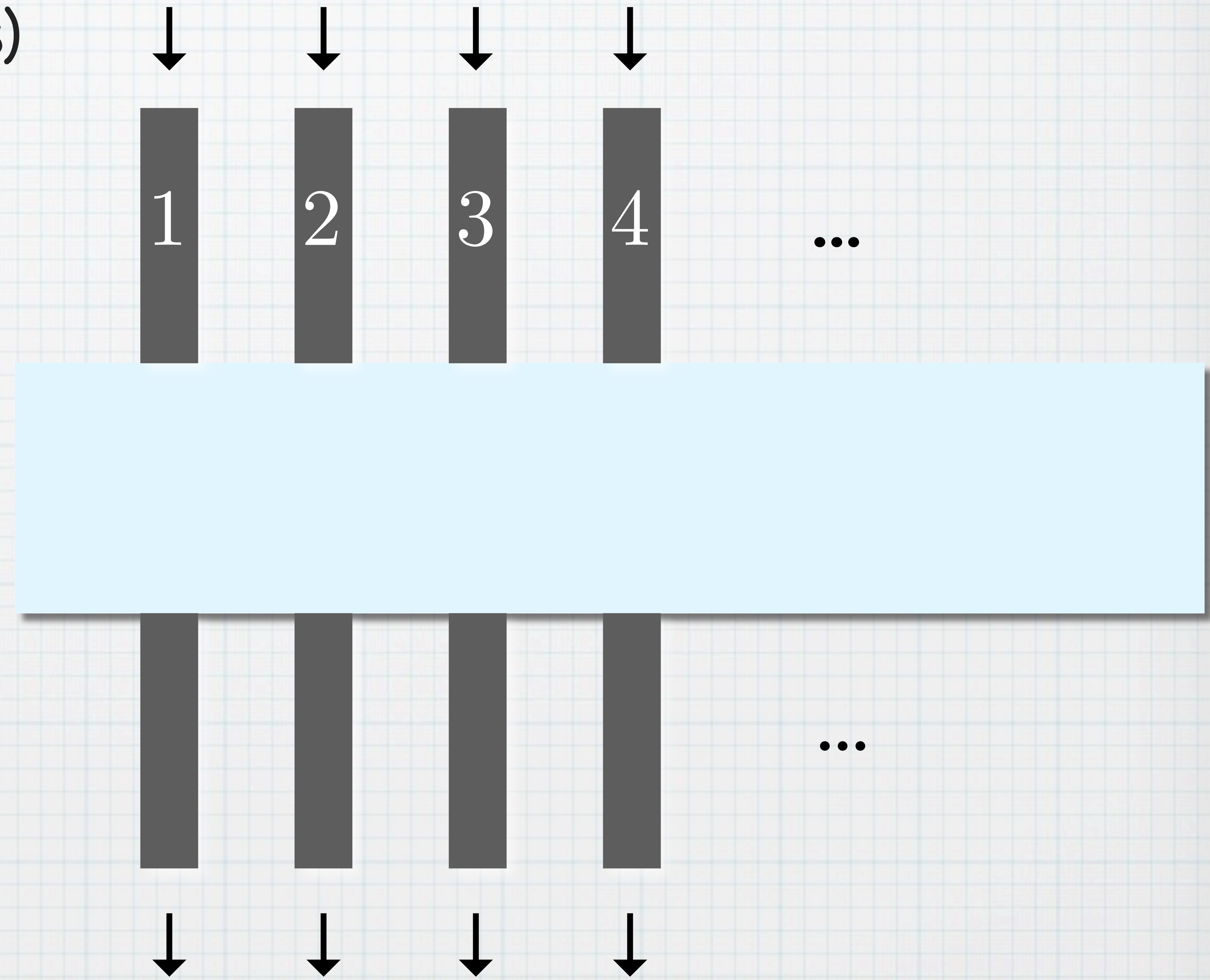
- **Pfn** (partial functions)
  - Pipes can be stuck
- **Rel** (relations)
  - Pipes can branch
- **DSRel**
  - Pipes can branch probabilistically



Hasuo (Tokyo)

Different Branching in The GoI Animation

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



A Coalgebraic View

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

* In my thesis (2008):
  * Coalgebras in a Kleisli category $Kl(T)$

$$\frac{X \to Y \text{ in } \mathcal{K}\ell(T)}{X \to TY \text{ in } \mathbf{Sets}}$$

  * ➜ Generic theory of **trace** and **simulations**

Hasuo (Tokyo)

Why Catego… Examples Other Than Pfn

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

* **Pfn** (partial functions)

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

(Potential) non-termination

* **Rel** (relations)

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

Non-determinism

* **DSRel**

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

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

Probabilistic branching

Why Catego… Examples

$Kl(T)$ for different branching monads T

* **Pfn** (partial functions)

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

(Potential) non-termination

* **Rel** (relations)

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

Non-determinism

* **DSRel**

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

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

Probabilistic branching

Branching Monad: Source of Particle-Style GoI Situations

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

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

is

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

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

Hasuo (Tokyo)

Branching Monad: Source of Particle-Style GoI Situations

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

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

is

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

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

Monads in [Hasuo,Jacobs&Sokolova07]

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

Hasuo (Tokyo)

Branching Monad: Source of Particle-Style GoI Situations

Thm. ([Jacobs,CMCS10])
Given a "branching monad" T on **Sets**, the monoidal category

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

is

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

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

Monads in [Hasuo,Jacobs&Sokolova07]

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

Hasuo (Tokyo)

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

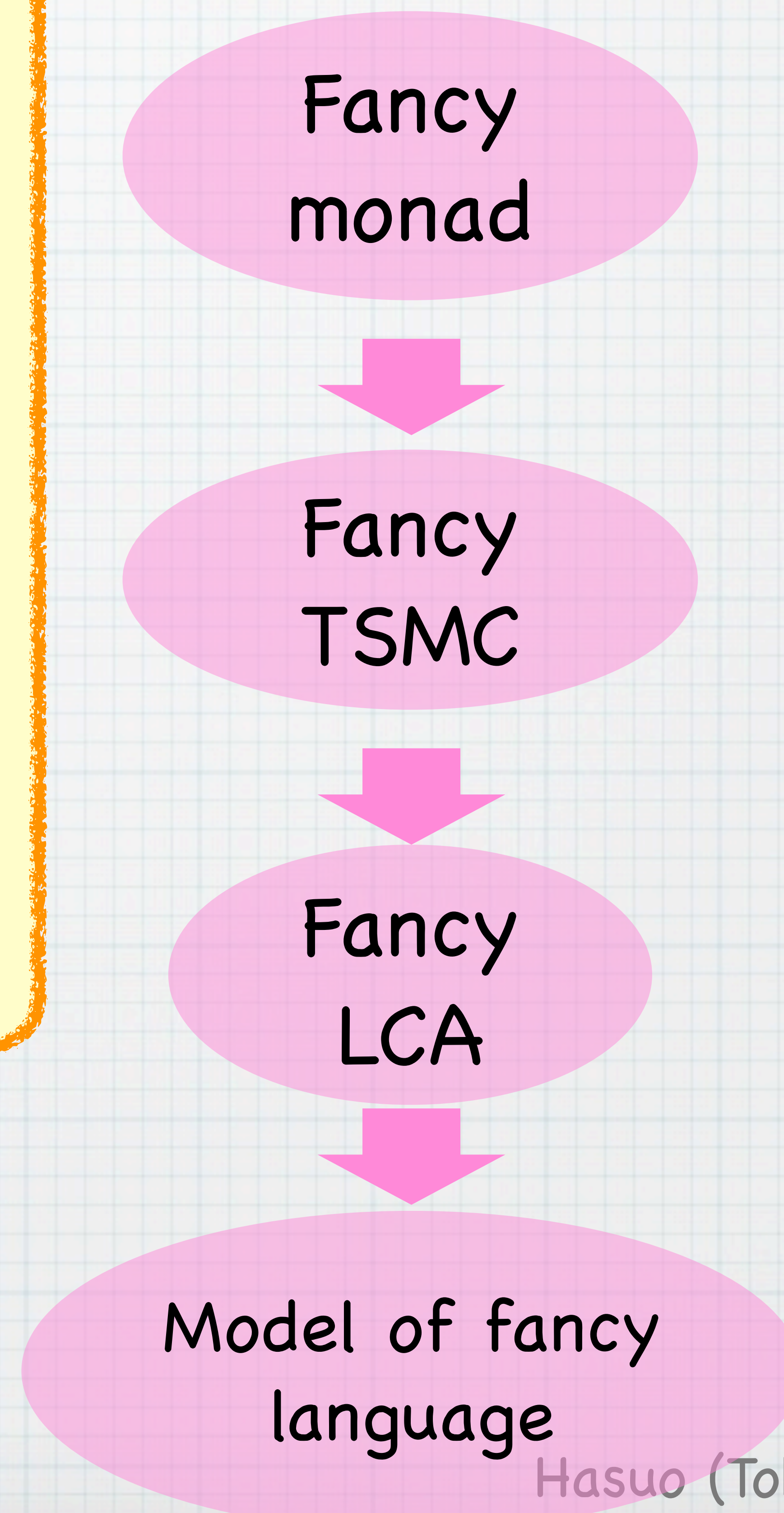
Model of fancy language

Workflow

* Model for (a variant of) the Selinger-Valiron **quantum λ-calculus**

  (linear λ + prep./Unitary/meas.)
  [Hasuo & Hoshino, LICS'11 & APAL'16]

* via the **quantum branching monad**

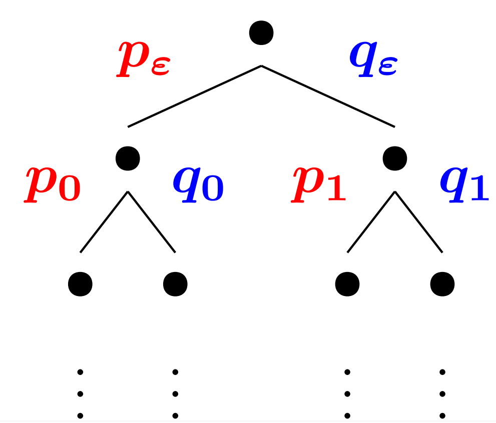


* Model for (a variant of) the Selinger-Valiron **quantum λ-calculus**
(linear λ + prep./Unitary/meas.)
[Hasuo & Hoshino, LICS'11 & APAL'16]
* via the **quantum branching monad**
* ... with considerable complication :(

$$[\![\Gamma \vdash M : \tau]\!] : \ [\![\Gamma]\!] \longrightarrow ([\![\tau]\!] \multimap R) \multimap R$$

where

$R = \Big\{$



$\Big| \ p_\alpha, q_\alpha \in [0,1] \Big\}$

Realizability

Linear category

Workflow

Fancy monad

Fancy TSMC

Fancy LCA

Model of fancy language

Workflow

- Model for (a variant of) the Selinger-Valiron **quantum λ-calculus**

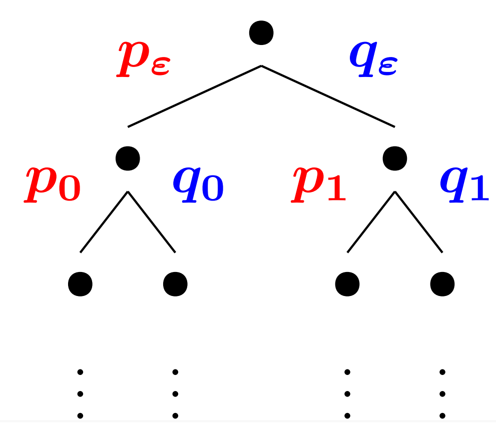
  (linear λ + prep./Unitary/meas.)
  [Hasuo & Hoshino, LICS'11 & APAL'16]

- via the **quantum branching monad**
- ... with considerable complication :(

$$[\![\Gamma \vdash M : \tau]\!] : \; [\![\Gamma]\!] \longrightarrow ([\![\tau]\!] \multimap R) \multimap R$$

where

$$R = \left\{ \quad \cdots \quad \middle|\; p_\alpha, q_\alpha \in [0,1] \right\}$$



- Records measurement outcomes
- R as a suitable **final coalgebra** in the realizability category

Fancy monad

Fancy LCA

Model of fancy language

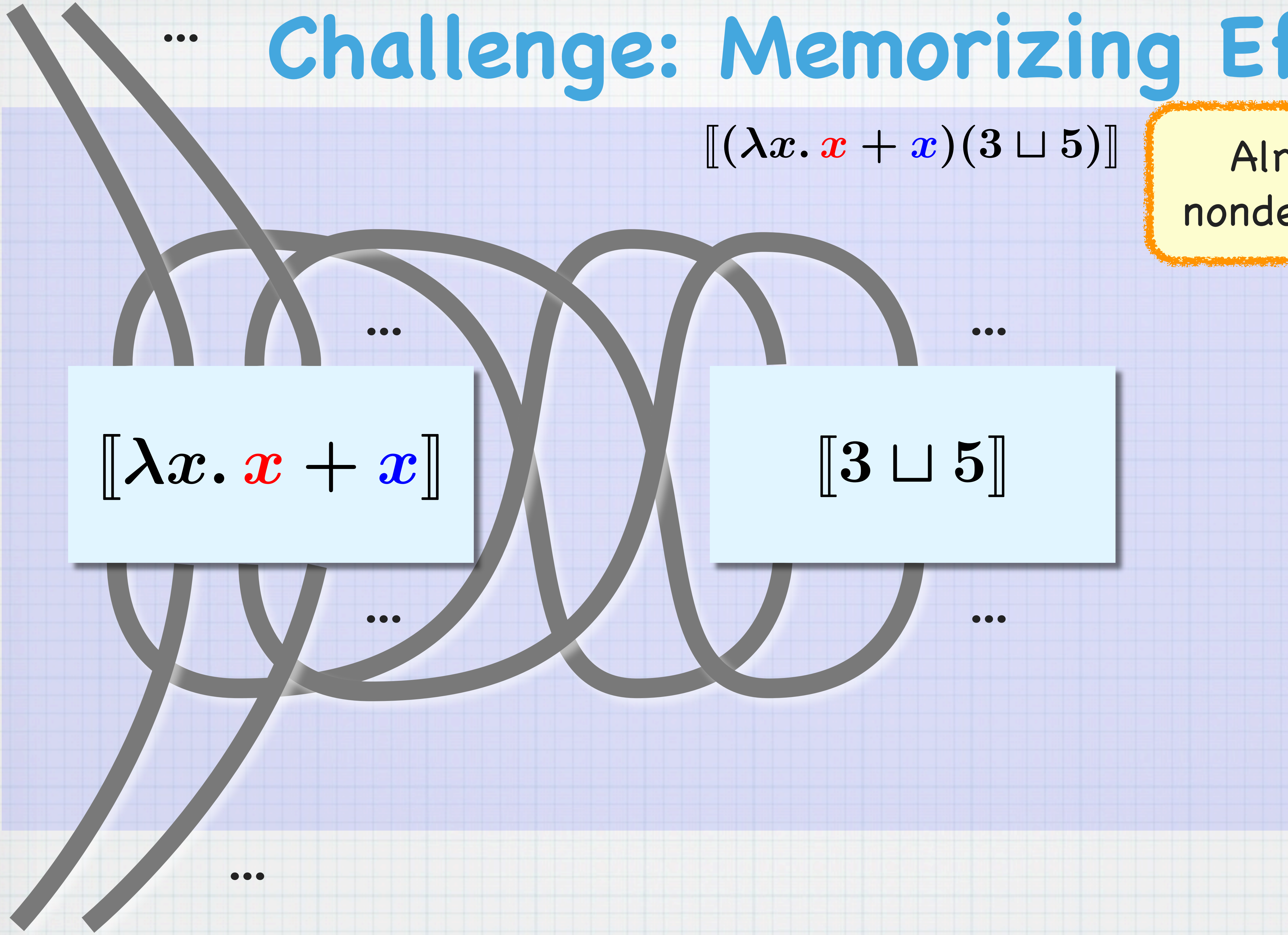
Linear category

Realizability

Challenge: Memorizing Effects

Already w/
nondeterminism!

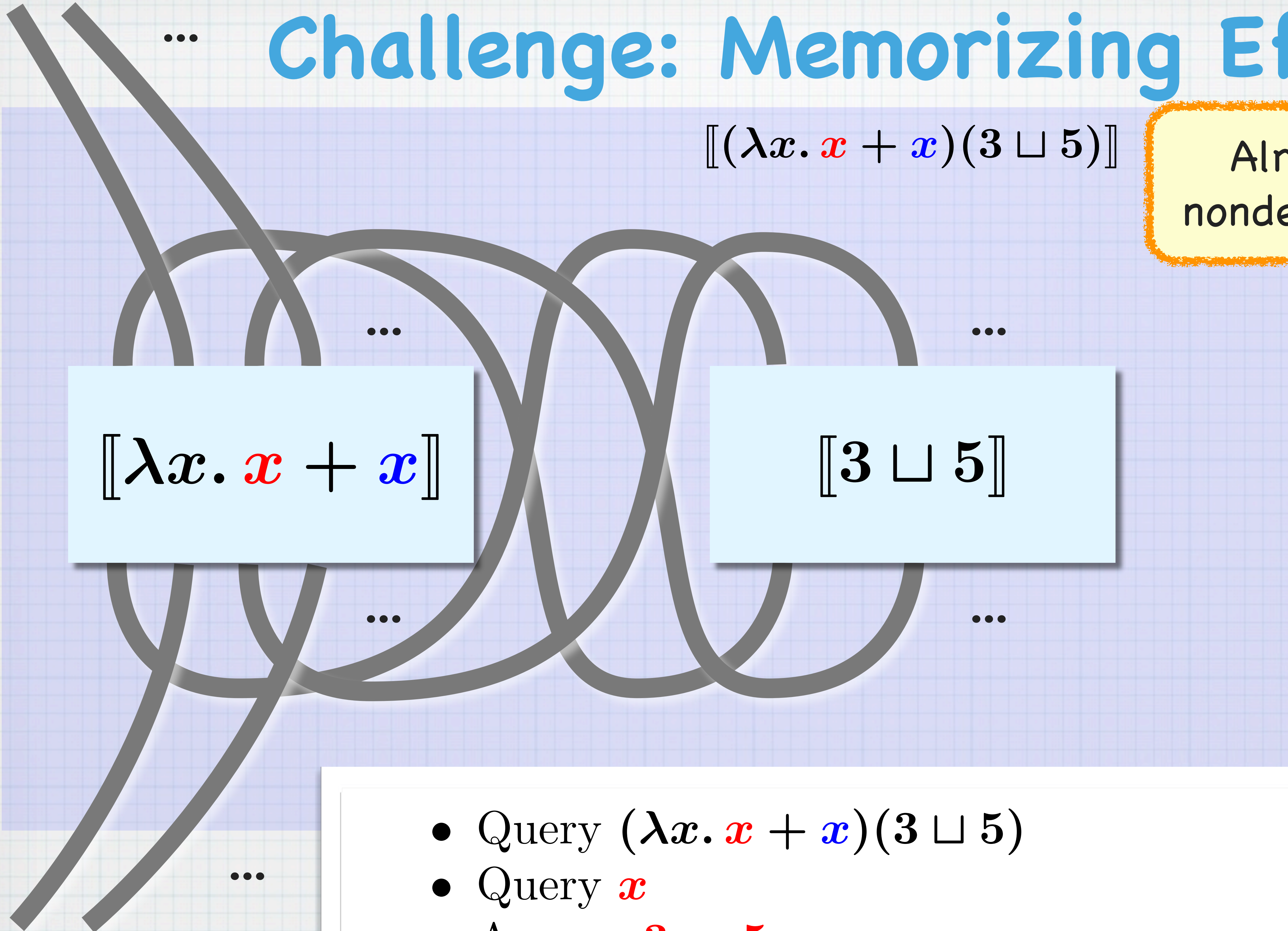
Challenge: Memorizing Effects



Already w/ nondeterminism!

Challenge: Memorizing Effects

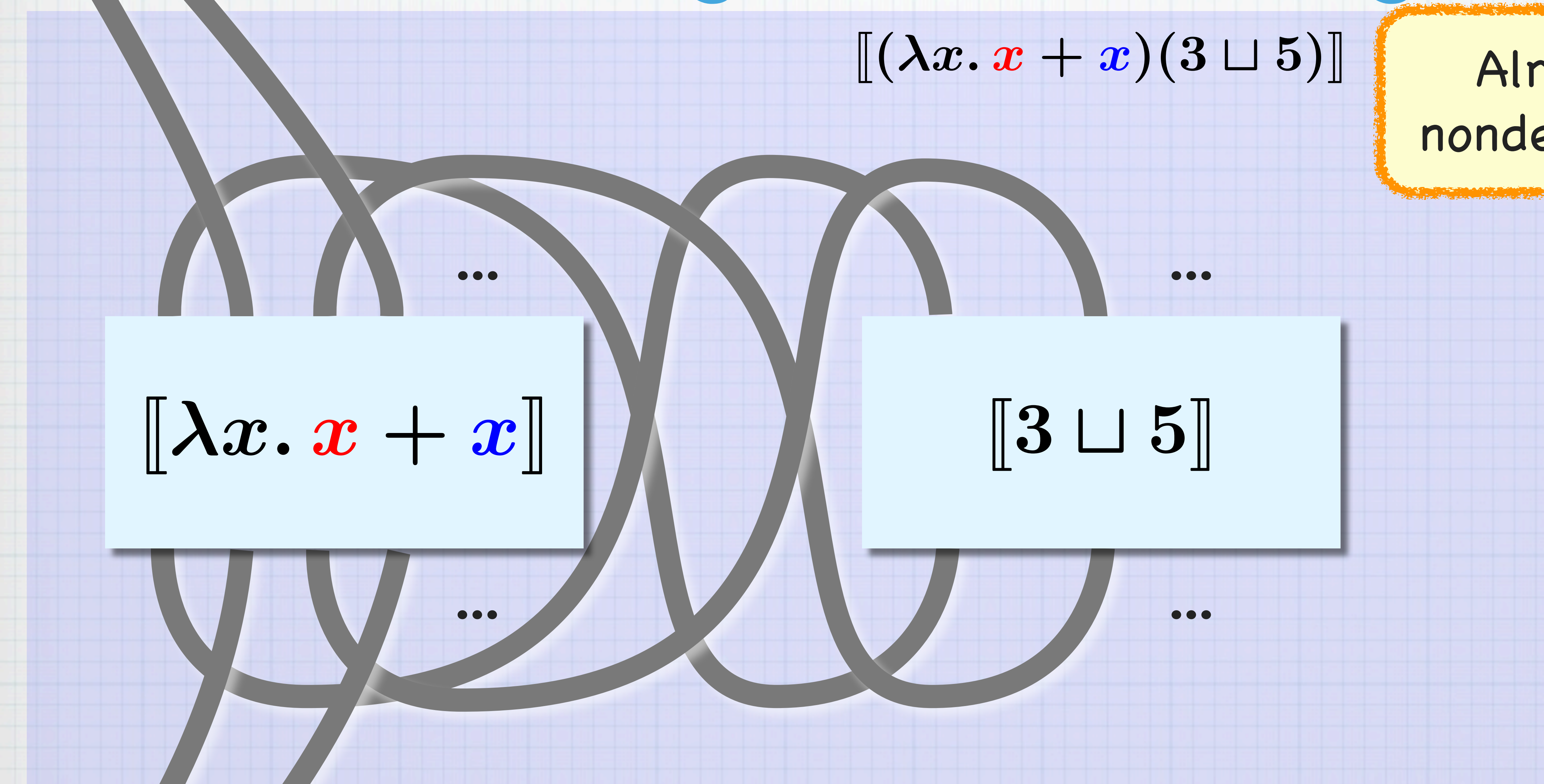
Already w/ nondeterminism!



- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer 3 or 5
- Query x
- Answer 3 or 5
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

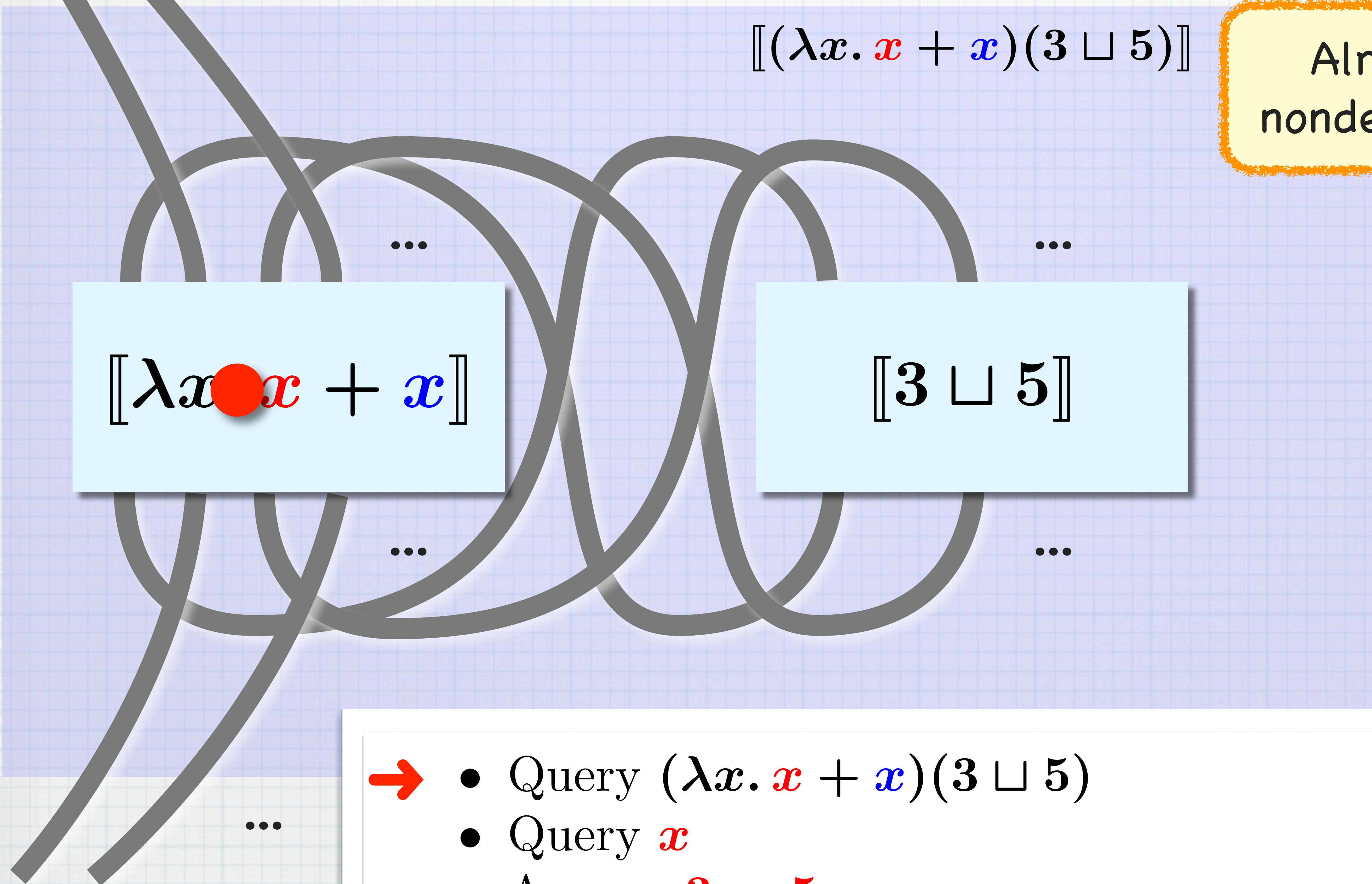
Challenge: Memorizing Effects

Already w/ nondeterminism!



- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Answer $\mathbf{3} + \mathbf{3}$, $\mathbf{3} + \mathbf{5}$, $\mathbf{5} + \mathbf{3}$ or $\mathbf{5} + \mathbf{5}$

Challenge: Memorizing Effects

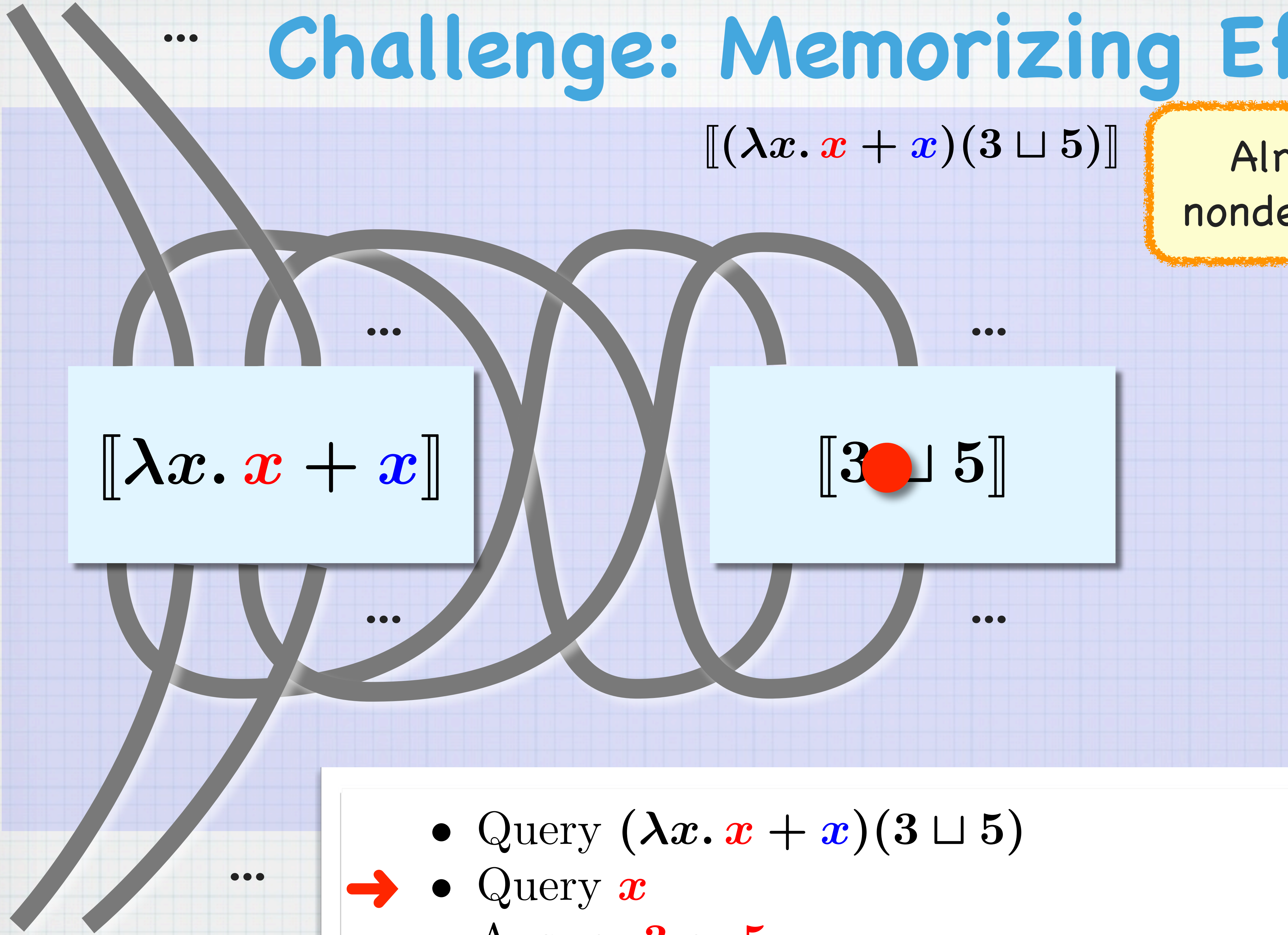


Already w/ nondeterminism!

- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Answer $\mathbf{3} + \mathbf{3}$, $\mathbf{3} + \mathbf{5}$, $\mathbf{5} + \mathbf{3}$ or $\mathbf{5} + \mathbf{5}$

Challenge: Memorizing Effects

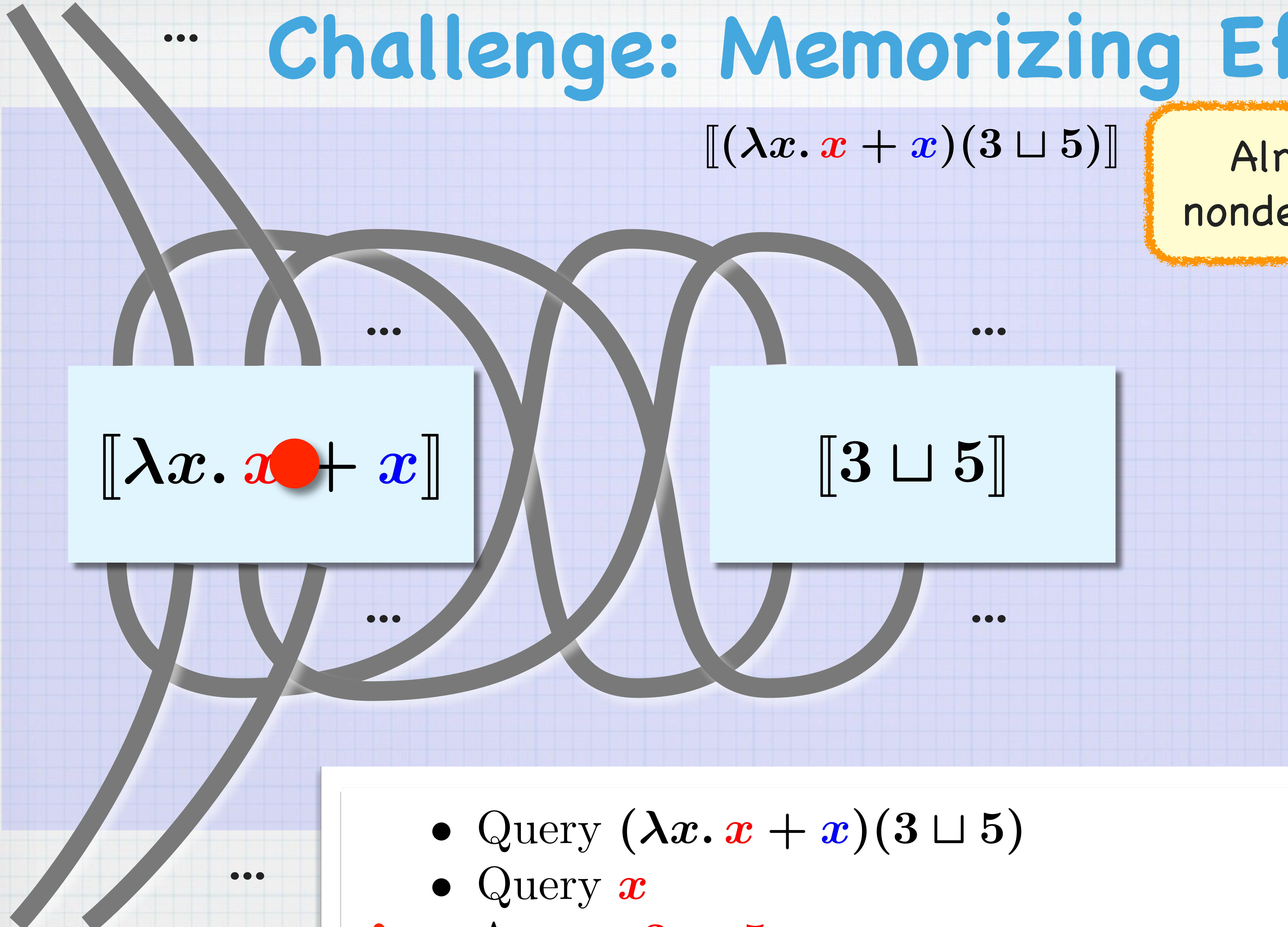
Already w/ nondeterminism!



- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- → Query x
- Answer 3 or 5
- Query x
- Answer 3 or 5
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

Challenge: Memorizing Effects

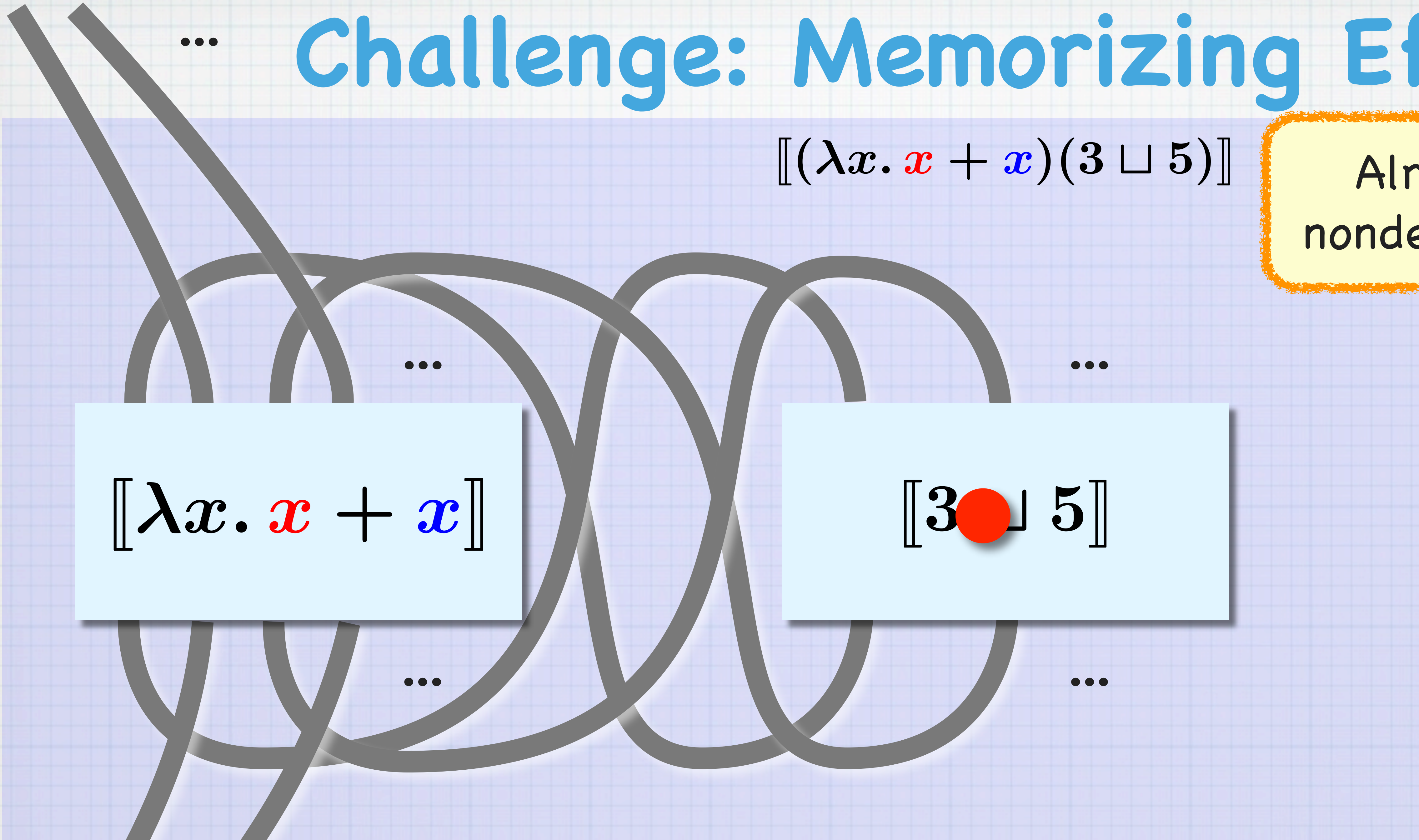
Already w/ nondeterminism!



- Query $(\lambda x.\, {\color{red}x} + {\color{blue}x})(3 \sqcup 5)$
- Query ${\color{red}x}$
- → Answer ${\color{red}3}$ or ${\color{red}5}$
- Query ${\color{blue}x}$
- Answer ${\color{blue}3}$ or ${\color{blue}5}$
- Answer ${\color{red}3} + {\color{blue}3}$, ${\color{red}3} + {\color{blue}5}$, ${\color{red}5} + {\color{blue}3}$ or ${\color{red}5} + {\color{blue}5}$

Challenge: Memorizing Effects

Already w/ nondeterminism!

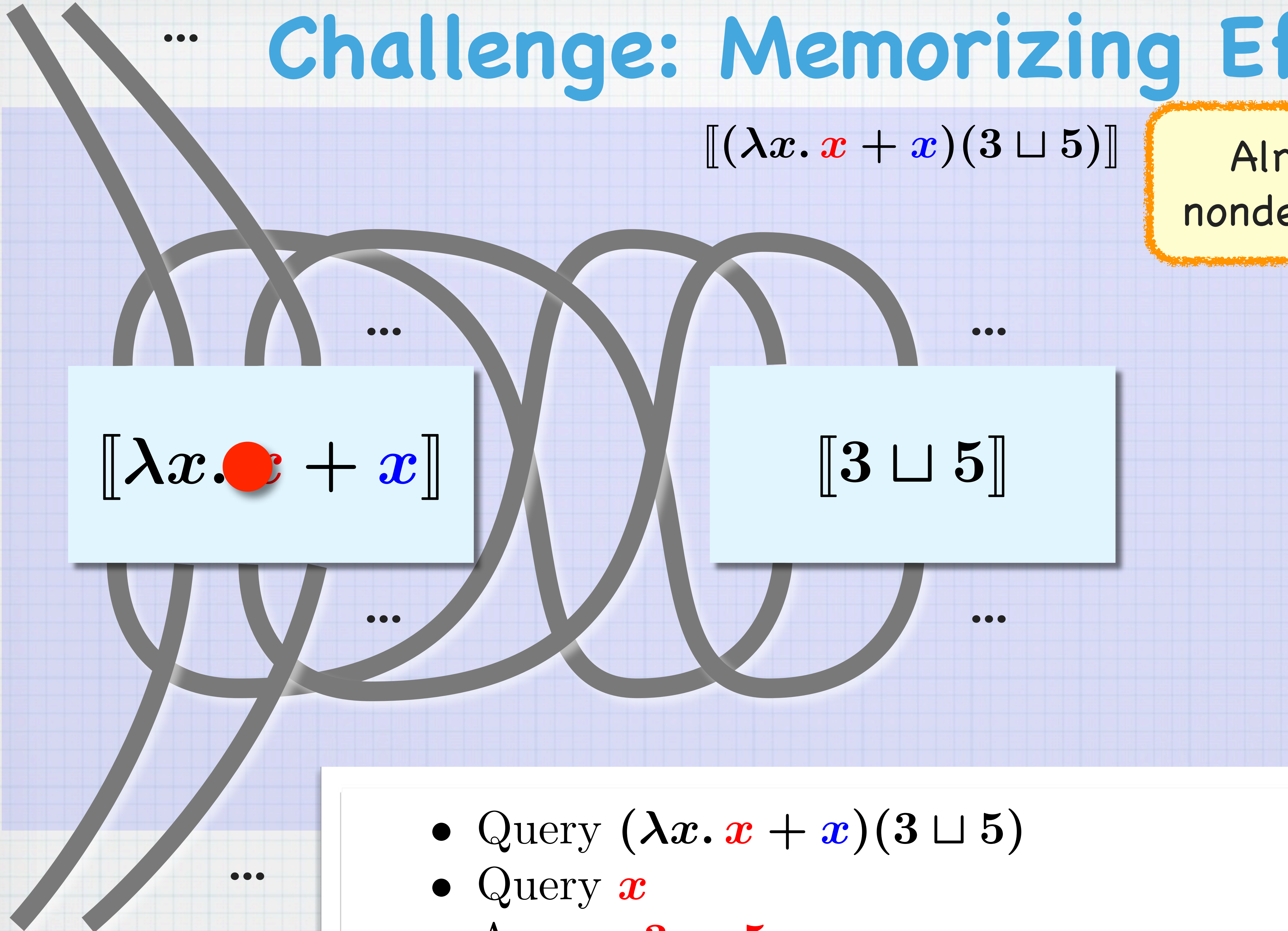


- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- ➜ Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Answer $\mathbf{3} + \mathbf{3}$, $\mathbf{3} + \mathbf{5}$, $\mathbf{5} + \mathbf{3}$ or $\mathbf{5} + \mathbf{5}$

Challenge: Memorizing Effects

$[\![(\lambda x.\, {\color{red}x} + {\color{blue}x})(3 \sqcup 5)]\!]$

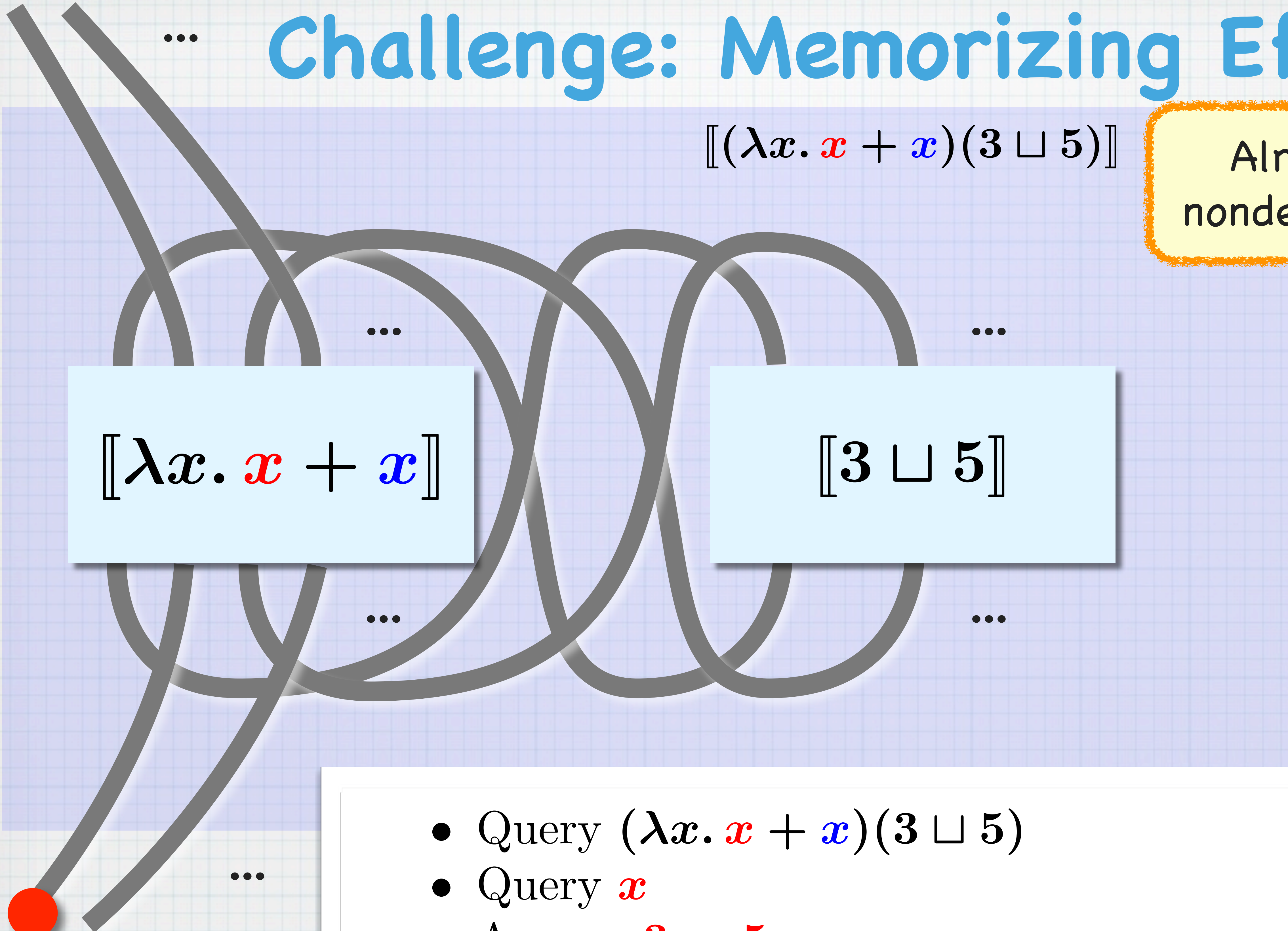
Already w/ nondeterminism!



