Talk based on:
I. Hasuo & N. Hoshino,
Semantics of Higher-Order Quantum Computation via Geometry of Interaction,
In Proc. Logic in Computer Science (LICS), June 2011.

# Quantum Geometry of Interaction


Ichiro Hasuo
University of Tokyo (JP)

Naohiko Hoshino
RIMS, Kyoto University (JP)


東京大学 THE UNIVERSITY OF TOKYO

京都大学 KYOTO UNIVERSITY



# What's Done

- The Categorical GoI workflow
  - GoI = "Geometry of Interaction"
  - General, standard construction of denotational models
- Applied to quantum computation
  - Quantum λ-calculus = linear λ-cal. + quantum constructs
  - with insights from theory of coalgebra
  - Outcome: first adequate denotational semantics for a full quantum language (with ! and recursion)

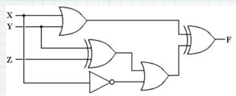
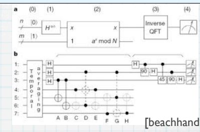


# Plan

- The categorical GoI workflow
  [Abramsky, Haghverdi, Scott, Jacobs, Longley, Lenisa, Hoshino, ...]
  - GoI + realizability
  - Generic — still concrete and dynamic
  - Coalgebraic view ➔ let's do something fancy
- Elements of quantum computation
  - Not much, really!
- The calculus $q\lambda_\ell$ Based on [Selinger-Valiron'09]
- The denotational model



# Quantum λ-calculus

| | Classical | | Quantum |
|---|---|---|---|
| (Boolean) circuit |  [Null-Lobur] | Quantum circuit |  [beachhandball.es] |
| Programming language | `int i,j; int factorial(int k) { j=1; for (i=1; i<=k; i++) j=j*i; return j; }` | Quantum programming language | telep = let $\langle x,y\rangle$ = **EPR** $*$ in let $f$ = **BellMeasure** $x$ in let $g$ = **U** $y$ in $\langle f,g\rangle$. [Selinger-Valiron] |

- Quantum λ:
  prototype of quantum functional language

## Quantum λ-Calculus: Prototype of Quantum Functional Languages

- **Why** (high-level) language?
  ➜ structured programming
  - Discovery of new algorithms
  - Program verification
- **Why** functional language?
  ➜ Mathematically nice and clean
  - Aids (denotational) semantics
  - Transfer from classical to quantum



## Quantum λ-Calculus: Prototype of Quantum Functional Languages

- **Why** denotational semantics?
  ➜ For quantum communication as well as for quantum computation
  - "Absolute security" via e.g. quantum key distr.
  - Being tested for real-world usege
  - Comm. protocols are notoriously error-prone; quantum primitives make it worse



## Quantum λ-Calculus: Prototype of Quantum Functional Languages

- Linear λ-calculus
  - "No cloning" by linearity:
  - Classical data (duplicable) via !
- + Quantum primitives
  - State preparation
  - Unitary transformation
  - Measurement



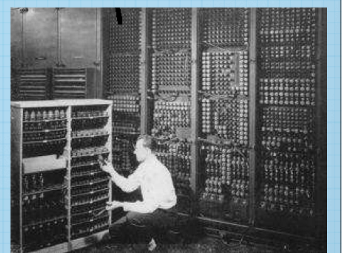
## "Quantum Data, Classical Control"

Illustration by N. Hoshino

Quantum data

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

Classical control



# Denotational Semantics for Quantum λ

* In **Hilb** ?
  * Not that easy. Classical data?
* [Selinger&Valiron'08] Den. sem. for the !-free fragment
* [Selinger&Valiron'09] Operational semantics (nice!)
* [Current Work]
  * The first model for the full fragment (with ! and recursion)
  * Categorical GoI: useful for "Quantum Data, Classical Control"



# Critical Acclaim (?) for:

## I. Hasuo & N. Hoshino, Semantics of Higher-Order Quantum Computation via Geometry of Interaction

* "[T]he amount of material ... goes far beyond the 10-page limit ... Now, I understand that self-containedness is an impossible objective in cases like this, but ..." —*Reviewer 3*
* "This is clearly a 30-page paper (or more) than has been compressed into 10 pages." —*Reviewer 4*
* **Now their pain is yours!!**



# Part 1

## Categorical GoI (Geometry of Interaction)



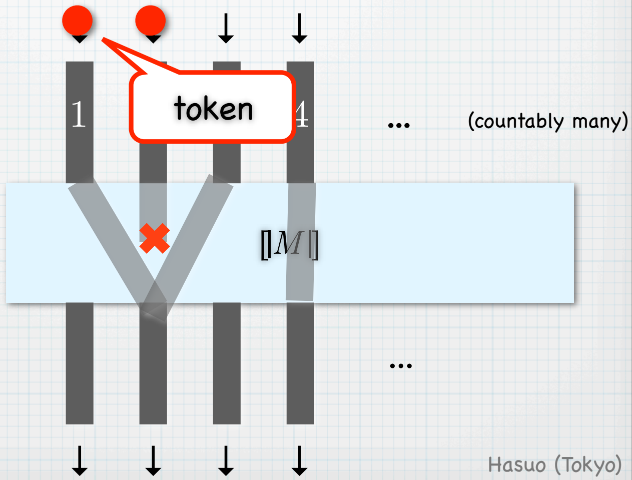
# GoI: Geometry of Interaction

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

# The GoI Animation

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

$=$ "piping"