- Query $(\lambda x.\, {\color{red}x} + {\color{blue}x})(3 \sqcup 5)$
- Query ${\color{red}x}$
- Answer ${\color{red}3}$ or ${\color{red}5}$
- Query ${\color{blue}x}$
- ➜ Answer ${\color{blue}3}$ or ${\color{blue}5}$
- Answer ${\color{red}3} + {\color{blue}3}$, ${\color{red}3} + {\color{blue}5}$, ${\color{red}5} + {\color{blue}3}$ or ${\color{red}5} + {\color{blue}5}$

Challenge: Memorizing Effects

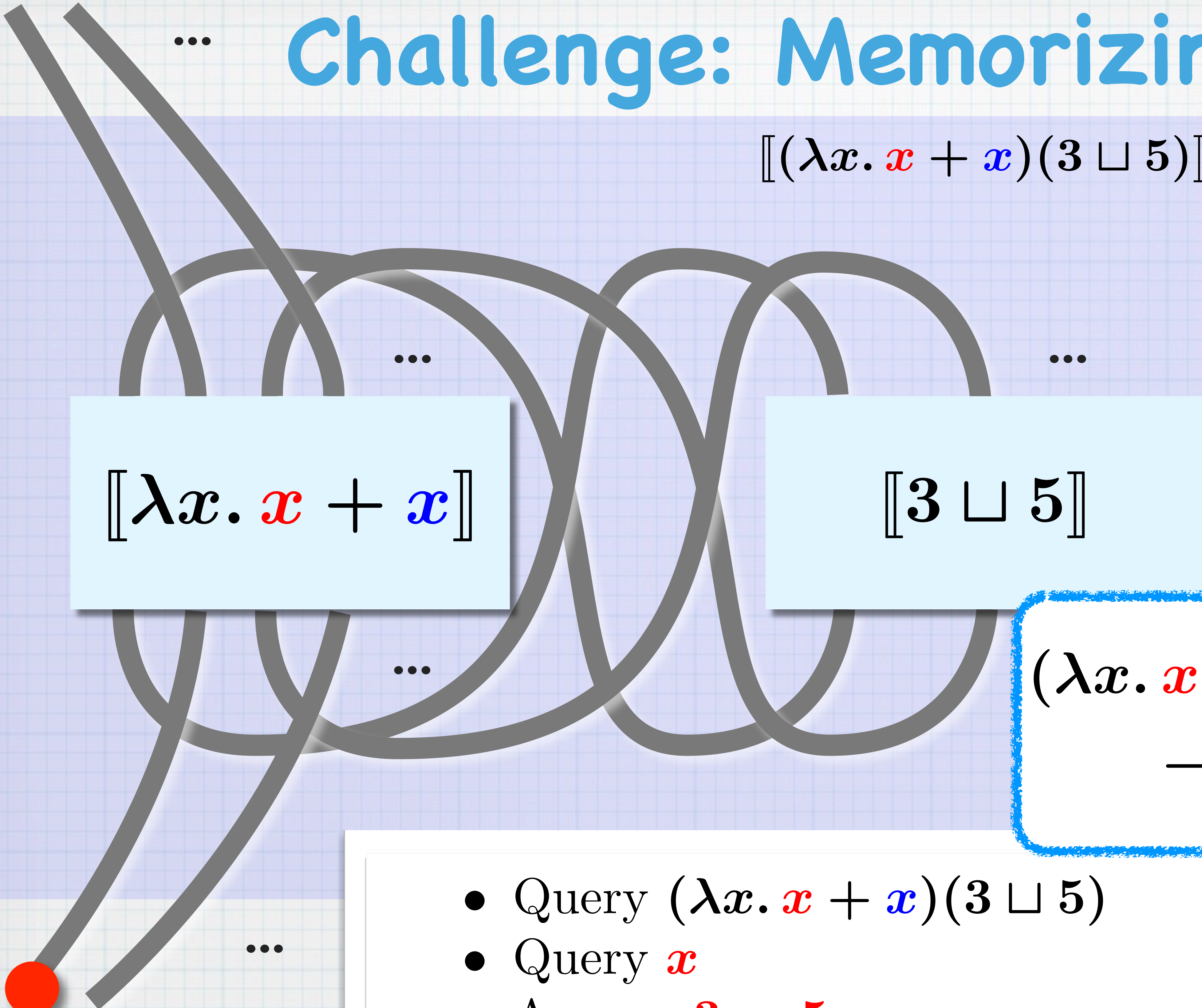
Already w/ nondeterminism!



- Query $(\lambda x.\, {\color{red}x} + {\color{blue}x})(3 \sqcup 5)$
- Query ${\color{red}x}$
- Answer ${\color{red}3}$ or ${\color{red}5}$
- Query ${\color{blue}x}$
- Answer ${\color{blue}3}$ or ${\color{blue}5}$
- Answer ${\color{red}3} + {\color{blue}3}$, ${\color{red}3} + {\color{blue}5}$, ${\color{red}5} + {\color{blue}3}$ or ${\color{red}5} + {\color{blue}5}$

Challenge: Memorizing Effects

$[\![(\lambda x.\, x + x)(3 \sqcup 5)]\!]$



Already w/ nondeterminism!

$(\lambda x.\, x + x)(3 \sqcup 5) \longrightarrow_{\mathrm{CBV}}$ **6** or **10** ??

- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer **3** or **5**
- Query x
- Answer **3** or **5**
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

Challenge: Memorizing Effects

- Nondeterministic choice is resolved
  ➜ we must **stick to it**!
- Is CBV to blame?
  (GoI is inherently CBN...)

$$(\lambda x.\, x + x)(3 \sqcup 5) \longrightarrow_{\mathrm{CBV}} \mathbf{6} \text{ or } \mathbf{10}$$

Challenge: Memorizing Effects

- Nondeterministic choice is resolved
  ➜ we must **stick to it**!
- Is CBV to blame?
  (GoI is inherently CBN...)

$$(\lambda x.\, x + x)(3 \sqcup 5) \longrightarrow_{\mathrm{CBV}} \mathbf{6} \text{ or } \mathbf{10}$$

- Not really: it's also hard to get

$$(M \sqcup N)L = ML \sqcup NL$$

Challenge: Memorizing Effects

- Nondeterministic choice is resolved
  ➜ we must **stick to it**!

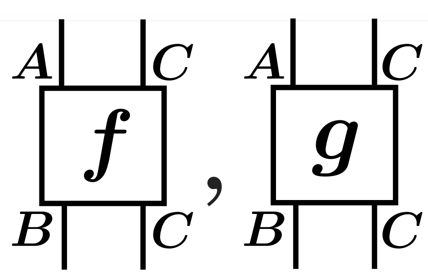
- Is CBV to blame?
  (GoI is inherently CBN...)

$$(\lambda x.\, \mathbf{x} + \mathbf{x})(3 \sqcup 5) \longrightarrow_{\text{CBV}} 6 \text{ or } 10$$

  - Not really: it's also hard to get

$$(M \sqcup N)L = ML \sqcup NL$$

  - Mathematically:

Given  $: A + C \longrightarrow \mathcal{P}(B + C)$,

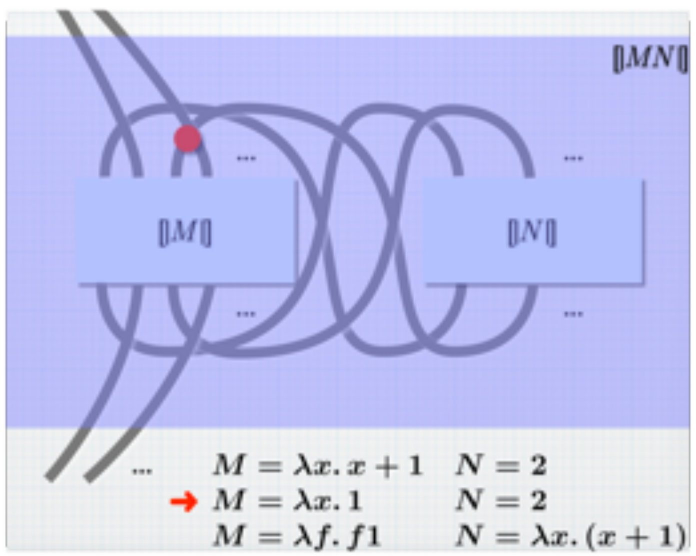
$$\mathbf{tr}(f \cup g) \neq \mathbf{tr}(f) \cup \mathbf{tr}(g)$$

Outline

Coalgebra meets **higher-order computation**

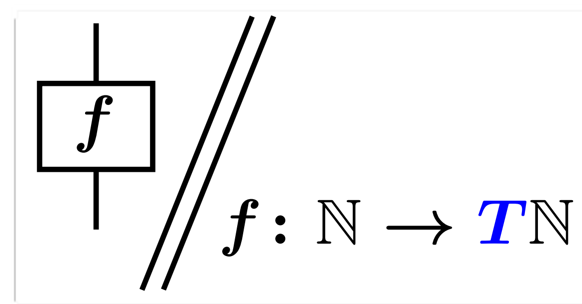
in **Geometry of Interaction** [Girard, LC'88]

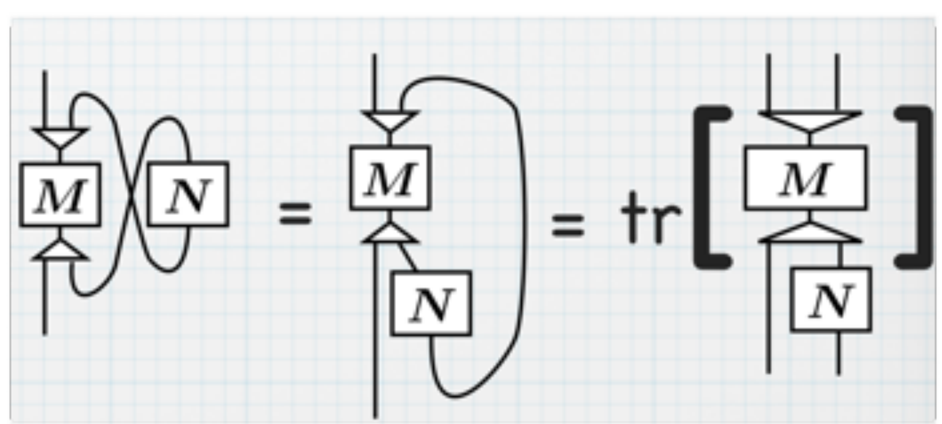
"GoI Animation"



GoI w/ T-branching

[IH & Hoshino, LICS'11]





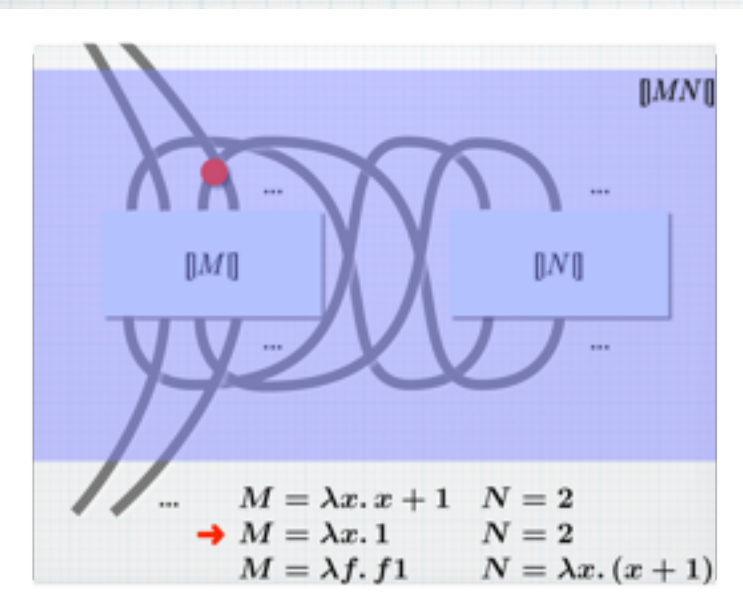
Categorical GoI

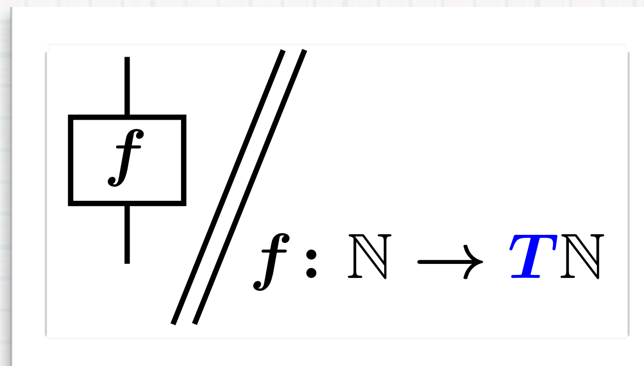
[Abramsky, Haghverdi & Scott, MSCS'02]

Outline

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

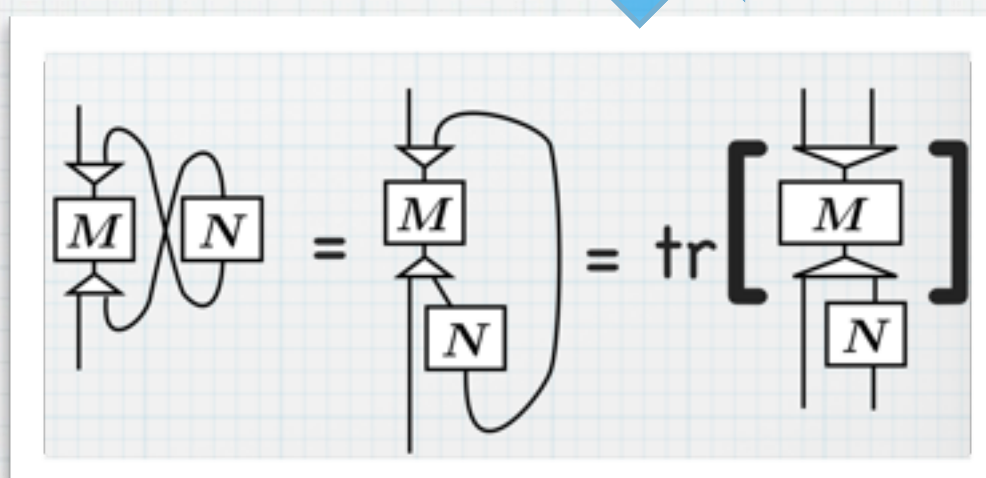
"GoI Animation"





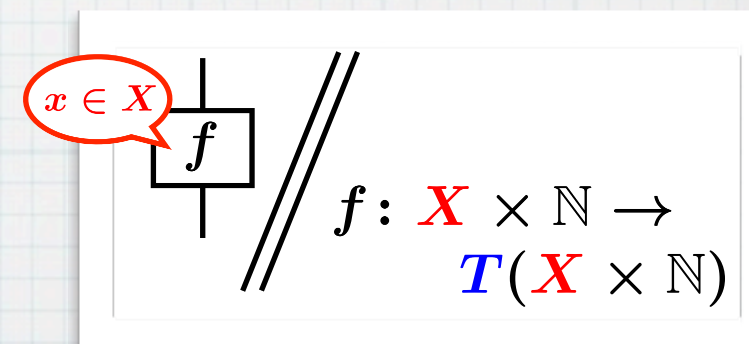
GoI w/ T-branching

[IH & Hoshino, LICS'11]



Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]

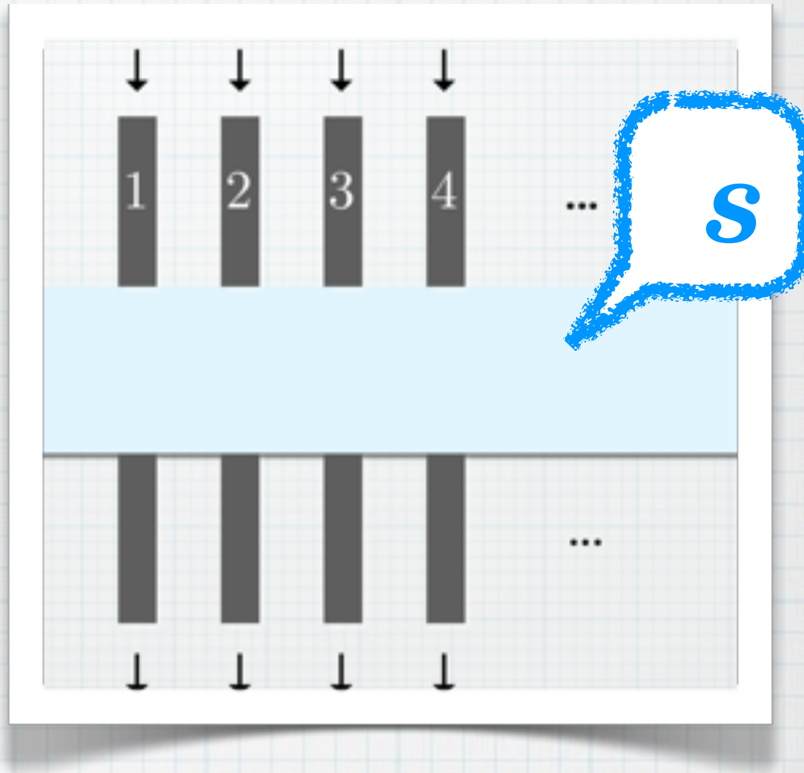


Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

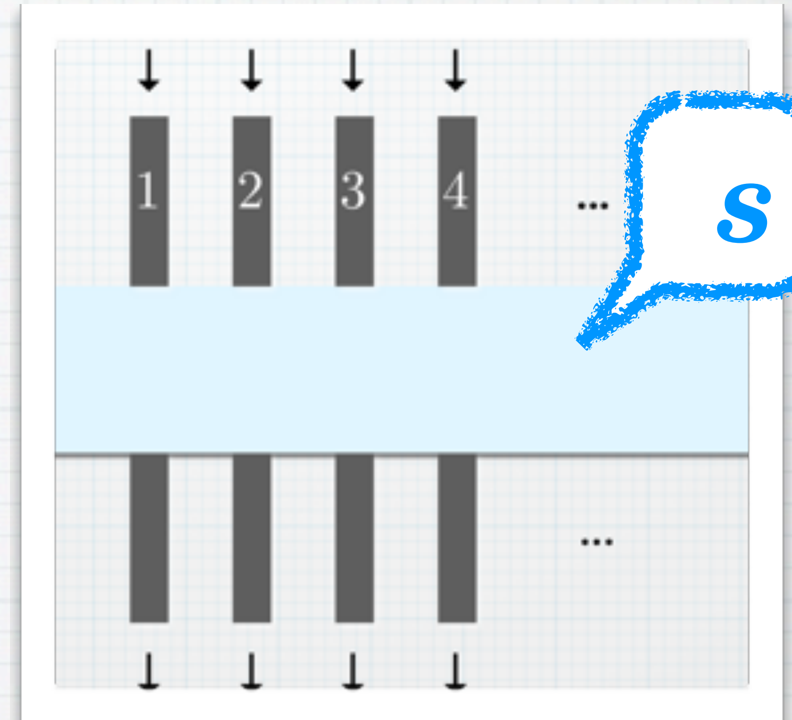
Memoryful GoI

* Equip piping with internal states, or **memory**



Hasuo (Tokyo)

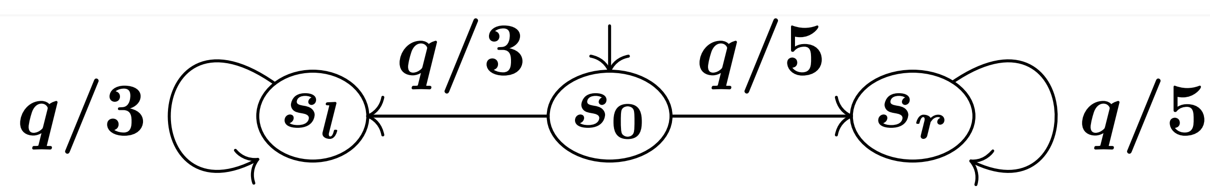
Memoryful GoI



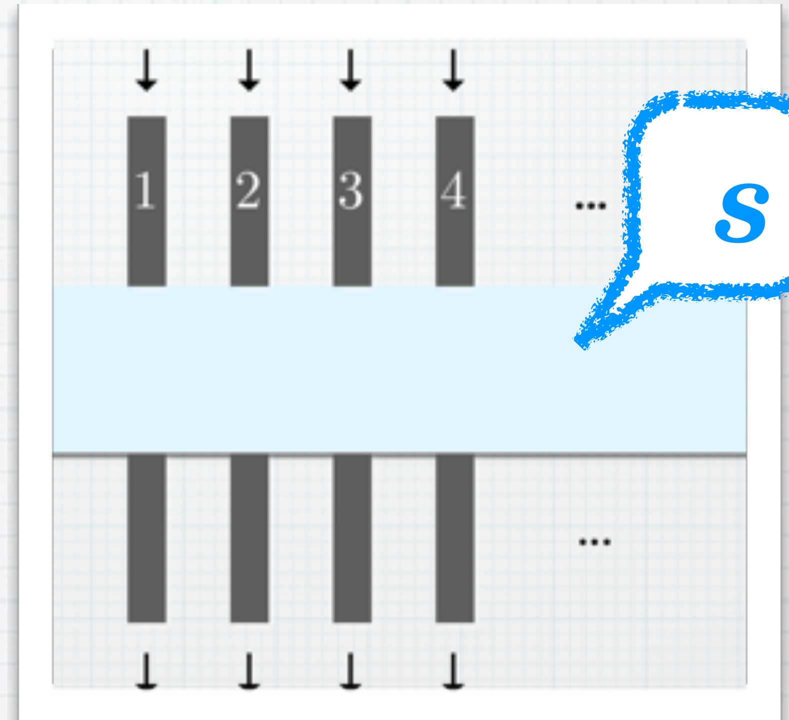
* Equip piping with internal states, or **memory**
* **not** $[\![3 \sqcup 5]\!] : \mathbb{N} \longrightarrow \mathcal{P}\mathbb{N} \ , \quad q \longmapsto \{3,5\}$

but a transducer (Mealy machine)

$[\![3 \sqcup 5]\!] : X \times \mathbb{N} \longrightarrow \mathcal{P}(X \times \mathbb{N}) \ ,$