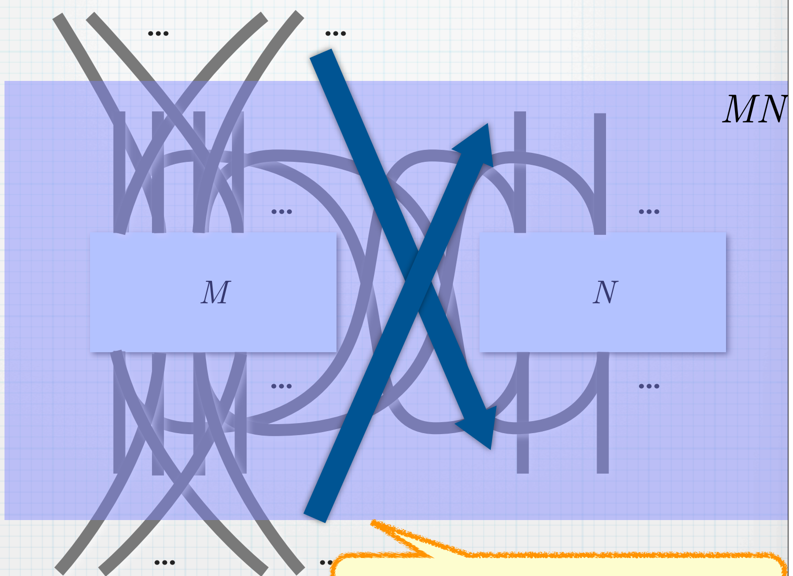

# The GoI Animation

- Function application $[\![MN]\!]$
  - by "parallel composition + hiding"



$[\![MN]\!]$

$=$

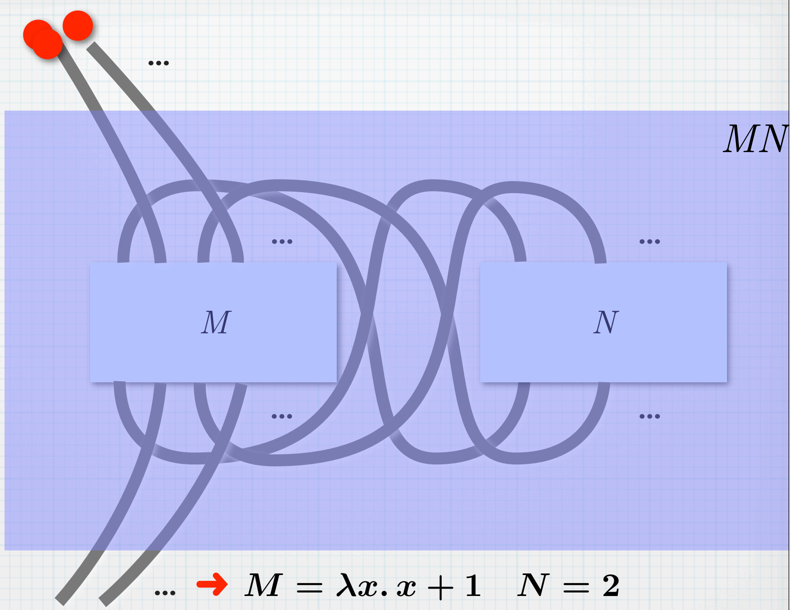



"parallel composition + hiding"
(cf. games)



$[\![MN]\!]$

$=$




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

# Categorical GoI

- Axiomatics of GoI in the categorical language
- Abstraction & genericity, which we exploit
- Our main reference (recommended!):
  - [AHS02] S. Abramsky, E. Haghverdi, and P. Scott, "Geometry of interaction and linear combinatory algebras," MSCS 2002
  - Especially its technical report version (Oxford CL), since it's more detailed



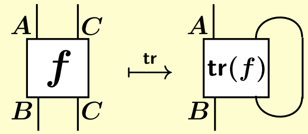
# The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]

Weak linear category Int($\mathbb{C}$)

Int-constr. [Joyal,Street&Verity96]




Categorical GoI [AHS02]

Linear combinatory algebra

- Applicative str. + combinators
- Model of untyped calculus

Realizability

Linear category

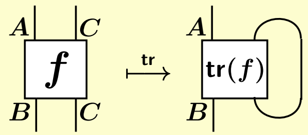
Model of typed calculus



# The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]




Categorical GoI [AHS02]

Linear combinatory algebra

- Applicative str. + combinators
- Model of untyped calculus

Realizability

Linear category

Model of typed calculus



# Linear Combinatory Algebra (LCA)

What we want (outcome)

**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.

- Model of untyped linear λ
- a ∈ A ≈ closed linear λ-term
- No **S** or **K** (linear!)
- Combinatory completeness: e.g.
$$\lambda xyz.\, zxy$$
designates elem. of A

What we use (ingredient)

# GoI situation

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

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

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

  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.