Memoryful GoI



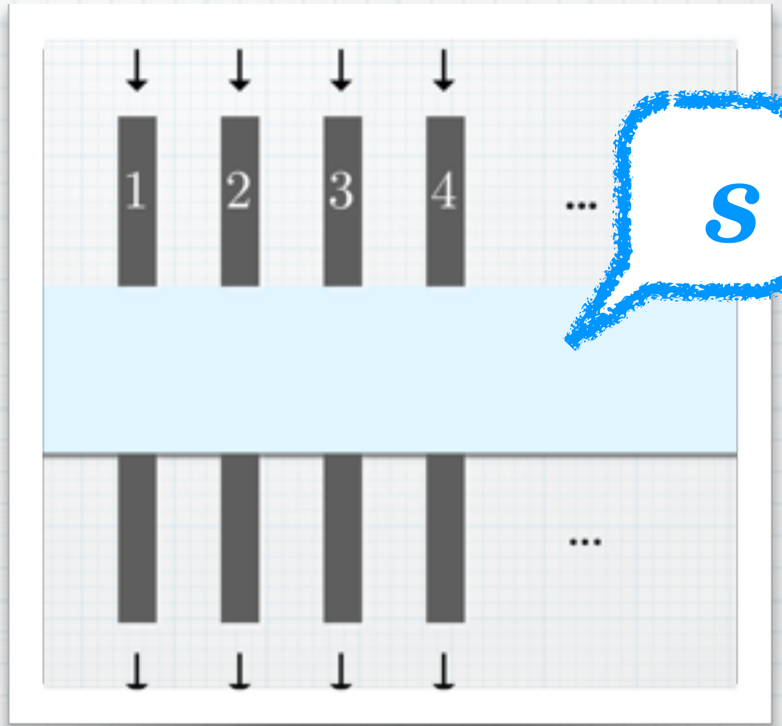
- Equip piping with internal states, or **memory**
- **not** $[\![3 \sqcup 5]\!] : \mathbb{N} \longrightarrow \mathcal{P}\mathbb{N} \ , \quad q \longmapsto \{3, 5\}$

  but a transducer (Mealy machine)

$$[\![3 \sqcup 5]\!] : \boldsymbol{X} \times \mathbb{N} \longrightarrow \mathcal{P}(\boldsymbol{X} \times \mathbb{N}) \ ,$$

$$\begin{tikzcd}
s_l \arrow[loop left, "q/3"] & s_0 \arrow[l, "q/3"'] \arrow[r, "q/5"] & s_r \arrow[loop right, "q/5"]
\end{tikzcd}$$

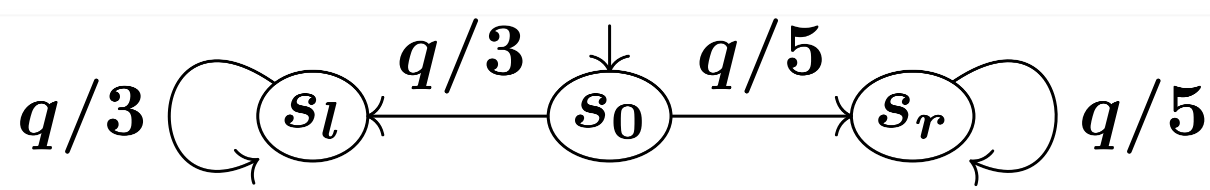
Memoryful GoI



- Equip piping with internal states, or **memory**
- **not** $[\![3 \sqcup 5]\!] : \mathbb{N} \longrightarrow \mathcal{P}\mathbb{N} \;,\quad q \longmapsto \{3,5\}$

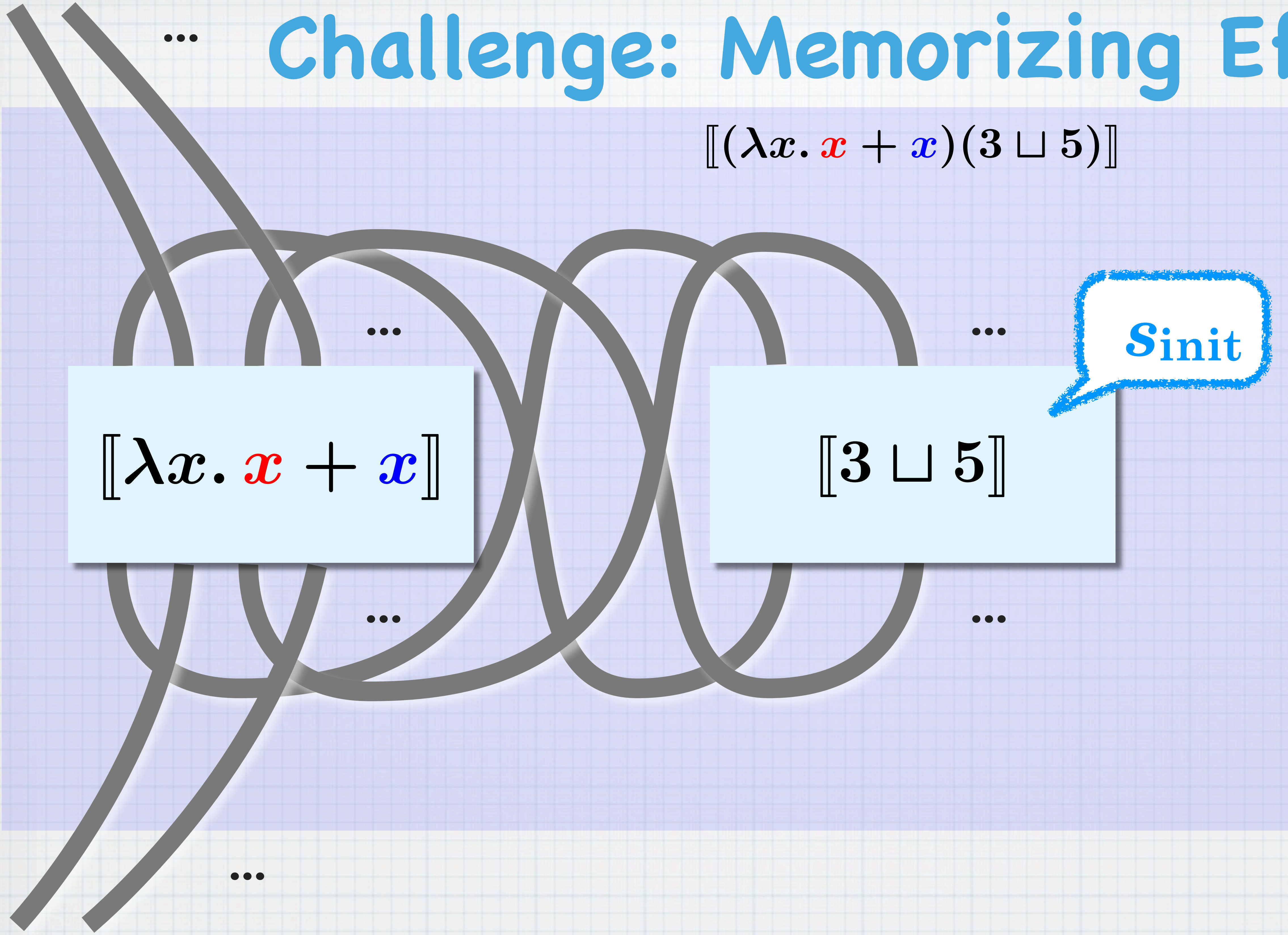
  but a transducer (Mealy machine)

  $[\![3 \sqcup 5]\!] : X \times \mathbb{N} \longrightarrow \mathcal{P}(X \times \mathbb{N}) \;,$

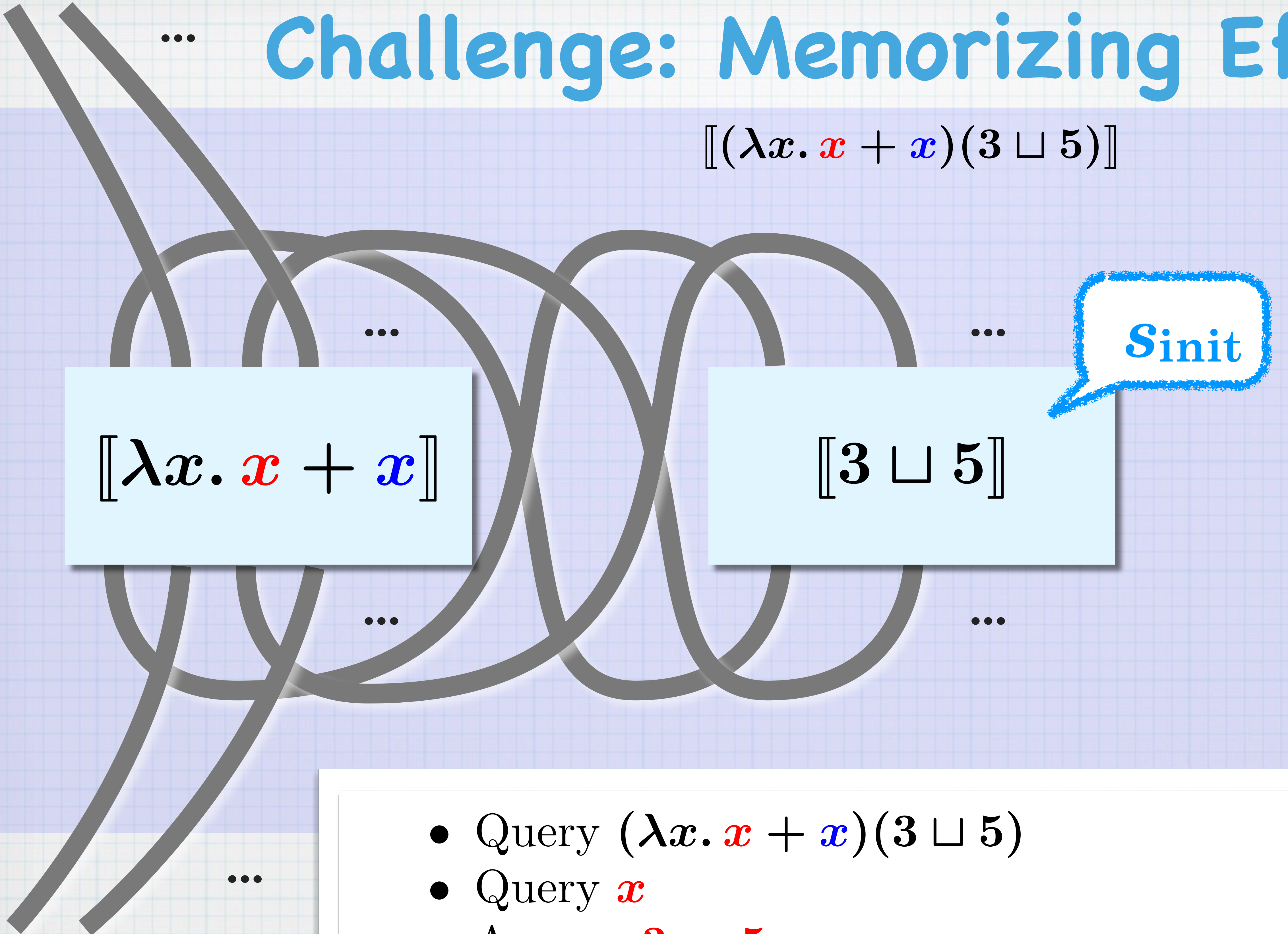


- Not a new idea:
  - Slices in GoI for additives [Laurent, TLCA'01]
  - Resumption GoI [Abramsky, CONCUR'96]

Challenge: Memorizing Effects

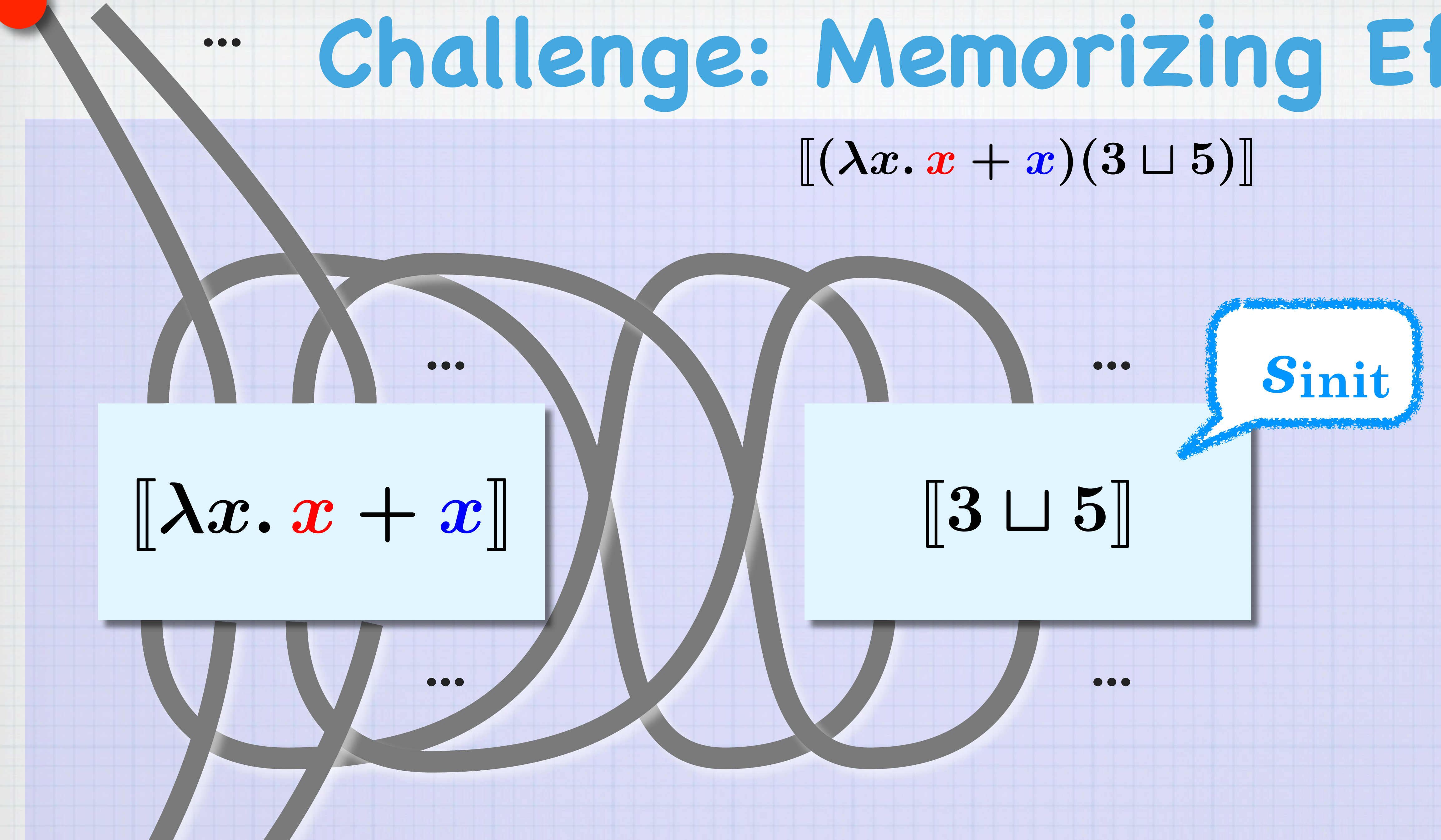


Challenge: Memorizing Effects



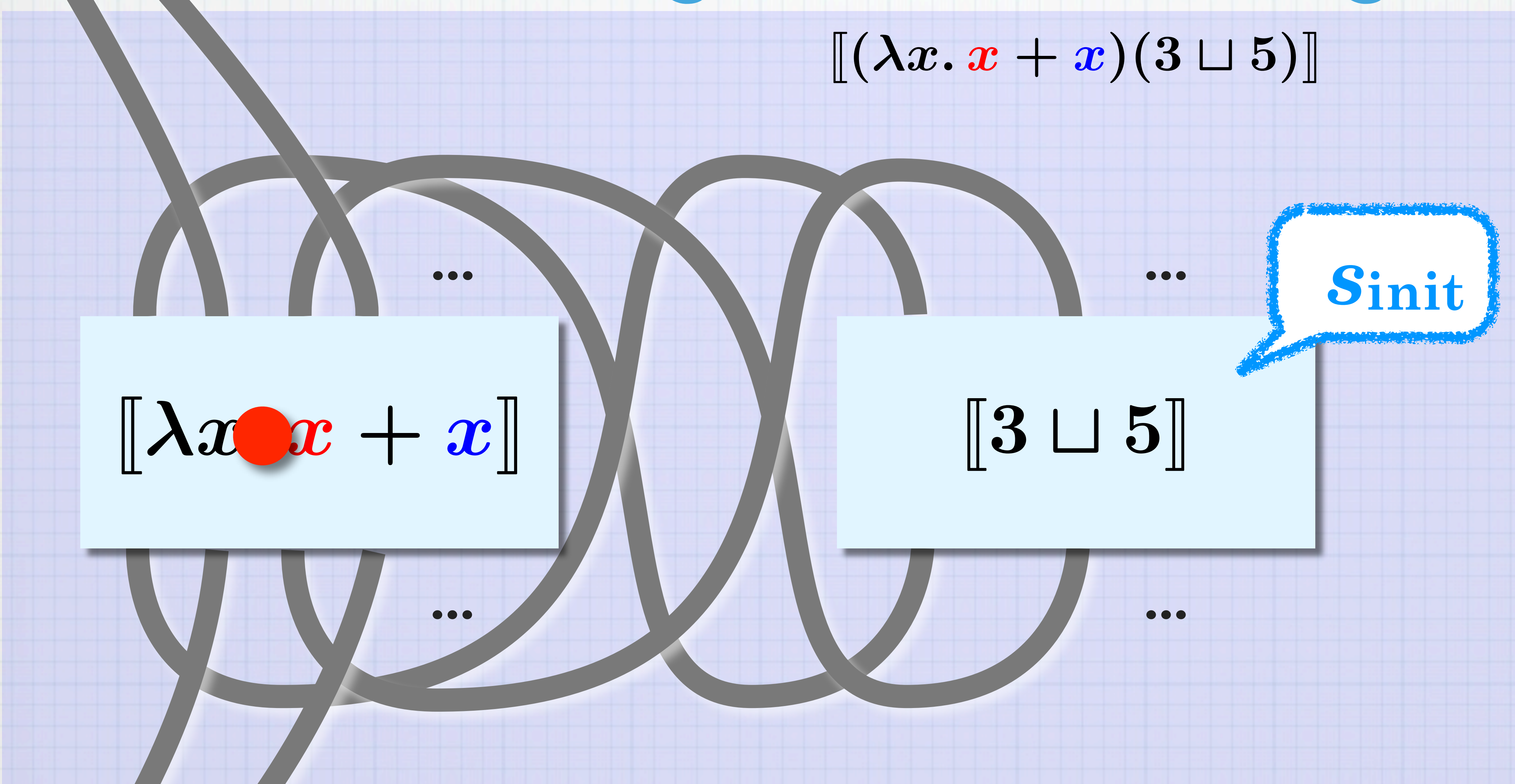
- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Answer $\mathbf{3} + \mathbf{3}$, $\mathbf{3} + \mathbf{5}$, $\mathbf{5} + \mathbf{3}$ or $\mathbf{5} + \mathbf{5}$

Challenge: Memorizing Effects



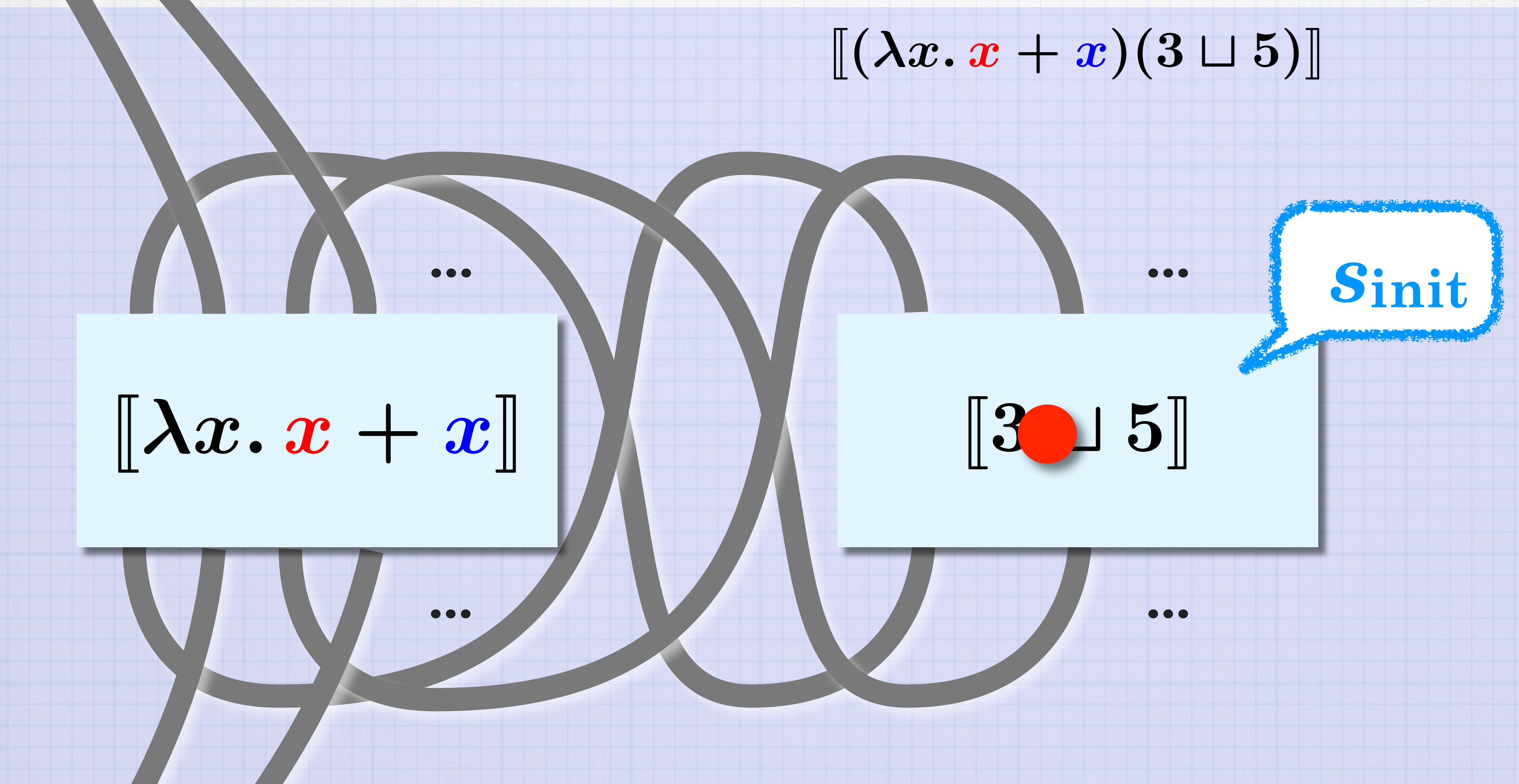
- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer 3 or 5
- Query x
- Answer 3 or 5
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