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



# GoI situation

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

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

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

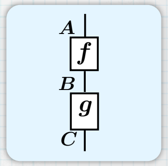
  Here $K_I$ is the constant functor into the monoidal unit $I$;

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

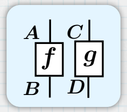
$$\begin{array}{c} j : U \otimes U \lhd U : k \\ I \lhd U \\ u : FU \lhd U : v \end{array}$$

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

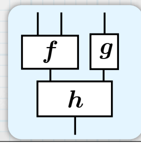
$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \xrightarrow{g \circ f} C}$$




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




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






# 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).

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

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

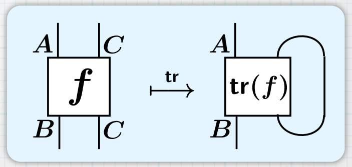
- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

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

- Traced monoidal category
  - "feedback"

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

that is

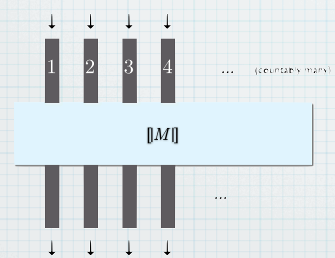




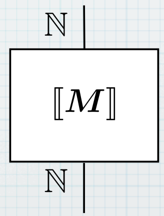

# String Diagram vs. "Pipe Diagram"

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

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




Pipe diagram

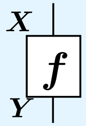



String diagram

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

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

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

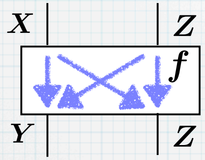



* is traced symmetric monoidal



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

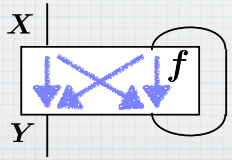
* Given $X + Z \xrightarrow{f} Y + Z$ in **Pfn**
*




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

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

* Trace operator:




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

* Execution formula
* Partiality is essential (infinite loop)



# GoI situation

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

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

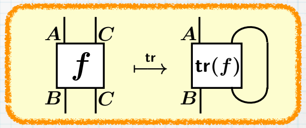
$$\begin{aligned} e &: FF \lhd F : e' && \text{Comultiplication} \\ d &: \mathrm{id} \lhd F : d' && \text{Dereliction} \\ c &: F \otimes F \lhd F : c' && \text{Contraction} \\ w &: K_I \lhd F : w' && \text{Weakening} \end{aligned}$$

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

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

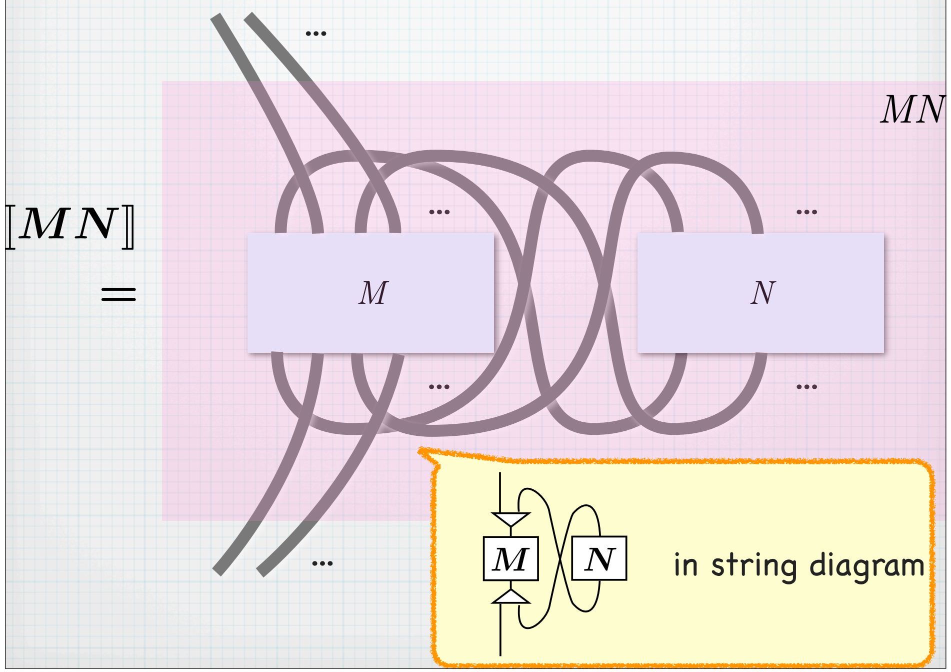
$$\begin{aligned} j : U \otimes U &\lhd U : k \\ I &\lhd U \\ u : FU &\lhd U : v \end{aligned}$$

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




  * Why for GoI?