Challenge: Memorizing Effects



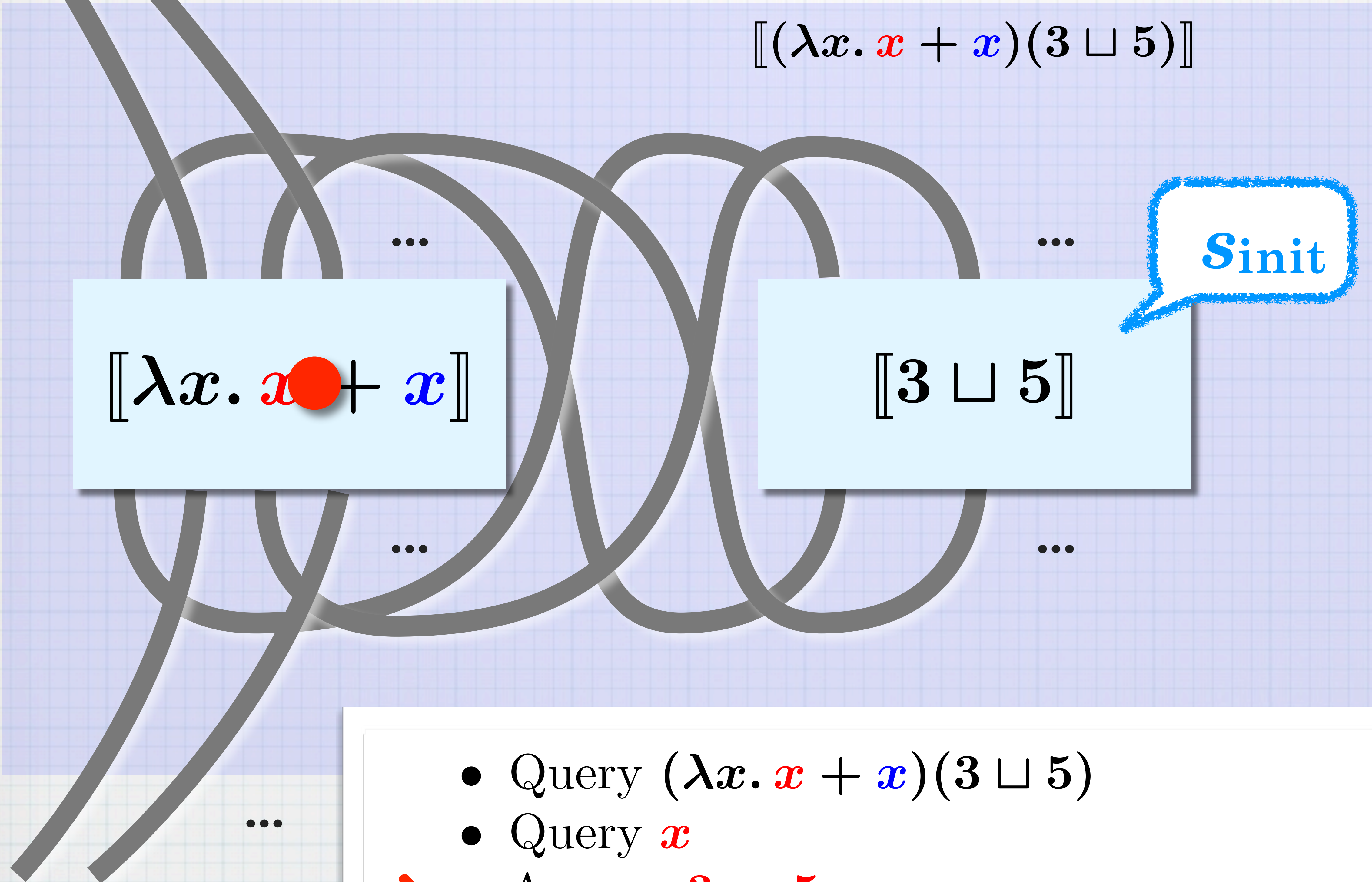
- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer 3 or 5
- Query x
- Answer 3 or 5
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

Challenge: Memorizing Effects



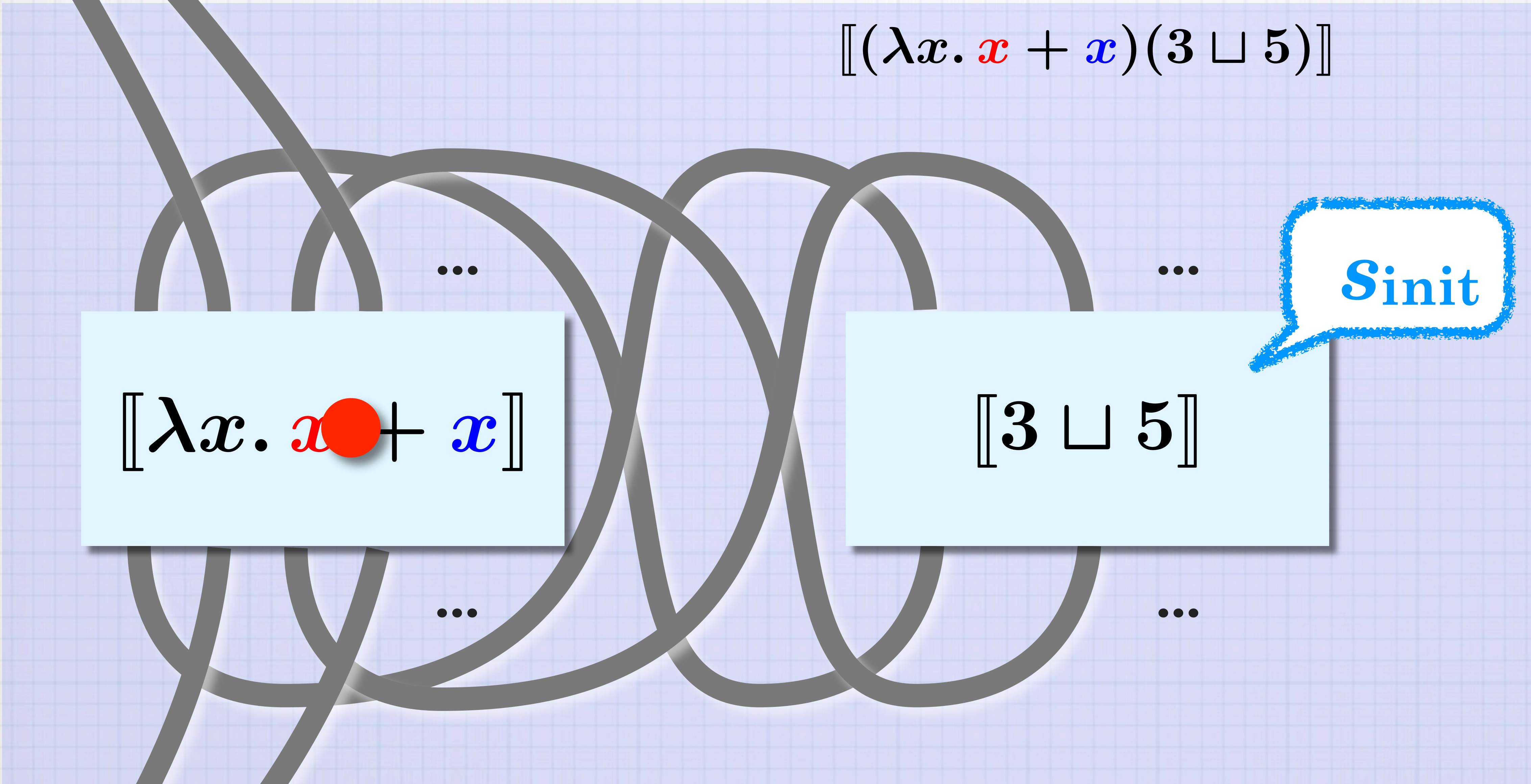
- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- ➔ Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Answer $\mathbf{3} + \mathbf{3}$, $\mathbf{3} + \mathbf{5}$, $\mathbf{5} + \mathbf{3}$ or $\mathbf{5} + \mathbf{5}$

Challenge: Memorizing Effects



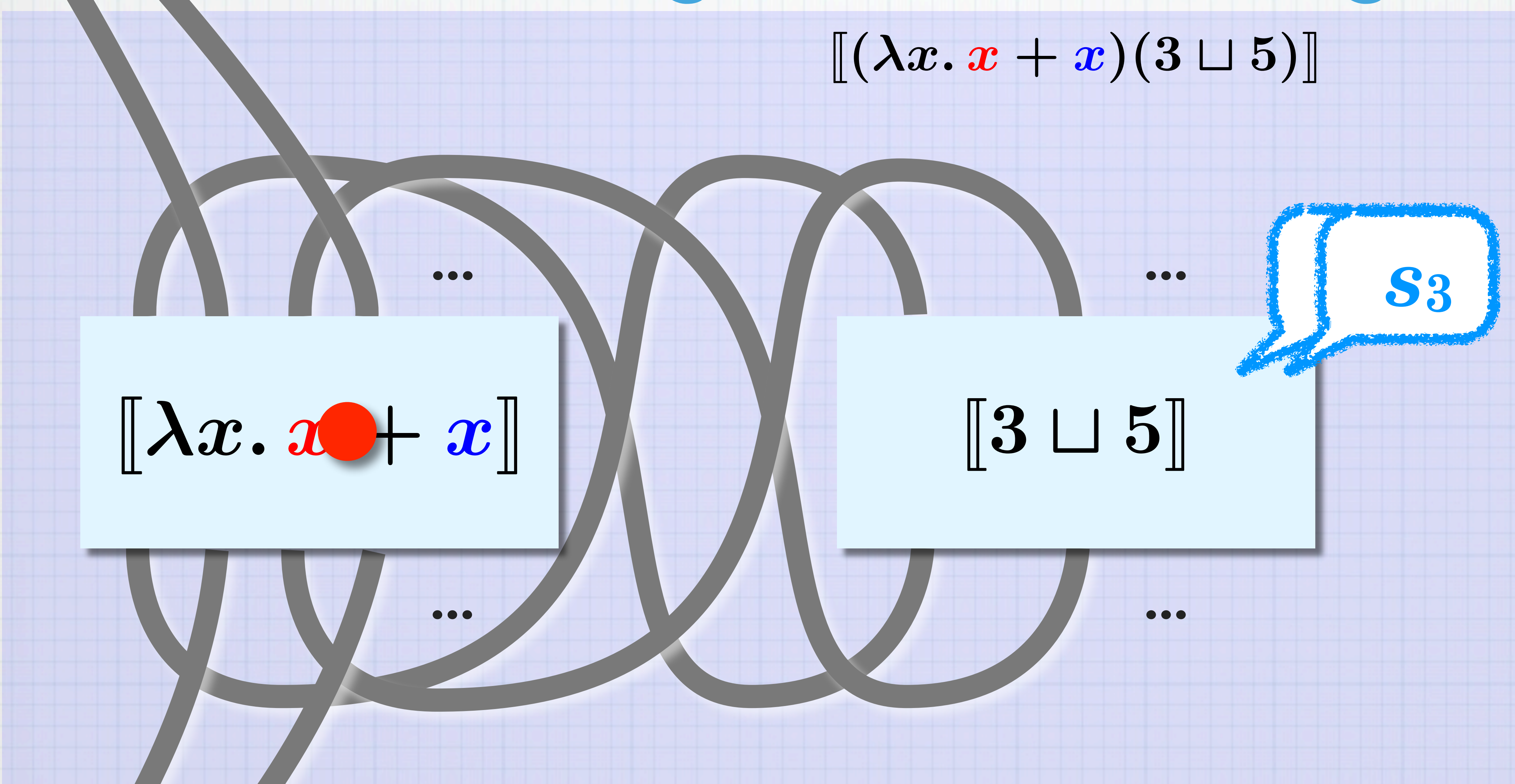
- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- ➜ Answer 3 or 5
- Query x
- Answer 3 or 5
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

Challenge: Memorizing Effects



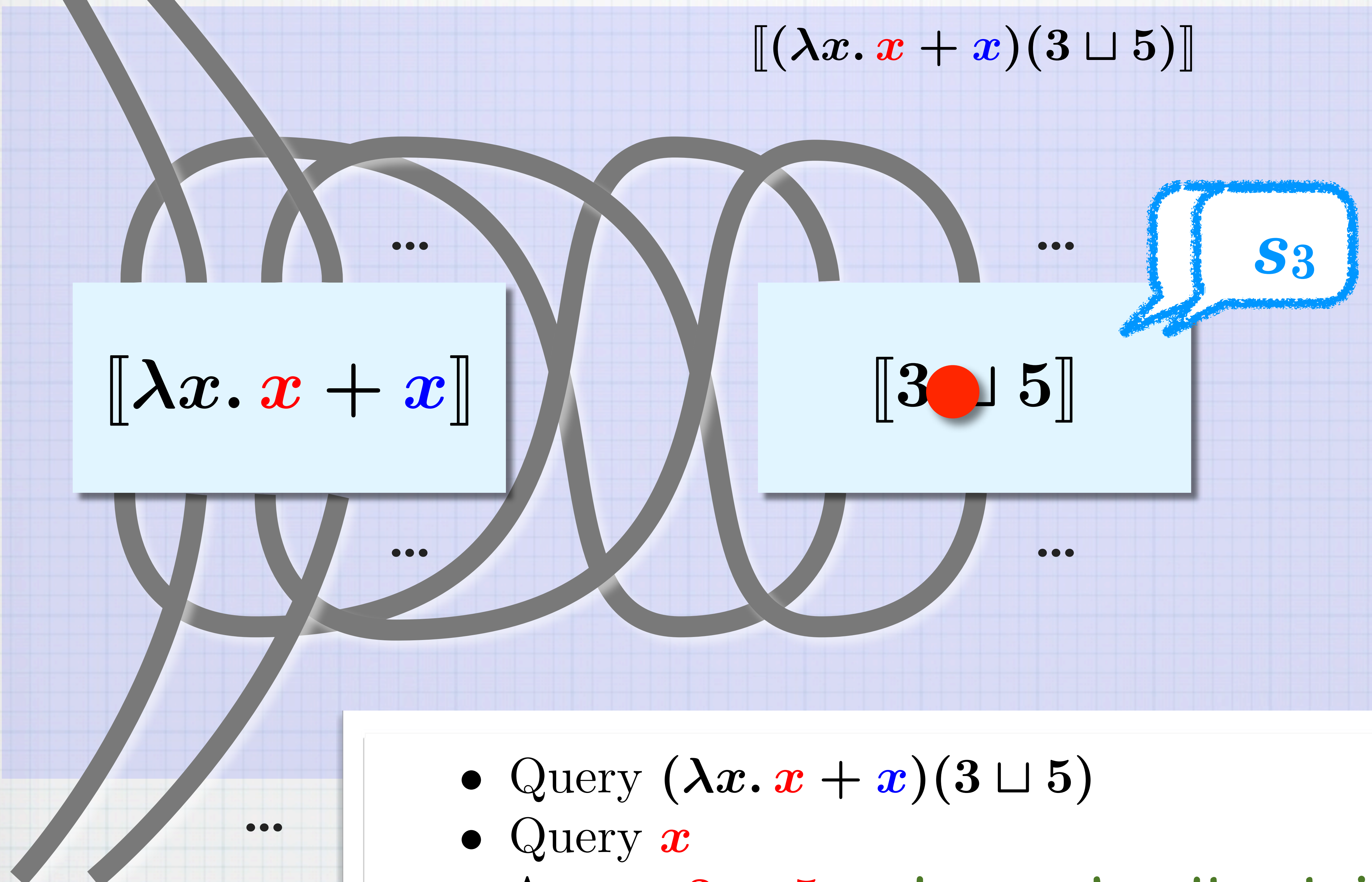
- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer 3 or 5 <u>**and remember the choice**</u>
- Query x
- Answer 3 or 5
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

Challenge: Memorizing Effects



- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- ➜ Answer 3 or 5 <u>and remember the choice</u>
- Query x
- Answer 3 or 5
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

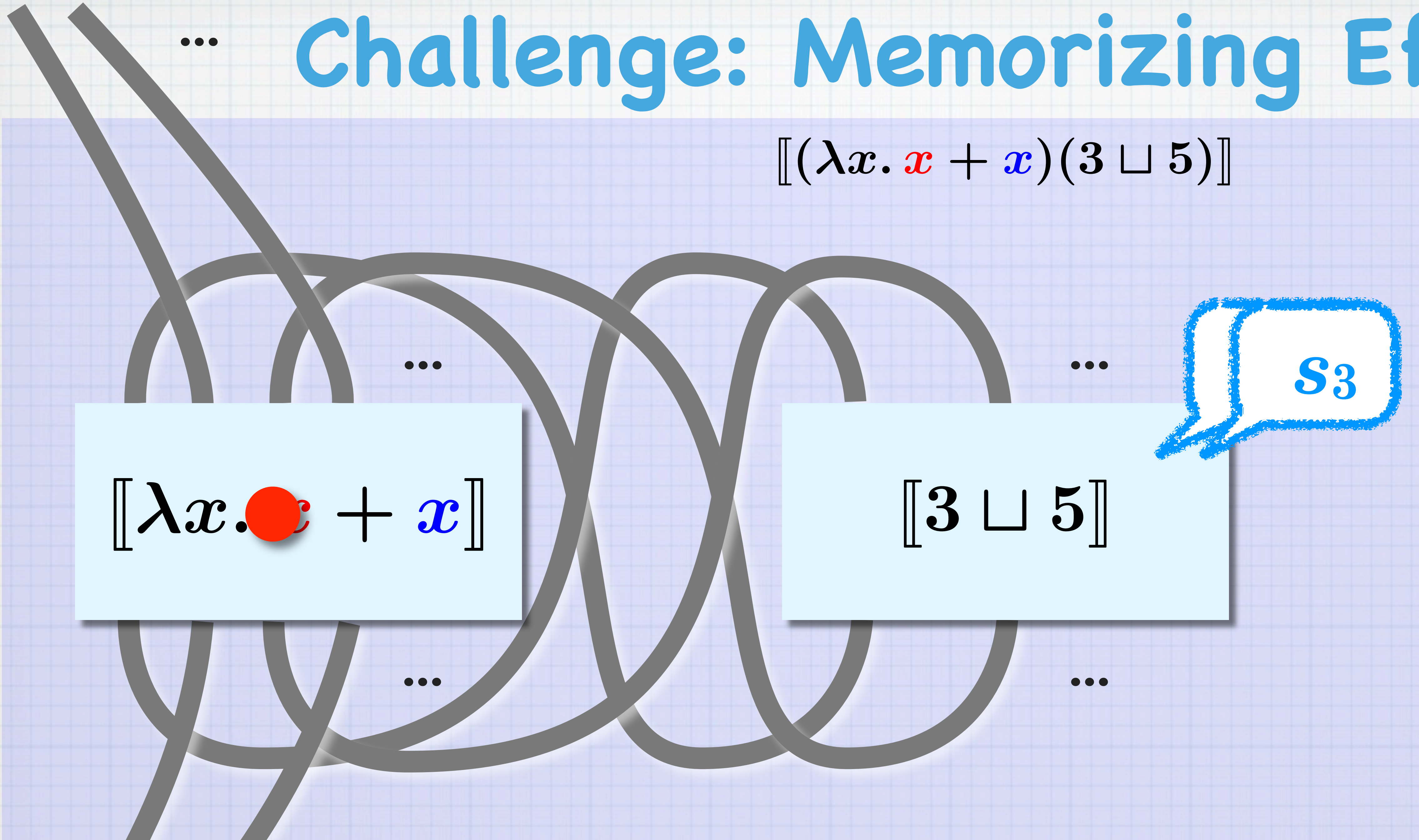
Challenge: Memorizing Effects



- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$ and remember the choice
- → Query x
- Answer $\mathbf{3}$ or $\mathbf{5}$
- Answer $\mathbf{3} + \mathbf{3}$, $\mathbf{3} + \mathbf{5}$, $\mathbf{5} + \mathbf{3}$ or $\mathbf{5} + \mathbf{5}$

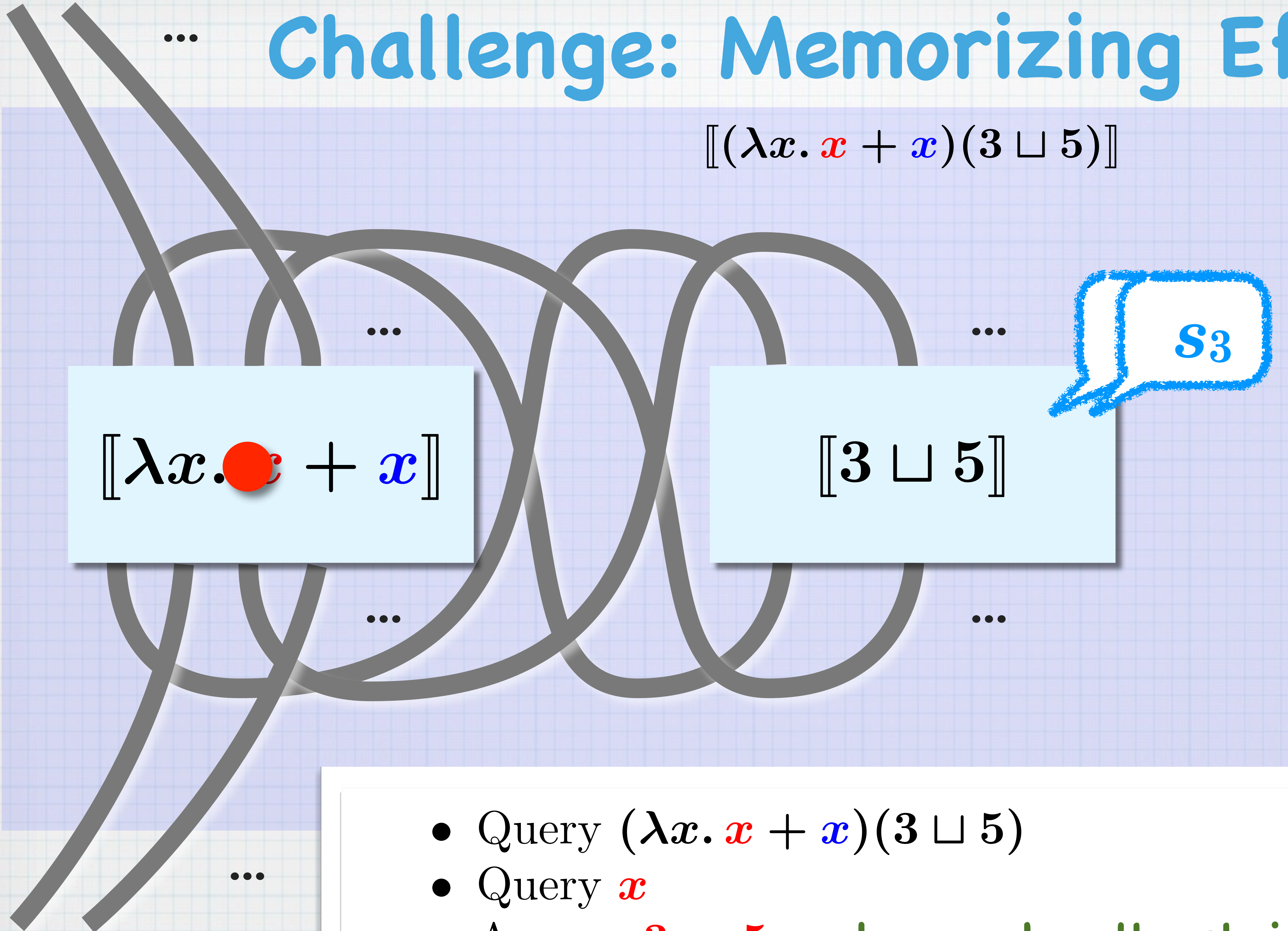
Challenge: Memorizing Effects

$$[\![(\lambda x.\, \textcolor{red}{x} + \textcolor{blue}{x})(3 \sqcup 5)]\!]$$



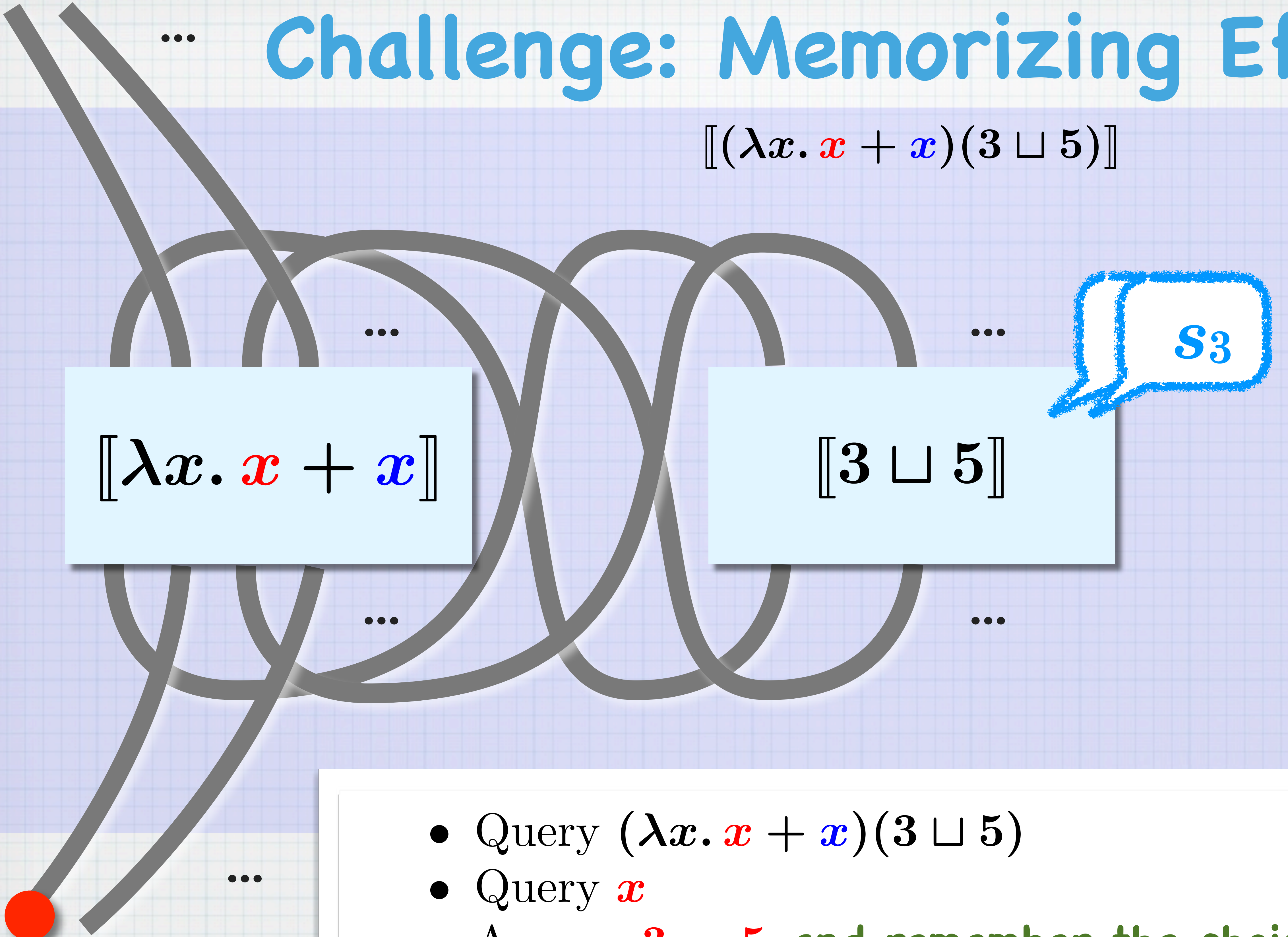
- Query $(\lambda x.\, \textcolor{red}{x} + \textcolor{blue}{x})(3 \sqcup 5)$
- Query $\textcolor{red}{x}$
- Answer $\textcolor{red}{3}$ or $\textcolor{red}{5}$ <u>and remember the choice</u>
- Query $\textcolor{blue}{x}$
- ➜ Answer $\textcolor{blue}{3}$ or $\textcolor{blue}{5}$
- Answer $\textcolor{red}{3} + \textcolor{blue}{3}$, $\textcolor{red}{3} + \textcolor{blue}{5}$, $\textcolor{red}{5} + \textcolor{blue}{3}$ or $\textcolor{red}{5} + \textcolor{blue}{5}$