# GoI situation

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

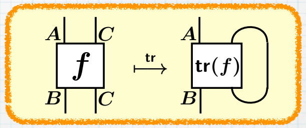
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

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

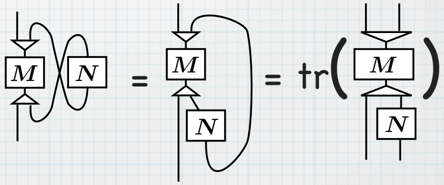
Here $K_I$ is the constant functor into the monoidal unit $I$;
- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

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

- 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 $(\mathbb{C}, F, U)$ 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).

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

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

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


**Defn.** (Retraction)
A *retraction* from $X$ to $Y$,

$$f : X \lhd Y : g \,,$$

is a pair of arrows

"embedding"

$$\mathrm{id} \circlearrowright X \underset{g}{\overset{f}{\rightleftarrows}} Y$$

"projection"

such that $g \circ f = \mathrm{id}_X$.

- 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

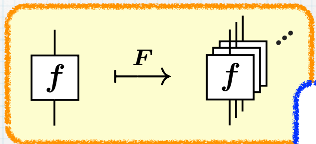
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

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

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

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


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




$$(n, x) \mapsto (n, f(x))$$

- Example in **Pfn**:

$$\mathbb{N} \cdot \_ : \quad \mathbf{Pfn} \longrightarrow \mathbf{Pfn}$$
$$\begin{array}{ccc} X & & \mathbb{N} \cdot X \\ \downarrow f & \longmapsto & \downarrow \mathbb{N} \cdot f \\ Y & & \mathbb{N} \cdot Y \end{array}$$



# GoI situation


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

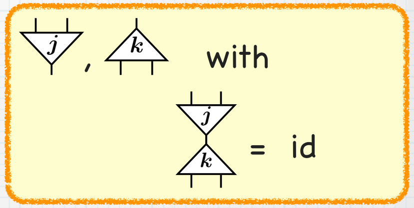
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

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

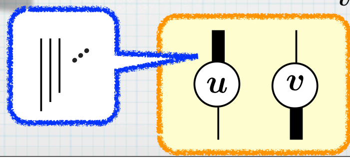
Here $K_I$ is the constant functor into the monoidal unit $I$;
- $U \in \mathbb{C}$ is an object (called *reflexive object*), equipped with the following retractions.

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


- The reflexive object $U$
  - Retr. $U \otimes U \underset{k}{\overset{j}{\rightleftarrows}} U$




  - Retr. $FU \underset{v}{\overset{u}{\rightleftarrows}} U$




Hasuo (Tokyo)

# GoI situation

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

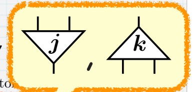
- $\mathbb{C} = (\mathbb{C}, \otimes, I)$ is a traced symmetric monoidal category (TSMC);
- $F : \mathbb{C} \to \mathbb{C}$ is a traced symmetric monoidal functor, equipped with the following retractions (which are monoidal natural transformations).

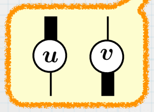
$$
\begin{array}{ll}
e : FF \lhd F : e' & \text{Comultiplication} \\
d : \mathrm{id} \lhd F : d' & \\
c : F \otimes F \lhd F : c' & \\
w : K_I \lhd F : w' &
\end{array}
$$

Here $K_I$ is the constant functo

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

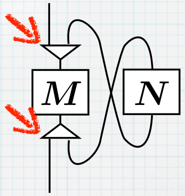
$$
\begin{array}{c}
j : U \otimes U \lhd U : k \\
I \lhd U \\
u : FU \lhd U : v
\end{array}
$$




* The reflexive object $U$

* Why for GoI?



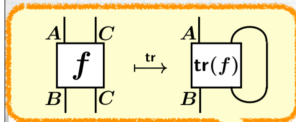

* Example in **Pfn**:

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

Hasuo (Tokyo)



# GoI situation: Summary




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

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

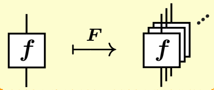
$$
\begin{array}{l}
e : FF \lhd F : e' \\
d : \mathrm{id} \lhd F : d' \\
c : F \otimes F \lhd F : c' \\
w : K_I \lhd F : w'
\end{array}
$$

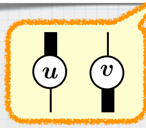
Here $K_I$ is the constant functor into t

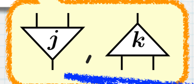
- $U \in \mathbb{C}$ is an object (called *reflexive obj*), equipped with the following retractions.