Challenge: Memorizing Effects



- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer 3 or 5 and remember the choice
- Query x
- → Answer 3 or 5 following the prev. choice
- Answer $3 + 3$, $3 + 5$, $5 + 3$ or $5 + 5$

Challenge: Memorizing Effects



- Query $(\lambda x.\, x + x)(3 \sqcup 5)$
- Query x
- Answer 3 or 5 <u>and remember the choice</u>
- Query x
- Answer 3 or 5 <u>following the prev. choice</u>
- Answer $3 + 3$, $\cancel{3 + 5}$, $\cancel{5 + 3}$ or $5 + 5$

Memoryful GoI

- That is...
  a traversing token rearranges piping!

Memoryful GoI

* We introduce memory in a structured manner...
  ➜
  the "traced monoidal category" of transducers

$\boxed{\mathbf{Trans}(T)}$	Objects:	sets $A, B, \ldots$
	Arrows:	$$\dfrac{A \longrightarrow B \text{ in } \mathbf{Trans}(T)}{\big(\, X,\ X \times A \xrightarrow{c} T(X \times B),\ x_0 \in X\ \big),\ T\text{-transducer}}$$

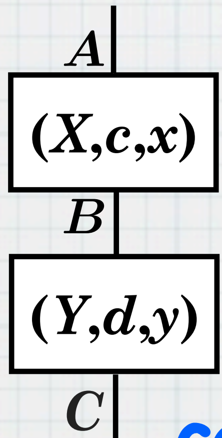
Memoryful GoI

- We introduce memory in a structured manner...
  ➜
  the "traced monoidal category" of transducers

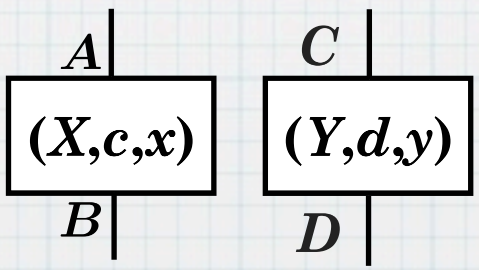
Trans(T) Objects: sets $A, B, \ldots$

Arrows:
$$\frac{A \longrightarrow B \text{ in } \mathbf{Trans}(T)}{\left(X,\ X \times A \xrightarrow{c} T(X \times B),\ x_0 \in X\ \right),\ T\text{-transducer}}$$

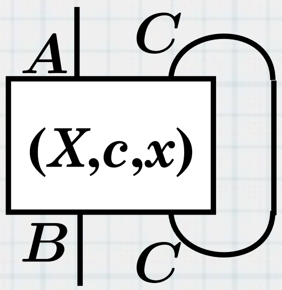
- with operations like



composition $\circ$



tensor $\otimes$



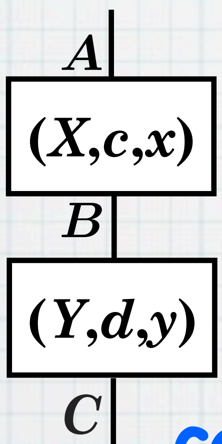
trace

Trans(T) by Coalgebraic Component Calculus

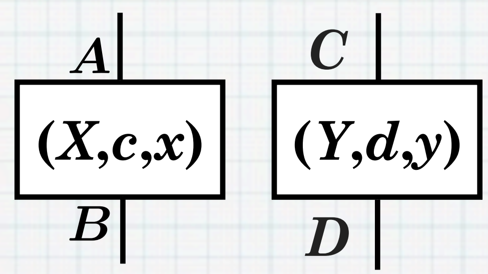
[Barbosa '03][IH & Jacobs '11]

$\boxed{\mathbf{Trans}(T)}$ Objects: sets $A, B, \ldots$

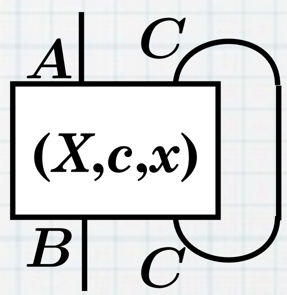
Arrows:
$$\frac{A \longrightarrow B \text{ in } \mathbf{Trans}(T)}{\left(X,\ X \times A \xrightarrow{c} T(X \times B),\ x_0 \in X\ \right),\ T\text{-transducer}}$$



composition $\circ$



tensor $\otimes$



trace

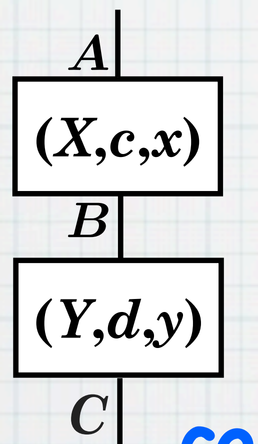
Hasuo (Tokyo)

Trans(T) by Coalgebraic Component Calculus

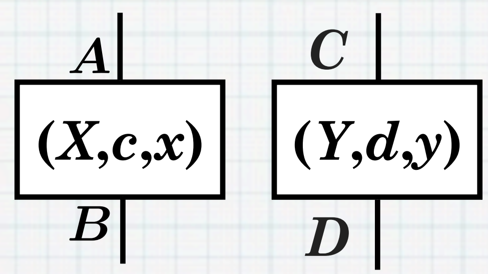
[Barbosa '03][IH & Jacobs '11]

$\boxed{\mathbf{Trans}(T)}$ Objects: sets $A, B, \ldots$

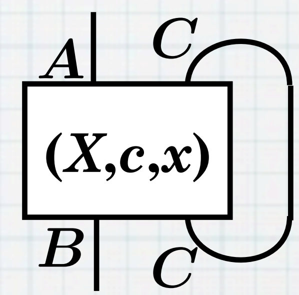
Arrows:
$$\frac{A \longrightarrow B \text{ in } \mathbf{Trans}(T)}{(X,\ X \times A \stackrel{c}{\to} T(X \times B),\ x_0 \in X\ ),\ T\text{-transducer}}$$



composition $\circ$



tensor $\otimes$



trace

$$\left(X \times Y,\ \begin{array}{ll} (X \times Y) \times A & \stackrel{\cong}{\longrightarrow} (X \times A) \times Y \\ & \stackrel{c \times Y}{\longrightarrow} T(X \times B) \times Y \\ & \stackrel{\mathrm{str}'}{\longrightarrow} T\big((X \times B) \times Y\big) \\ & \stackrel{\cong}{\longrightarrow} T\big(X \times (Y \times B)\big) \\ & \stackrel{T(X \times d)}{\longrightarrow} T\big(X \times T(Y \times C)\big) \\ & \stackrel{T\mathrm{str}}{\longrightarrow} TT\big(X \times (Y \times C)\big) \\ & \stackrel{\mu^T}{\longrightarrow} T\big(X \times (Y \times C)\big) \\ & \stackrel{\cong}{\longrightarrow} T\big((X \times Y) \times C\big) \end{array},\ (x, y) \right)$$

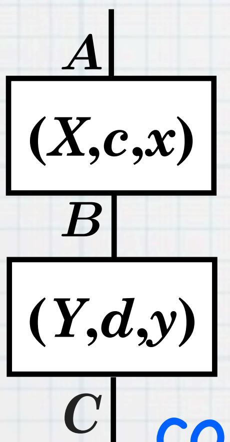
Hasuo (Tokyo)

Trans(T) by Coalgebraic Component Calculus

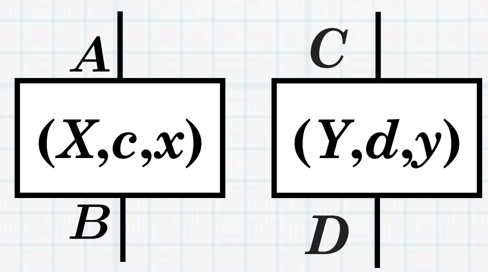
[Barbosa '03][IH & Jacobs '11]

$\boxed{\mathbf{Trans}(T)}$ Objects: sets $A, B, \dots$

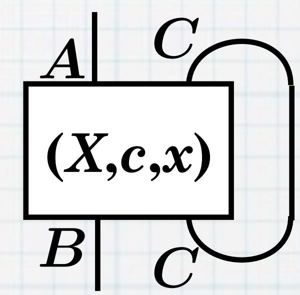
Arrows:
$$\frac{\color{red}{A} \longrightarrow \color{blue}{B} \text{ in } \mathbf{Trans}(T)}{\big(\, X,\ X \times \color{red}{A} \xrightarrow{c} T(X \times \color{blue}{B}),\ x_0 \in X\ \big),\ T\text{-transducer}}$$



composition

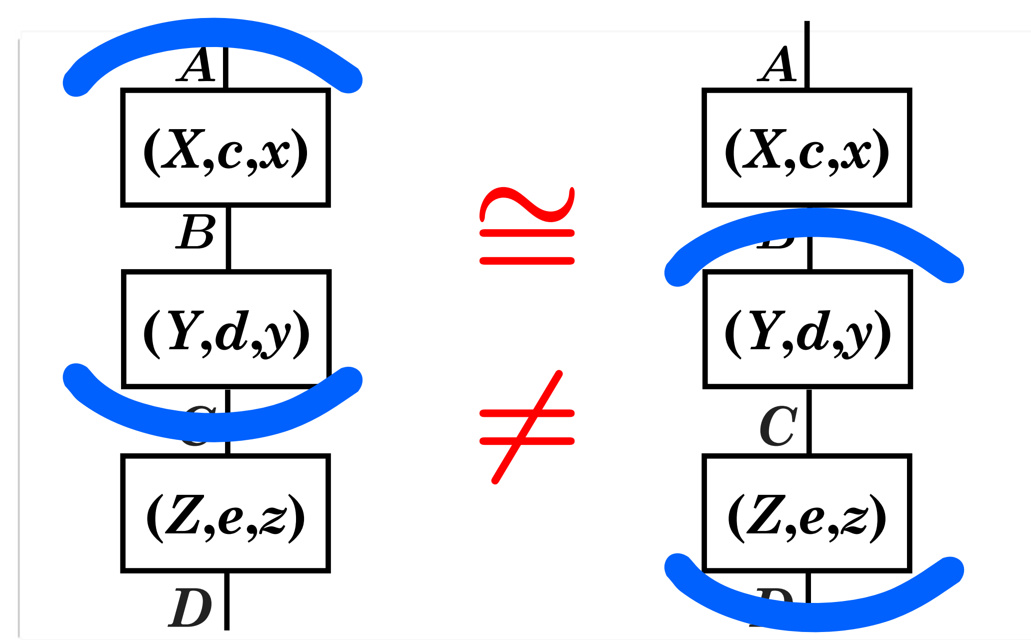


tensor



trace

* Trans(T) is a "category"...



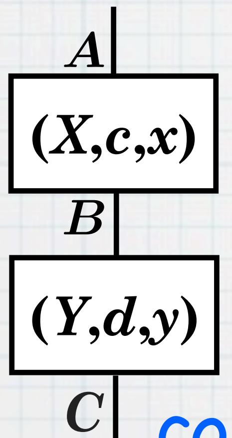
Hasuo (Tokyo)

Trans(T) by Coalgebraic Component Calculus

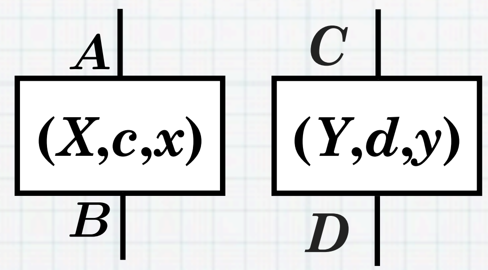
[Barbosa '03][IH & Jacobs '11]

$\boxed{\mathbf{Trans}(T)}$ Objects: sets $A, B, \ldots$

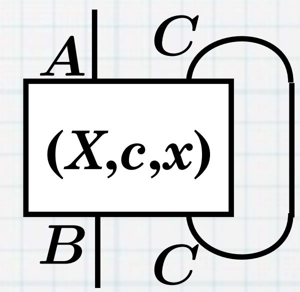
Arrows:
$$\frac{\color{red}{A} \longrightarrow \color{blue}{B} \text{ in } \mathbf{Trans}(T)}{\overline{\left(\, X,\ X \times \color{red}{A} \xrightarrow{c} T(X \times \color{blue}{B}),\ x_0 \in X \,\right),\ T\text{-transducer}}}$$



composition

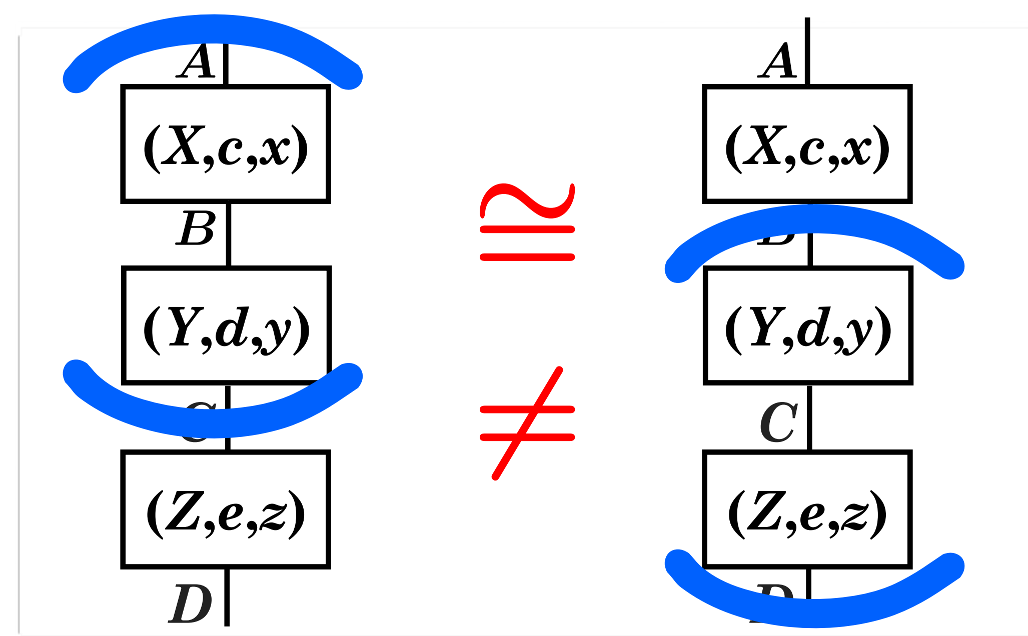


tensor



trace

* Trans(T) is a "category"...
* Fix: quotient modulo **behavioral equivalence** (homomorphisms of T-transducers) ➜ **resumptions** [Abramsky]



The Memoryful GoI Framework

* Given:
  * a **monad** T on $\mathbf{Sets}$, s.t. $\mathbf{Kl}(T)$ is $\mathbf{Cppo}$-enriched
  * an alg. signature Σ with **algebraic operations** on T [Plotkin & Power]

$$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$$

* For the calculus: $\lambda_{\mathbf{c}}$ + (alg. opr. from Σ) + (co)products + arith.
* We give

The Memoryful GoI Framework

- Given:
  - a **monad** T on $\mathbf{Sets}$, s.t. $\mathbf{Kl}(T)$ is $\mathbf{Cppo}$-enriched
  - an alg. signature Σ with **algebraic operations** on T [Plotkin & Power]

$$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$$

- *Exception* $\mathbf{1} + E + (\_)$
  - with $\mathbf{0}$-ary opr. $\mathbf{raise}_e$ ($e \in E$)
- *Nondeterminism* $\mathcal{P}$
  - with binary opr. $\sqcup$
- *Probability* $\mathcal{D}$, where $\mathcal{D}X = \{d \colon X \to [0,1] \mid \sum_x d(x) \le 1\}$
  - with binary opr. $\sqcup_p$ ($p \in [0,1]$)
- *Global state* $(\mathbf{1} + S \times \_)^S$
  - with $|V|$-ary $\mathbf{lookup}_l$ and unary $\mathbf{update}_{l,v}$

- For the calculus: λ_c + (alg. opr. from Σ) + (co)products + arith.
- We give

The Memoryful GoI Framework

- Given:
  - a **monad** T on $\mathbf{Sets}$, s.t. $\mathbf{Kl}(T)$ is $\mathbf{Cppo}$-enriched
  - an alg. signature Σ with **algebraic operations** on T [Plotkin & Power]

$$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$$

- *Exception* $\mathbf{1} + E + (\_)$
  - with $\mathbf{0}$-ary opr. $\mathbf{raise}_e$ $(e \in E)$
- *Nondeterminism* $\mathcal{P}$
  - with binary opr. $\sqcup$
- *Probability* $\mathcal{D}$, where $\mathcal{D}X = \{d \colon X \to [0,1] \mid \sum_x d(x) \leq 1\}$
  - with binary opr. $\sqcup_p$ $(p \in [0,1])$
- *Global state* $(\mathbf{1} + S \times \_)^S$
  - with $|V|$-ary $\mathbf{lookup}_l$ and unary $\mathbf{update}_{l,v}$

- For the calculus: λ_c + (alg. opr. from Σ) + (co)products + arith.