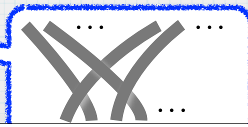
$$
\begin{array}{c}
j : U \otimes U \lhd U : k \\
I \lhd U \\
u : FU \lhd U : v
\end{array}
$$

For ! , via

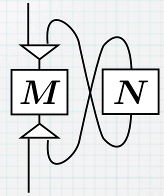


* Categorical axiomatics of the "GoI animation"




* Example:

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

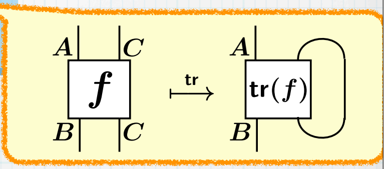
Hasuo (Tokyo)



# The Categorical GoI Workflow

Traced monoidal category $\mathbb{C}$

+ other constructs ➔ "GoI situation" [AHS02]




Categorical GoI [AHS02]

Linear combinatory algebra

* Applicative str. + combinators
* Model of untyped calculus

Realizability

Linear category

Model of typed calculus

Hasuo (Tokyo)

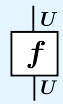


# Categorical GoI: Constr. of an LCA

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

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

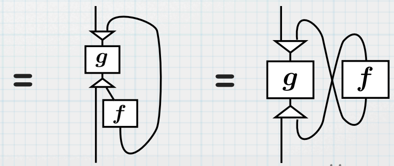
carries a canonical LCA structure.




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

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

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




Hasuo (Tokyo)

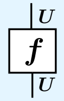
# Categorical GoI: Constr. of an LCA

**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.

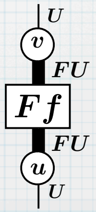
- Applicative str.
- ! operator
- Combinators B, C, I, ...



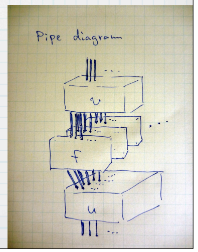

$\in \mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$

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

$=$




$=$






# Categorical GoI: Constr. of an LCA

- Combinator $\boldsymbol{B}xyz = x(yz)$

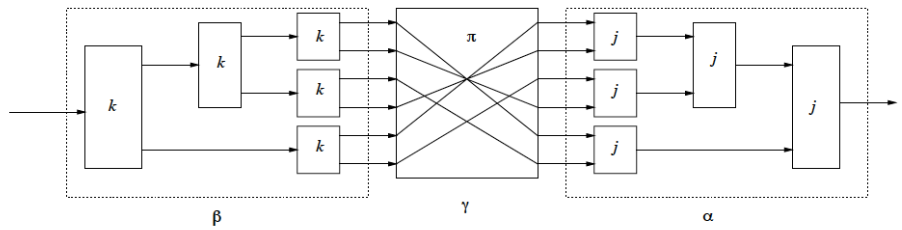


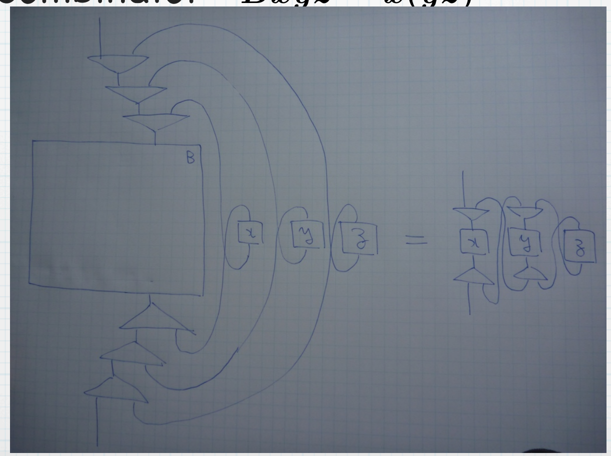

Figure 7: Composition Combinator B

from [AHS02]

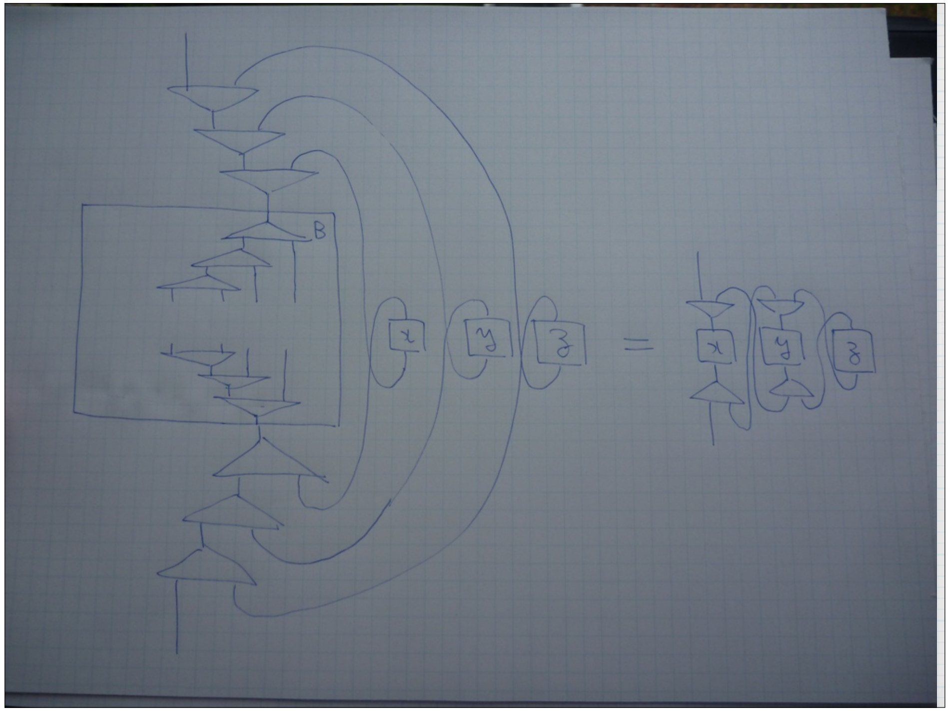


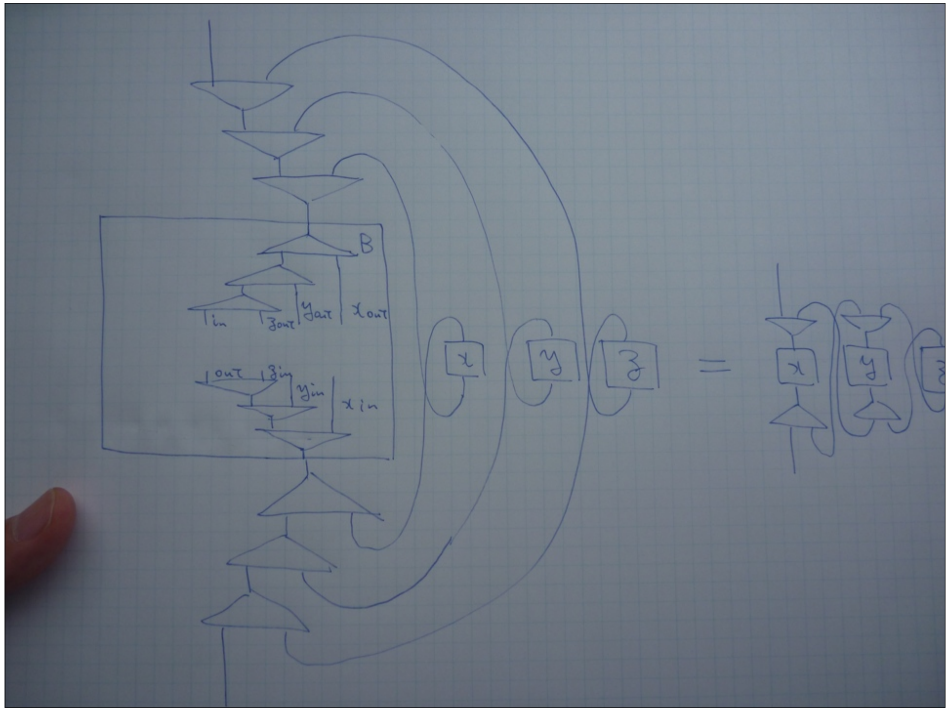
# Categorical GoI: Constr. of an LCA

- Combinator $\boldsymbol{B}xyz = x(yz)$

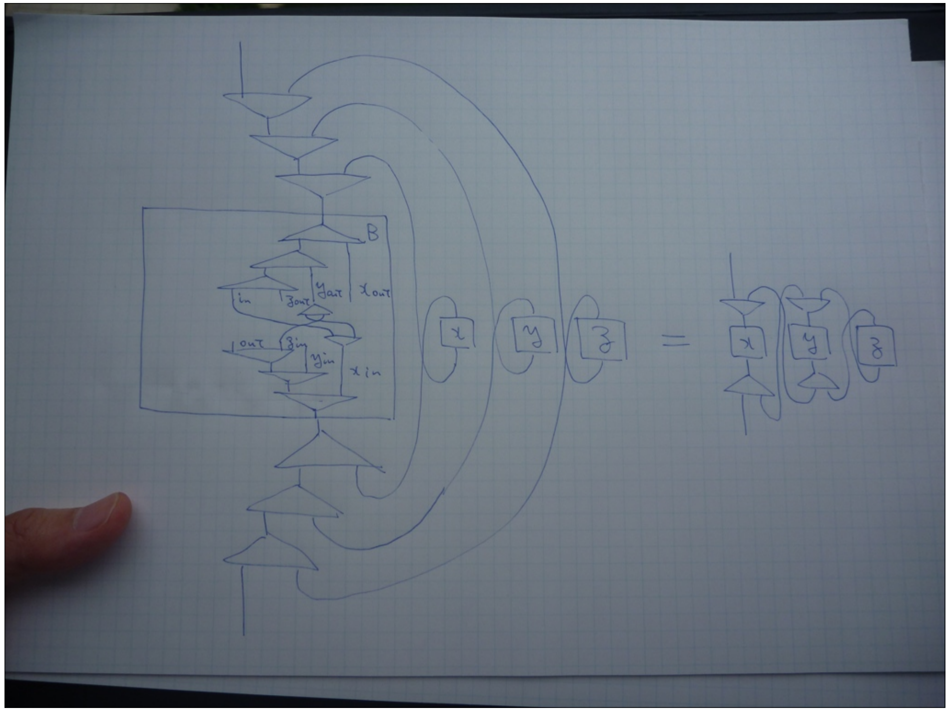




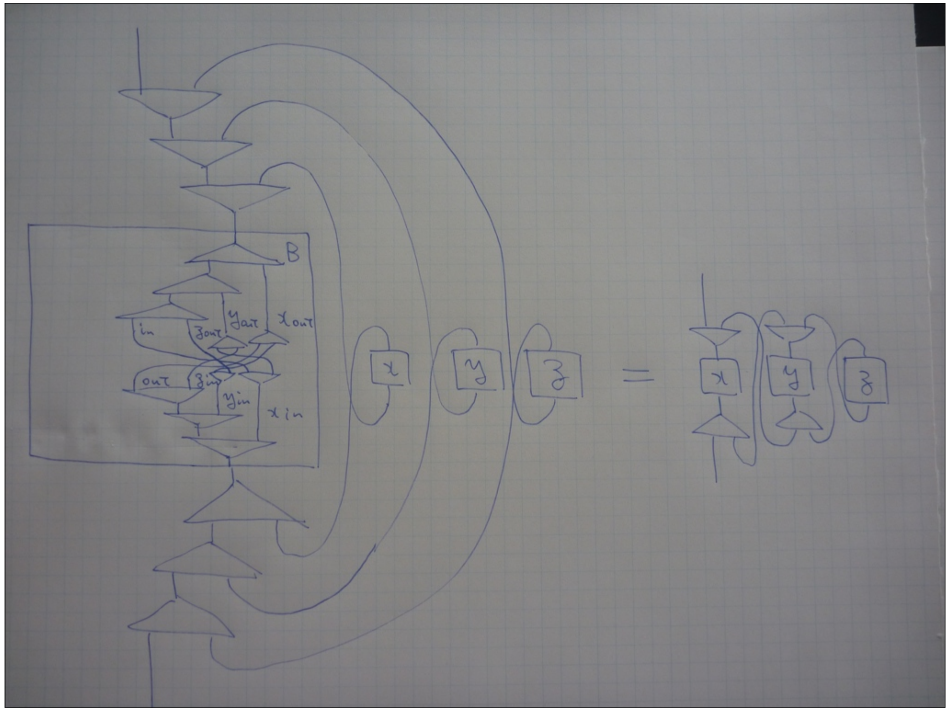














# Categorical GoI: Constr. of an LCA

- Combinator $Bxyz = x(yz)$

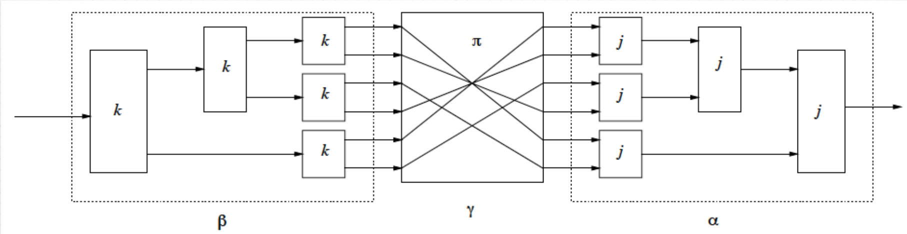



Figure 7: Composition Combinator B

from [AHS02]

Nice dynamic interpretation of (linear) computation!!

# Summary: Categorical GoI

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

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

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

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

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

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

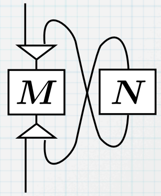
$$\mathbb{C}(\boldsymbol{U}, \boldsymbol{U})$$

carries a canonical LCA structure.



# Why Categorical Generalization?: Examples Other Than Pfn

- Strategy: find a TSMC!