The Memoryful GoI Framework

* Given:
  * a **monad** T on $\mathbf{Sets}$, s.t. $\mathbf{Kl}(T)$ is $\mathbf{Cppo}$-enriched
  * an alg. signature Σ with **algebraic operations** on T [Plotkin & Power]

$$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$$

- *Exception* $\mathbf{1} + E + (\_)$
  - with $\mathbf{0}$-ary opr. $\mathbf{raise}_e$ ($e \in E$)
- *Nondeterminism* $\mathcal{P}$
  - with binary opr. $\sqcup$
- *Probability* $\mathcal{D}$, where $\mathcal{D}X = \{d \colon X \to [0,1] \mid \sum_x d(x) \leq 1\}$
  - with binary opr. $\sqcup_p$ ($p \in [0,1]$)
- *Global state* $(\mathbf{1} + S \times \_)^S$
  - with $|V|$-ary $\mathbf{lookup}_l$ and unary $\mathbf{update}_{l,v}$

* For the calculus: λ_c + (alg. opr. from Σ) + (co)products + arith.

$$\frac{\Gamma \vdash M_1 : \tau \quad \cdots \quad \Gamma \vdash M_{|\alpha|} : \tau}{\Gamma \vdash \alpha(M_1, \ldots, M_{|\alpha|}) : \tau}\ \alpha \in \Sigma$$

The Memoryful GoI Framework

- Given:
  - a **monad** T on $\mathbf{Sets}$, s.t. $\mathbf{Kl}(T)$ is $\mathbf{Cppo}$-enriched
  - an alg. signature Σ with **algebraic operations** on T
    [Plotkin & Power]

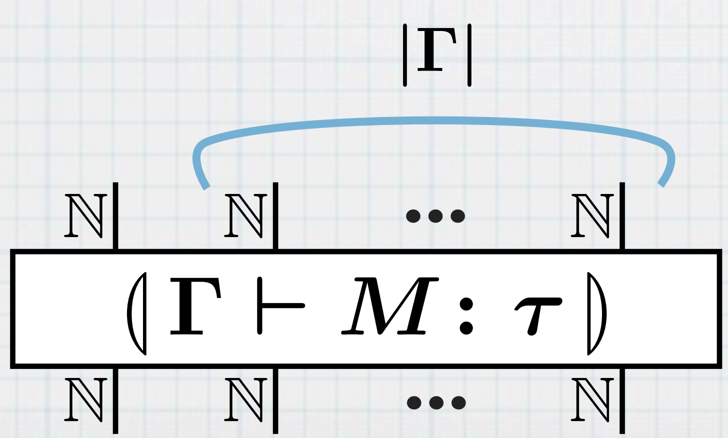
$$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$$

- *Exception* $\mathbf{1} + E + (\_)$
  - with $\mathbf{0}$-ary opr. $\mathbf{raise}_e$ $(e \in E)$
- *Nondeterminism* $\mathcal{P}$
  - with binary opr. $\sqcup$
- *Probability* $\mathcal{D}$, where $\mathcal{D}X = \{d \colon X \to [0,1] \mid \sum_x d(x) \leq 1\}$
  - with binary opr. $\sqcup_p$ $(p \in [0,1])$
- *Global state* $(\mathbf{1} + S \times \_)^S$
  - with $|V|$-ary $\mathbf{lookup}_l$ and unary $\mathbf{update}_{l,v}$

- For the calculus: λ_c + (alg. opr. from Σ) + (co)products + arith.

$$\frac{\Gamma \vdash M_1 : \tau \quad \cdots \quad \Gamma \vdash M_{|\alpha|} : \tau}{\Gamma \vdash \alpha(M_1, \ldots, M_{|\alpha|}) : \tau} \; \alpha \in \Sigma$$

- We give



in $\mathbf{Trans}(T)$

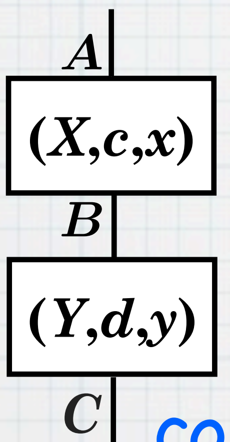
Hasuo (Tokyo)

Trans(T) by Coalgebraic Component Calculus

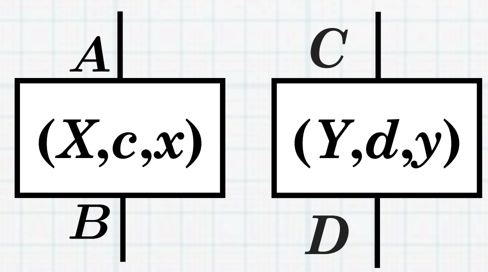
[Barbosa '03][IH & Jacobs '11]

Trans(T) Objects: sets $A, B, \ldots$

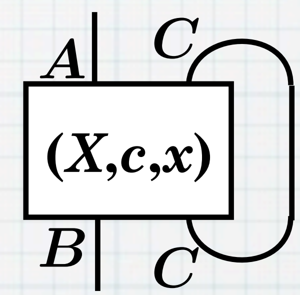
Arrows:
$$\frac{A \longrightarrow B \text{ in } \mathbf{Trans}(T)}{\big(\, X,\ X \times A \xrightarrow{c} T(X \times B),\ x_0 \in X \,\big),\ T\text{-transducer}}$$



composition



tensor



trace

$$\left(X \times Y,\ \begin{array}{rl} (X \times Y) \times A & \xrightarrow{\cong} (X \times A) \times Y \\ & \xrightarrow{c \times Y} T(X \times B) \times Y \\ & \xrightarrow{\mathrm{str}'} T\big((X \times B) \times Y\big) \\ & \xrightarrow{\cong} T\big(X \times (Y \times B)\big) \\ & \xrightarrow{T(X \times d)} T\big(X \times T(Y \times C)\big) \\ & \xrightarrow{T\mathrm{str}} TT\big(X \times (Y \times C)\big) \\ & \xrightarrow{\mu^T} T\big(X \times (Y \times C)\big) \\ & \xrightarrow{\cong} T\big((X \times Y) \times C\big) \end{array},\ (x, y) \right)$$

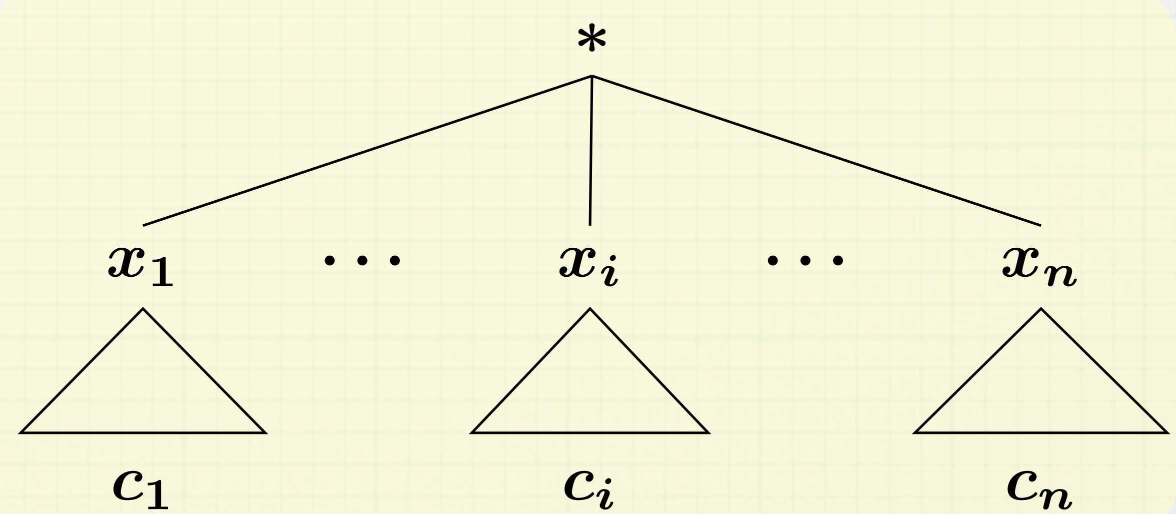
Hasuo (Tokyo)

Missing Ingredient I: Alg. Opr.

* $\alpha \in \Sigma_n$ an alg. operation

$$\alpha\left(\underset{B}{\overset{A}{\boxed{(X_1,c_1,x_1)}}}, \ldots, \underset{B}{\overset{A}{\boxed{(X_n,c_n,x_n)}}}\right)$$

$$= \Bigl(\{*\} + X_1 + \cdots + X_n ,$$



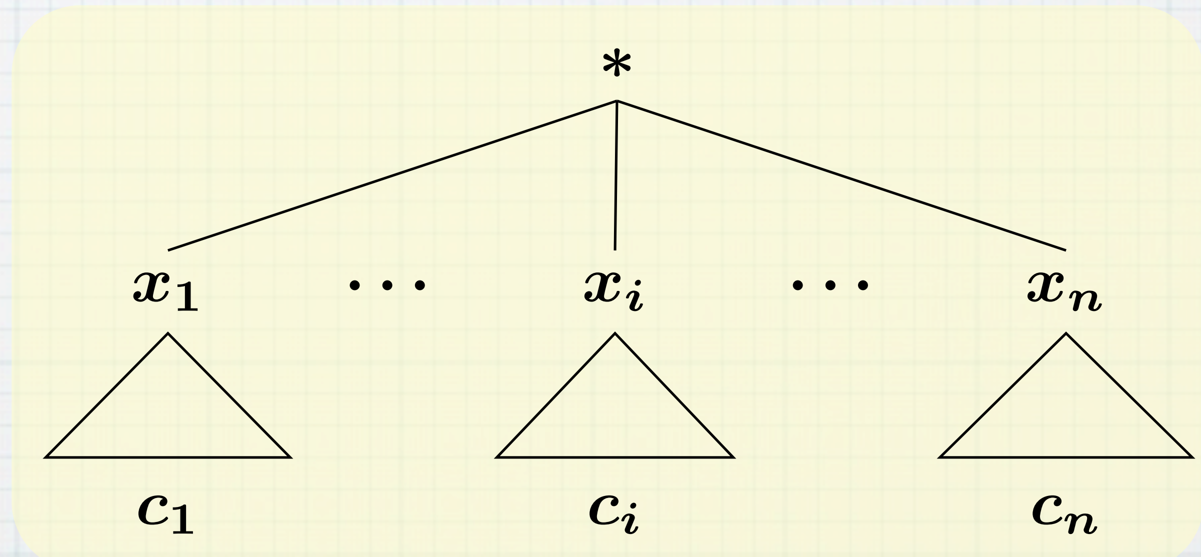
$$, \; * \Bigr)$$

Missing Ingredient I: Alg. Opr.

* $\alpha \in \Sigma_n$ an alg. operation

$$\alpha\Big(\overset{A}{\underset{B}{\boxed{(X_1,c_1,x_1)}}}\ ,\ \dots\ ,\ \overset{A}{\underset{B}{\boxed{(X_n,c_n,x_n)}}} \Big)$$

$$= \Big(\{*\} + X_1 + \cdots + X_n\ ,$$



$$,\ * \Big)$$

Fresh initial state

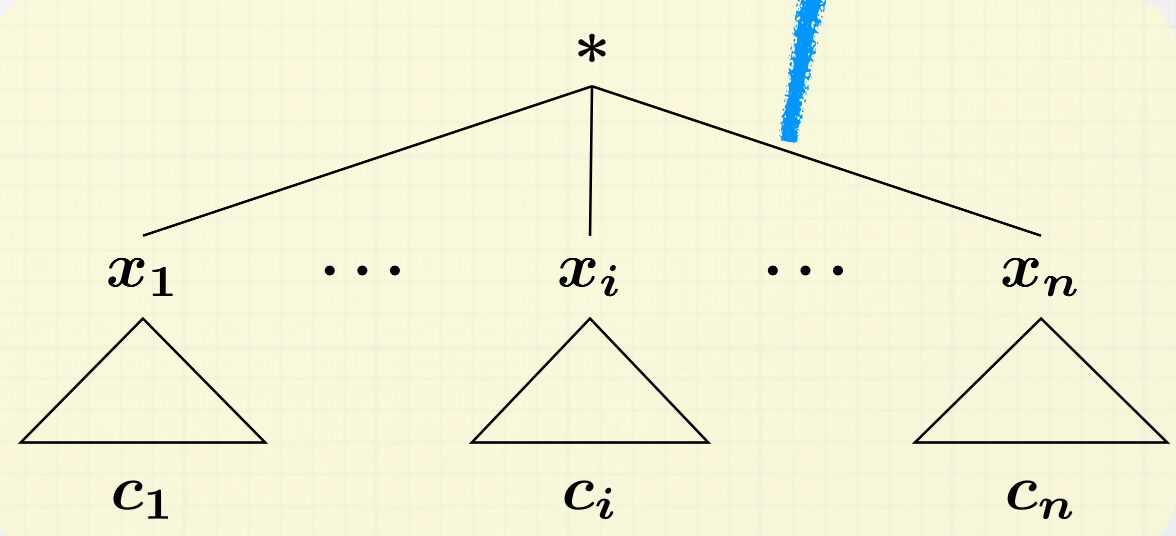
Missing Ingredient I: Alg. Opr.

- $\alpha \in \Sigma_n$ an alg. operation

T-branching given by
$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$

$$\alpha\left(\overset{A}{\underset{B}{\boxed{(X_1,c_1,x_1)}}} \;,\; \ldots \;,\; \overset{A}{\underset{B}{\boxed{(X_n,c_n,x_n)}}} \right)$$

$$= \left(\{*\} + X_1 + \cdots + X_n \;,\; \right.$$

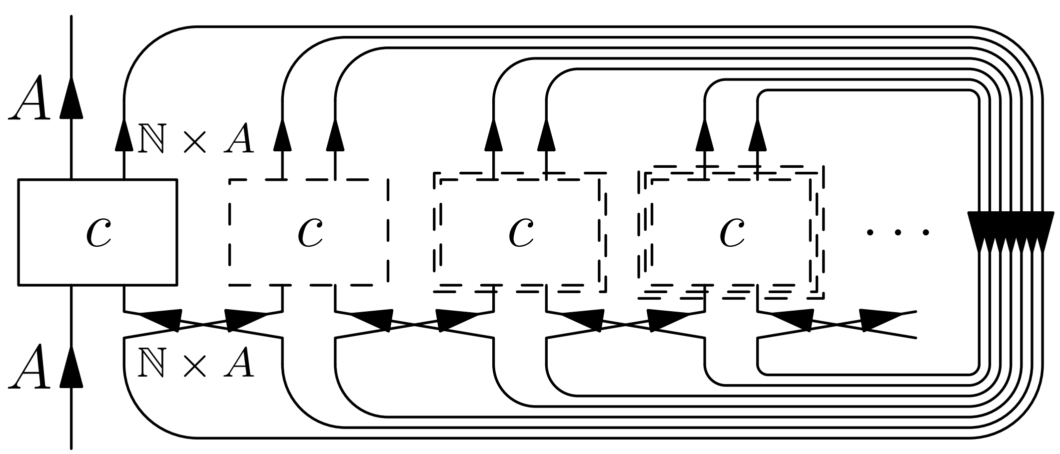


$$\left. ,\; * \right)$$

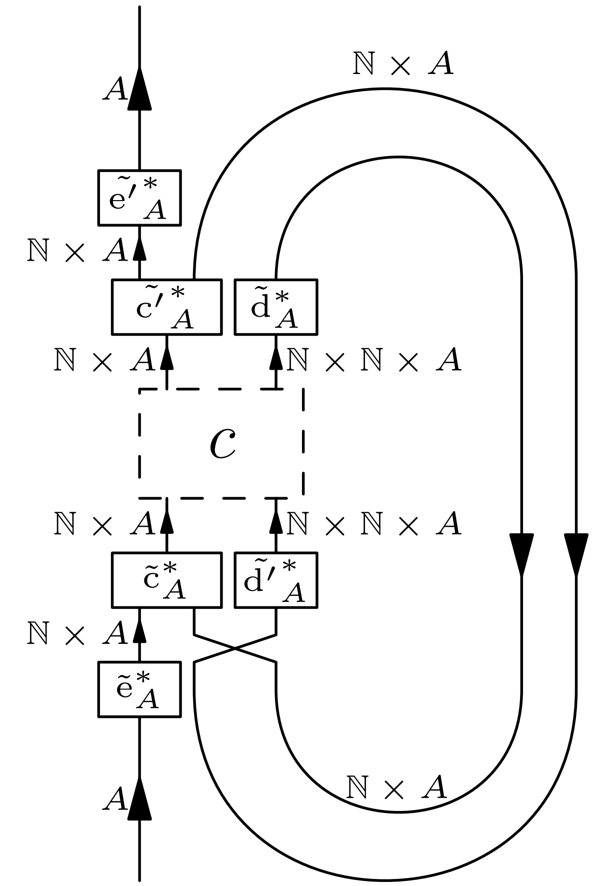
Fresh initial state

Missing Ingredient II: Recursion

Girard style
fixed point operator



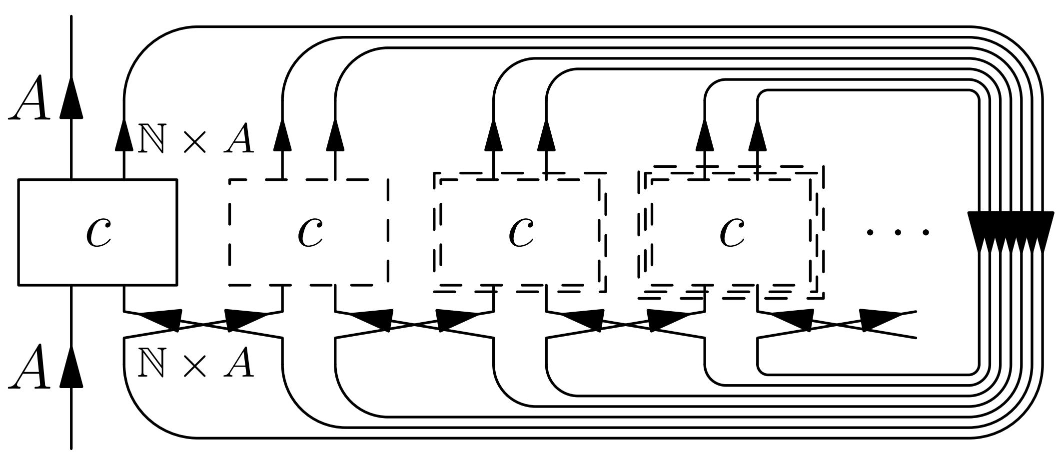
Mackie style
fixed point operator



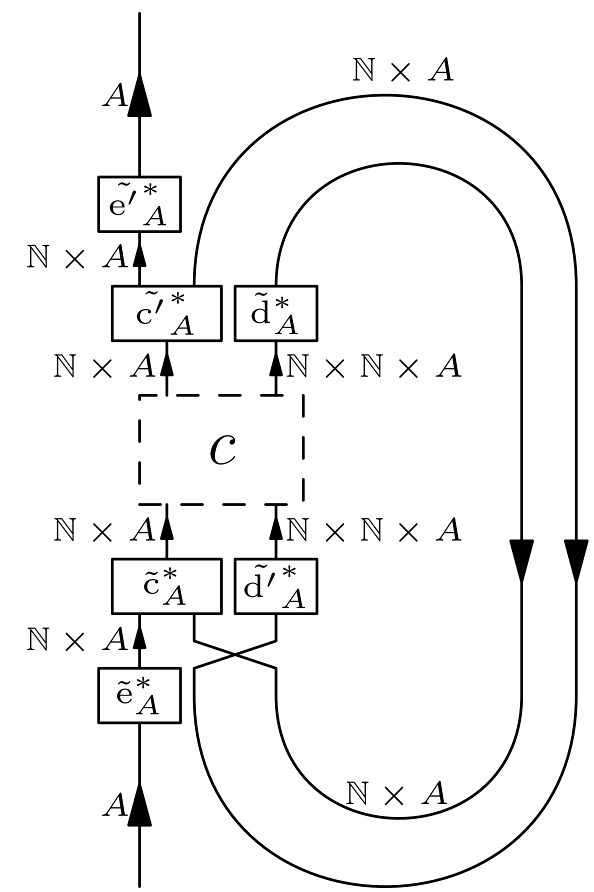
* Obviously a fixed point
* Fixed-point induction
* Finitary string diagram

Missing Ingredient II: Recursion

Girard style
fixed point operator



Mackie style
fixed point operator



- Obviously a fixed point
- Fixed-point induction
- Finitary string diagram

Theorem The two coincide. (for any suitable T!)

Hasuo (Tokyo)

The Memoryful GoI Framework

- Given:
  - a **monad** T on $\mathbf{Sets}$, s.t. $\mathbf{Kl}(T)$ is $\mathbf{Cppo}$-enriched
  - an alg. signature Σ with **algebraic operations** on T [Plotkin & Power]

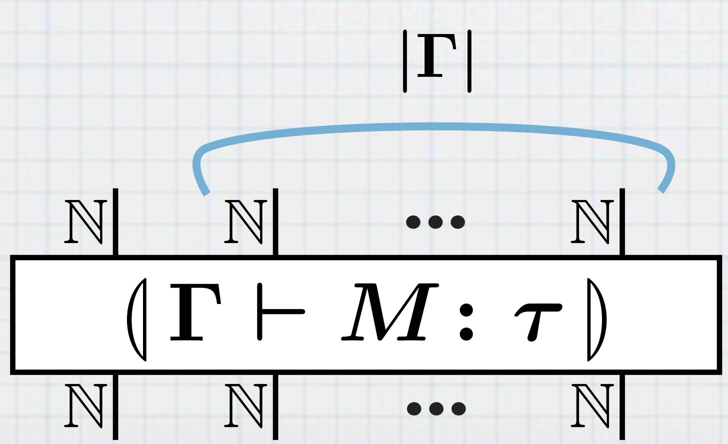
$$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$$

- *Exception* $1 + E + (\_)$
  - with 0-ary opr. $\mathbf{raise}_e$ $(e \in E)$
- *Nondeterminism* $\mathcal{P}$
  - with binary opr. $\sqcup$
- *Probability* $\mathcal{D}$, where $\mathcal{D}X = \{d \colon X \to [0,1] \mid \sum_x d(x) \leq 1\}$
  - with binary opr. $\sqcup_p$ $(p \in [0,1])$
- *Global state* $(1 + S \times \_)^S$
  - with $|V|$-ary $\mathbf{lookup}_l$ and unary $\mathbf{update}_{l,v}$

- For the calculus: λ_c + (alg. opr. from Σ) + (co)products

$$\frac{\Gamma \vdash M_1 : \tau \quad \cdots \quad \Gamma \vdash M_{|\alpha|} : \tau}{\Gamma \vdash \alpha(M_1, \ldots, M_{|\alpha|}) : \tau} \; \alpha \in \Sigma$$

- We give



in $\mathbf{Trans}(T)$

The Memoryful GoI Framework

* Given:
  * a **monad** T on $\mathbf{Sets}$, s.t. $\mathbf{Kl}(T)$ is $\mathbf{Cppo}$-enriched
  * an alg. signature Σ with **algebraic operations** on T
    [Plotkin & Power]

$$\left\{ \alpha_{A,B} \colon (A \Rightarrow TB)^{|\alpha|} \longrightarrow (A \Rightarrow TB) \right\}_{A \in \mathbf{Sets}, B \in \mathcal{K}\ell(T)}$$

- *Exception* $\mathbf{1} + E + (\_)$
  - with $\mathbf{0}$-ary opr. $\mathbf{raise}_e$ $(e \in E)$
- *Nondeterminism* $\mathcal{P}$
  - with binary opr. $\sqcup$
- *Probability* $\mathcal{D}$, where
  $\mathcal{D}X = \{d \colon X \to [0,1] \mid \sum_x d(x) \leq 1\}$
  - with binary opr. $\sqcup_p$ $(p \in [0,1])$
- *Global state* $(\mathbf{1} + S \times \_)^S$
  - with $|V|$-ary $\mathbf{lookup}_l$ and unary $\mathbf{update}_{l,v}$

* For the calculus: λ_c + (alg. opr. from Σ) + (co)products

$$\frac{\Gamma \vdash M_1 : \tau \quad \cdots \quad \Gamma \vdash M_{|\alpha|} : \tau}{\Gamma \vdash \alpha(M_1, \ldots, M_{|\alpha|}) : \tau} \; \alpha \in \Sigma$$

* We give

in $\mathbf{Trans}(T)$

The Memoryful GoI Framework

<u>Theorem</u> (Adequacy)

Let $\vdash \boldsymbol{M} : \mathtt{nat}$. Then, as elem. of $\boldsymbol{T}(\mathbb{N})$,

$$\left(\begin{array}{c} \mathbb{N} \\ \boxed{(\!|\vdash \boldsymbol{M} : \mathtt{nat}\,|\!)} \\ \mathbb{N} \end{array} \right)^{\dagger} = [\![\, |\boldsymbol{M}| \,]\!] \,.$$

* For the calculus: λ_c + (alg. opr. from Σ) + (co)products

$$\frac{\Gamma \vdash M_1 : \tau \quad \cdots \quad \Gamma \vdash M_{|\alpha|} : \tau}{\Gamma \vdash \alpha(M_1, \ldots, M_{|\alpha|}) : \tau}\ \alpha \in \Sigma$$

* We give

$$\begin{array}{c} \overbrace{\mathbb{N} \quad \mathbb{N} \quad \cdots \quad \mathbb{N}}^{|\Gamma|} \\ \boxed{(\!|\Gamma \vdash M : \tau|\!)} \\ \mathbb{N} \quad \mathbb{N} \quad \cdots \quad \mathbb{N} \end{array}$$

in $\mathbf{Trans}(T)$

The Memoryful GoI Framework

Theorem (Adequacy)

Let $\vdash \boldsymbol{M} : \mathtt{nat}$. Then, as elem. of $\boldsymbol{T}(\mathbb{N})$,

$$\left(\begin{array}{c} \mathbb{N} \\ \boxed{(\!| \vdash \boldsymbol{M} : \mathtt{nat} |\!)} \\ \mathbb{N} \end{array} \right)^{\dagger} = [\![\, |\boldsymbol{M}| \,]\!] \, .$$

feeding a query
and observing
the outcome

The Memoryful GoI Framework

Theorem (Adequacy)

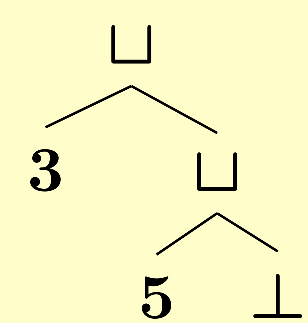
Let $\vdash M : \mathtt{nat}$. Then, as elem. of $T(\mathbb{N})$,

$$\left(\begin{array}{c}\mathbb{N}\\ \boxed{(\!|\vdash M : \mathtt{nat}\,|\!)}\\ \mathbb{N}\end{array}\right)^{\dagger} = [\![\,|M|\,]\!] \;.$$

feeding a query and observing the outcome

Opr. sem.: Plotkin-Power **effect-value**. E.g.

$$|\,\mathbf{3} \sqcup (\mathbf{5} \sqcup \mathtt{div})\,| =$$



Interpretation

$$[\![\_]\!] : \mathbf{EffVal}_{\mathbb{N}}^{\Sigma} \longrightarrow T(\mathbb{N})$$

(exploiting free conti. Σ-alg.)

Theorem (Adequacy)

Let $\vdash M : \mathtt{nat}$. Then, as elem. of $T(\mathbb{N})$,

$$\left(\begin{array}{c} \mathbb{N} \\ \boxed{(\!|\vdash M : \mathtt{nat}|\!)} \\ \mathbb{N} \end{array} \right)^{\dagger} = [\![\, |M| \,]\!] \; .$$

feeding a query and observing the outcome

Opr. sem.: Plotkin-Power **effect-value**. E.g.

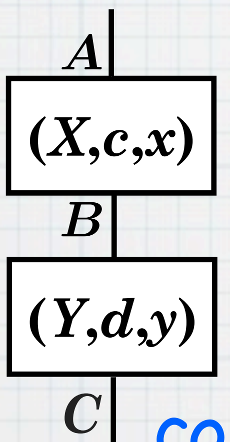
$$|\,3 \sqcup (5 \sqcup \mathtt{div})\,| = \begin{array}{c} \sqcup \\ 3 \quad \sqcup \\ \quad\;\; 5 \quad \bot \end{array}$$

Trans(T) by Coalgebraic Component Calculus

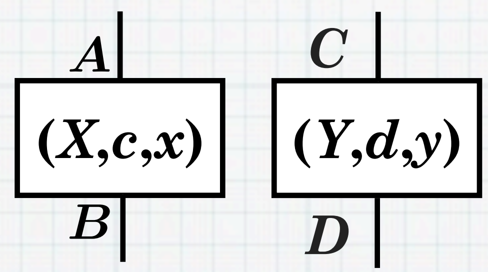
[Barbosa '03][IH & Jacobs '11]

$\boxed{\mathbf{Trans}(T)}$ Objects: sets $A, B, \ldots$

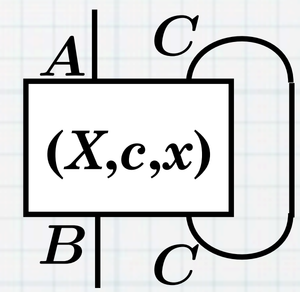
Arrows:
$$\frac{{\color{red}A} \longrightarrow {\color{blue}B} \text{ in } \mathbf{Trans}(T)}{\big(\, X,\ X \times {\color{red}A} \xrightarrow{c} T(X \times {\color{blue}B}),\ x_0 \in X\ \big),\ T\text{-transducer}}$$



composition



tensor

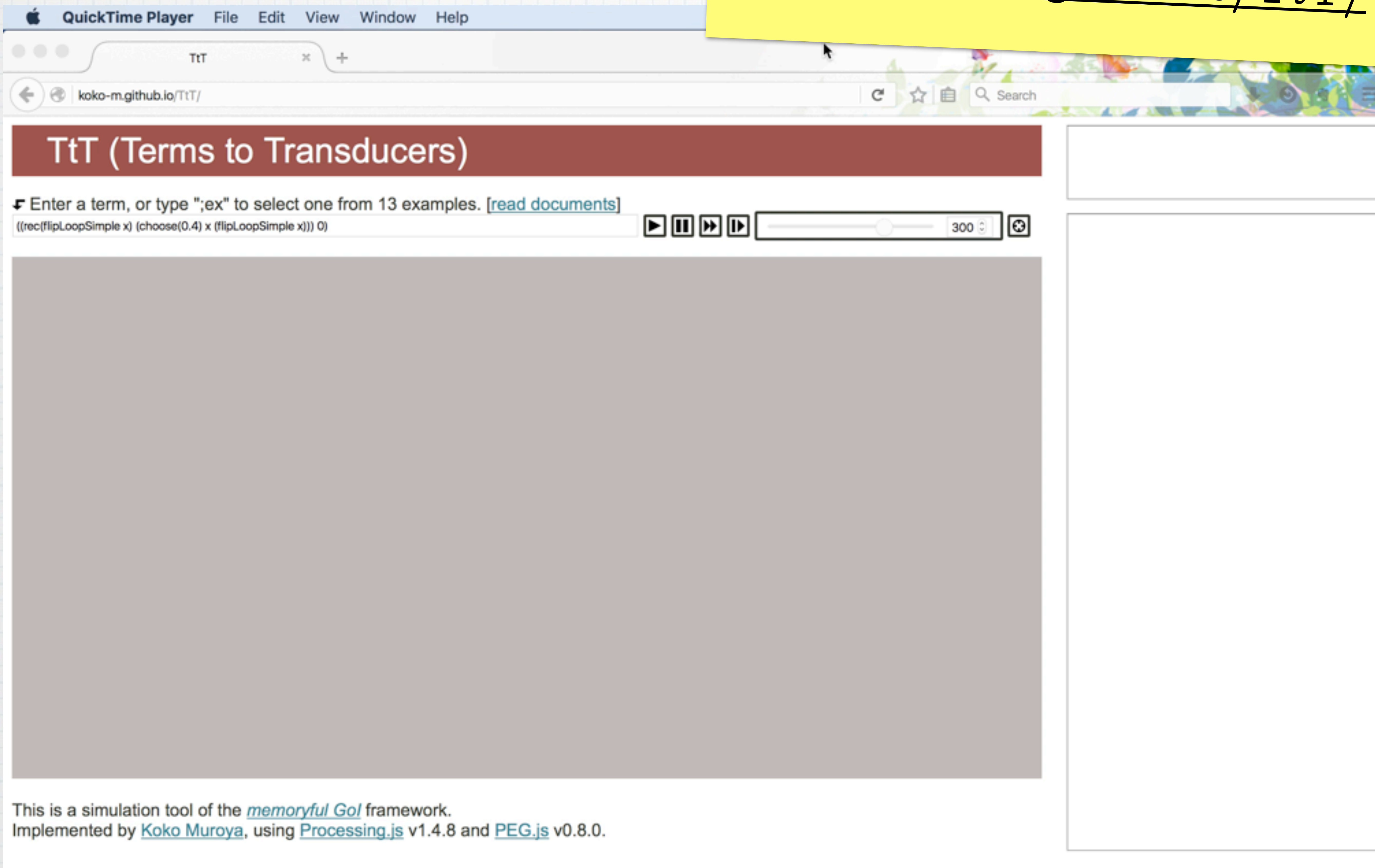


trace

$$\left(\ X \times Y,\quad \begin{aligned} (X \times Y) \times A &\xrightarrow{\cong} (X \times A) \times Y \\ &\xrightarrow{c \times Y} T(X \times B) \times Y \\ &\xrightarrow{\mathrm{str}'} T\big((X \times B) \times Y\big) \\ &\xrightarrow{\cong} T\big(X \times (Y \times B)\big) \\ &\xrightarrow{T(X \times d)} T\big(X \times T(Y \times C)\big) \\ &\xrightarrow{T\mathrm{str}} TT\big(X \times (Y \times C)\big) \\ &\xrightarrow{\mu^T} T\big(X \times (Y \times C)\big) \\ &\xrightarrow{\cong} T\big((X \times Y) \times C\big) \end{aligned},\quad (x, y)\ \right)$$

Our Tool TtT

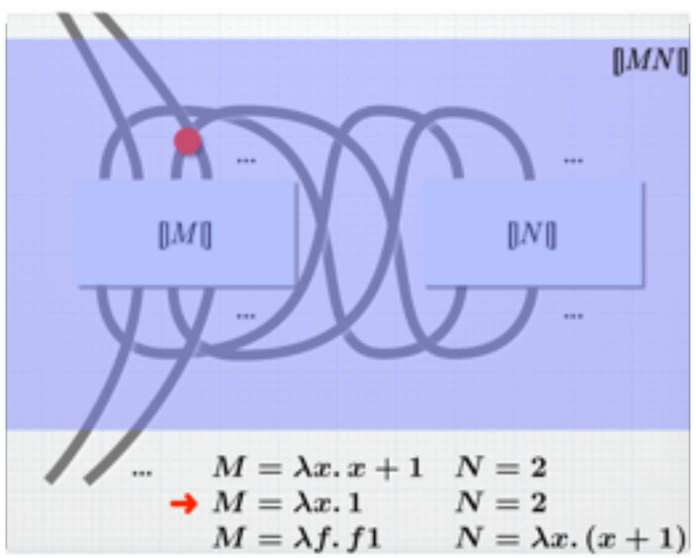
Developed by Koko Muroya
http://koko-m.github.io/TtT/

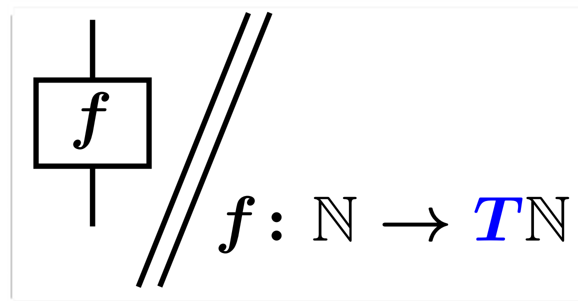


Summary

Coalgebra meets **higher-order computation** in **Geometry of Interaction** [Girard, LC'88]

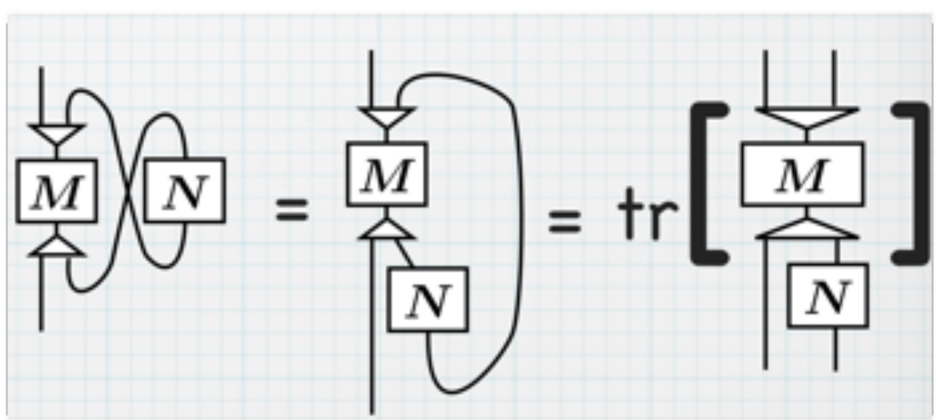
"GoI Animation"





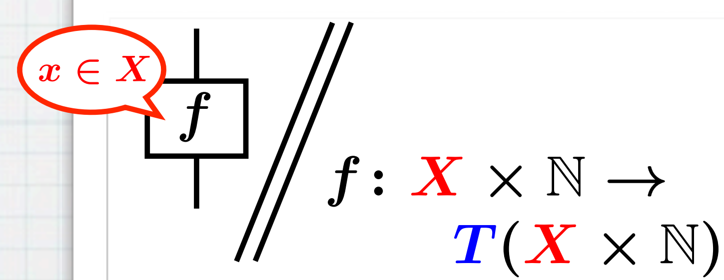
GoI w/ T-branching

[IH & Hoshino, LICS'11]



Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]



Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

Hasuo (Tokyo)

Friday, June 24, 16

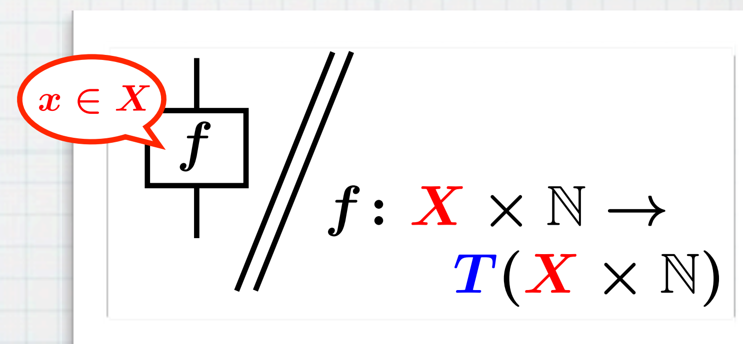
Summary

Coalgebra meets **higher-order computation**
in **Geometry of Interaction** [Girard, LC'88]

"GoI Animation"

GoI w/ T-branching
[IH & Hoshino, LICS'11]

Categorical GoI
[Abramsky, Haghverdi & Scott, MSCS'02]



Memoryful GoI
[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

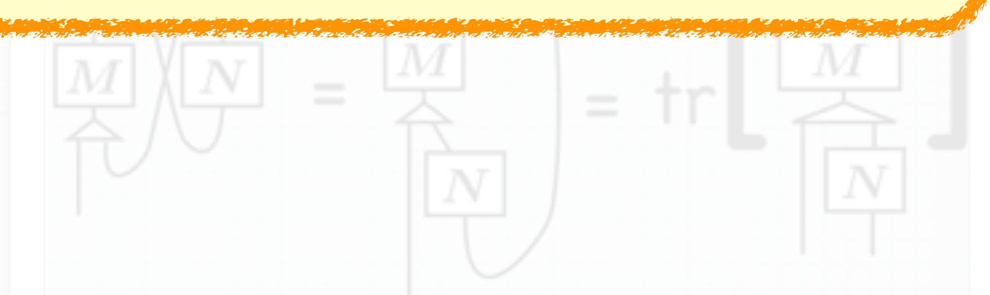
Hasuo (Tokyo)

* GoI + algebraic effects [Plotkin & Power]

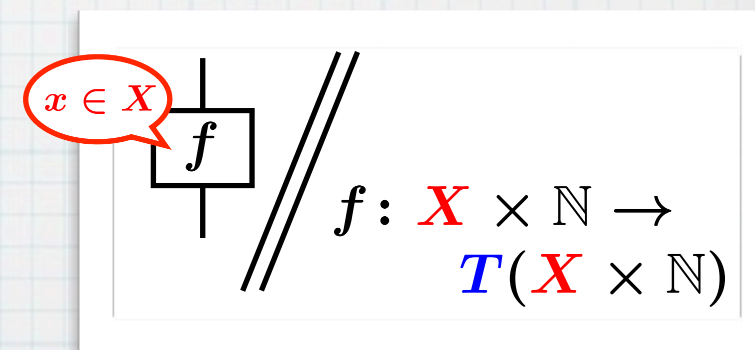
r computation

on [Girard, LC'88]

GoI w/ T-branching [IH & Hoshino, LICS'11]



Categorical GoI [Abramsky, Haghverdi & Scott, MSCS'02]



Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

Hasuo (Tokyo)

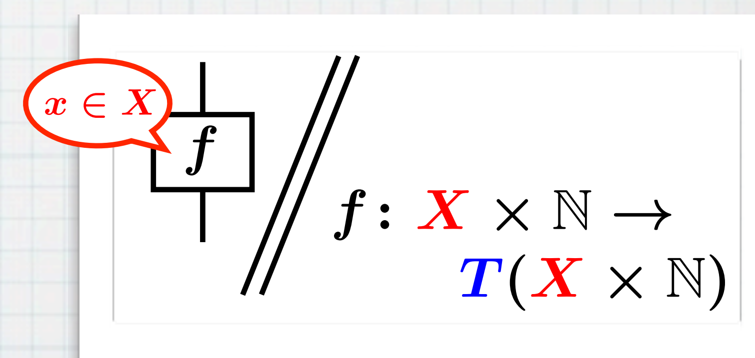
* GoI + algebraic effects [Plotkin & Power]
* via T-branching transducers

r computation

on [Girard, LC'88]

GoI w/ T-branching

[IH & Hoshino, LICS'11]



Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]

Hasuo (Tokyo)

* GoI + algebraic effects [Plotkin & Power]
* via T-**branching transducers**
* Compositional translation

$$(\!|\, \Gamma \vdash M : \tau \,|\!)$$

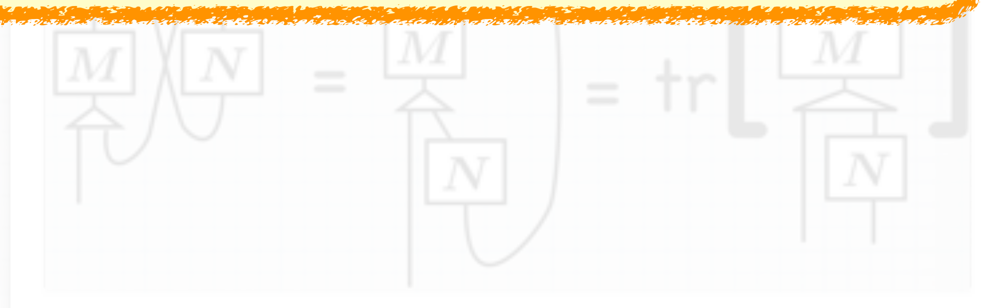
==> **implementation TtT**

r computation

on [Girard, LC'88]

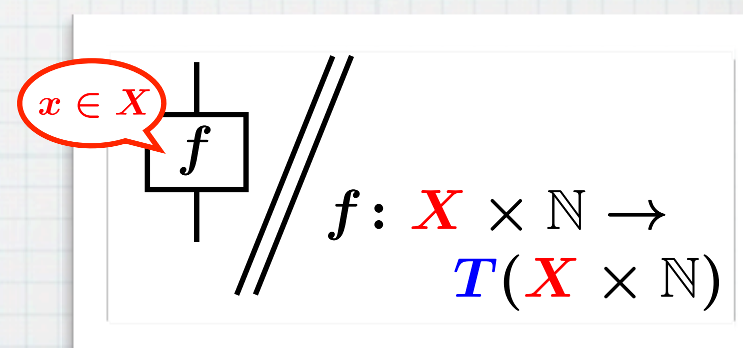
GoI w/ T-branching

[IH & Hoshino, LICS'11]



Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]



Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

Hasuo (Tokyo)

- GoI + algebraic effects [Plotkin & Power]
- via T-**branching transducers**
- Compositional translation

$$(\!|\, \Gamma \vdash M : \tau \,|\!)$$

==> **implementation TtT**

- (Categorical GoI + realizability)

==> **categorical model**

  - <u>Thm.</u> Adequacy
  - "Correct-by-construction" compilation!

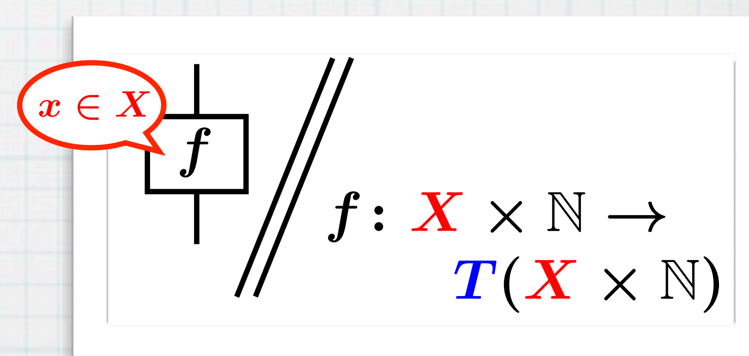
[Girard, LC'88]

GoI w/ T-branching

[IH & Hoshino, LICS'11]

Categorical GoI

[Abramsky, Haghverdi & Scott, MSCS'02]



Memoryful GoI

[Hoshino, Muroya & IH, CSL-LICS'14 & POPL'16]

Hasuo (Tokyo)

Retracing some paths in Process Algebra

Samson Abramsky
Laboratory for the Foundations of Computer Science
University of Edinburgh

1 Introduction

The very existence of the CONCUR conference bears witness to the fact that "concurrency theory" has developed into a subject unto itself, with substantially different emphases and techniques to those prominent elsewhere in the semantics of computation.

Whatever the past merits of this separate development, it seems timely to look for some convergence and unification. In addressing these issues, I have found it instructive to trace some of the received ideas in concurrency back to their origins in the early 1970's. In particular, I want to focus on a seminal paper by Robin Milner [Mil75][1], which led in a fairly direct line to his enormously influential work on CCS [Mil80, Mil89]. I will take (to the extreme) the liberty of of applying hindsight, and show how some different paths could have been taken, which, it can be argued, lead to a more unified approach to the semantics of computation, and moreover one which may be better suited to modelling today's concurrent, object-oriented languages, and the type systems and logics required to support such languages.

2 The semantic universe: transducers

Milner's starting point was the classical automata-theoretic notion of *transducers*, *i.e.* structures

$$(Q, X, Y, q_0, \delta)$$

CONCUR'96

Hasuo (Tokyo)

Retracing some paths in Process Algebra

Samson Abramsky
Laboratory for the Foundations of Computer Science
University of Edinburgh

1 Introduction

The very existence of the CONCUR conference bears witness to the fact that "concurrency theory" has developed into a subject unto itself, with substantially different emphases and techniques to those prominent elsewhere in the semantics of computation.

Whatever the past merits of this separate development, it seems timely to look for some convergence and unification. In addressing these issues, I have found it instructive to trace some of the received ideas in concurrency back to their origins in the early 1970's. In particular, I want to focus on a seminal paper by Robin Milner [Mil75][1], which led in a fairly direct line to his enormously influential work on CCS [Mil80, Mil89]. I will take (to the extreme) the liberty of of applying hindsight, and show how some different paths could have been taken, which, it can be argued, lead to a more unified approach to the semantics of computation, and moreover one which may be better suited to modelling today's concurrent, object-oriented languages, and the type systems and logics required to support such languages.

2 The semantic universe: transducers

Milner's starting point was the classical automata-theoretic notion of *transducers*, *i.e.* structures

$$(Q, X, Y, q_0, \delta)$$

CONCUR'96

Hasuo (Tokyo)

Retracing some paths in Process Algebra

Samson Abramsky
Laboratory for the Foundations of Computer Science
University of Edinburgh

1 Introduction

The very existence of the CONCUR conference bears witness to the fact that "concurrency theory" has developed into a subject unto itself, with substantially different emphases and techniques to those prominent elsewhere in the semantics of computation.

Whatever the past merits of this separate development, it seems timely to look for some convergence and unification. In addressing these issues, I have found it instructive to trace some of the received ideas in concurrency back to their origins in the early 1970's. In particular, I want to focus on a seminal paper by Robin Milner [Mil75][1], which led in a fairly direct line to his enormously influential work on CCS [Mil80, Mil89]. I will take (to the extreme) the liberty of of applying hindsight, and show how some different paths could have been taken, which, it can be argued, lead to a more unified approach to the semantics of computation, and moreover one which may be better suited to modelling today's concurrent, object-oriented languages, and the type systems and logics required to support such languages.

2 The semantic universe: transducers

Milner's starting point was the classical automata-theoretic notion of *transducers*, *i.e.* structures

$$(Q, X, Y, q_0, \delta)$$

CONCUR'96

Hasuo (Tokyo)

Retracing some paths in Process Algebra

Samson Abramsky
Laboratory for the Foundations of Computer Science
University of Edinburgh

Thank you for your attention!
Ichiro Hasuo (Dept. CS, U Tokyo)
http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro/

1 Introduction

The very existence of the CONCUR conference bears witness to the fact that "concurrency theory" has developed into a subject unto itself, with substantially different emphases and techniques to those prominent elsewhere in the semantics of computation.

Whatever the past merits of this separate development, it seems timely to look for some convergence and unification. In addressing these issues, I have found it instructive to trace some of the received ideas in concurrency back to their origins in the early 1970's. In particular, I want to focus on a seminal paper by Robin Milner [Mil75][1], which led in a fairly direct line to his enormously influential work on CCS [Mil80, Mil89]. I will take (to the extreme) the liberty of of applying hindsight, and show how some different paths could have been taken, which, it can be argued, lead to a more unified approach to the semantics of computation, and moreover one which may be better suited to modelling today's concurrent, object-oriented languages, and the type systems and logics required to support such languages.

2 The semantic universe: transducers

Milner's starting point was the classical automata-theoretic notion of *transducers*, *i.e.* structures

$$(Q, X, Y, q_0, \delta)$$

CONCUR'96