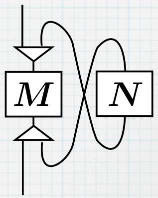

- "Wave-style" examples
  - $\otimes$ is Cartesian product(-like)
    - in which case,

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



# Why Categorical Generalization?: Examples Other Than Pfn

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




  - Examples:
    - Partial functions $\big( (\mathbf{Pfn}, +, 0),\ \mathbb{N} \cdot \_,\ \mathbb{N} \big)$
    - Non-det. functions (i.e. relations) $\big( (\mathbf{Rel}, +, 0),\ \mathbb{N} \cdot \_,\ \mathbb{N} \big)$
    - Probabilistic functions ("discrete stochastic relations") $\big( (\mathbf{DSRel}, +, 0),\ \mathbb{N} \cdot \_,\ \mathbb{N} \big)$



# Why Categorical Generalization?: Examples Other Than Pfn

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

- **Pfn** (partial functions)

  (Potential) non-termination

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

- **Rel** (relations)

  Non-determinism

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

- **DSRel**

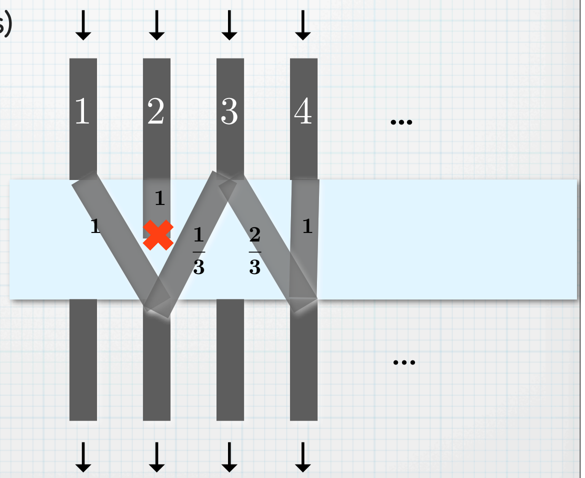
  Probabilistic branching

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

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

# Different Branching in The GoI Animation

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




Hasuo (Tokyo)



# Why Categorical Generalization?: Examples Other Than Pfn

* **Pfn** (partial functions)

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

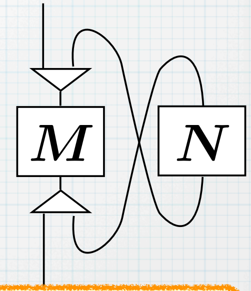
* **Rel** (relations)

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

* **DSRel**

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

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




Essential to have subdistribution, for infinite loops



# The Coauthor

* Naohiko Hoshino
  * DSc
    * Kyoto U. (JP), 2011
    * Supervisor: Masahito "Hassei" Hasegawa
  * Assist. Prof., RIMS, Kyoto U. (2011-)



# 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 $\mathcal{K}\ell(B)$

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

  * ➜ Generic theory of "trace semantics"

Hasuo (Tokyo)

## Why Catego Examples

$Kl(B)$ for different branching monads $B$

- **Pfn** (partial functions)

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

(Potential) non-termination

- **Rel** (relations)

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

Non-determinism

- **DSRel**

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

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

Probabilistic branching

Hasuo (Tokyo)



## Branching Monad: Source of Particle-Style GoI Situations

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

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

is

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

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

Monads in [Hasuo,Jacobs&Sokolova07]

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

Particle-style: trace via the execution formula

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

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

Hasuo (Tokyo)



## What is Fancy, Nowadays?

- Biology?
- Hybrid systems?
  - Both discrete and continuous data, typically in **cyber-physical systems** (CPS)
  - ➜ Our approach via **non-standard analysis** [Suenaga&Hasuo,ICALP11]
- Quantum?
  - Yes this worked!

Hasuo (Tokyo)

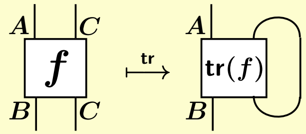
## The Categorical GoI Workflow

Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➔ "GoI situation" [AHS02]




Categorical GoI [AHS02]

Linear combinatory algebra

* Applicative str. + combinators
* Model of untyped calculus

Realizability

Linear category

Model of typed calculus



# Part 2

## Realizability:
## from Untyped to Typed



## Realizability

* Dates back to Kleene
* Cf. the Brouwer-Heyting-Kolmogorov (BHK) interpretation
  * A p'f of A∧B is a pair: (p'f of A, p'f of B)
  * A p'f of A→B is a function carrying (p'f of A) to (p'f of B)
  * Proof = "realizer"



## Realizability

* Our technical view on realizability: a construction
  * from a combinatory algebra,
  * of a categorical model of a typed calculus
* Here: construct a linear category from an LCA
* References:

  * [AL05] S. Abramsky and M. Lenisa, "Linear realizability and full completeness for typed lambda-calculi," APAA 2005.
  * [Hos07] N. Hoshino, "Linear realizability," CSL 2007.

# Realizability

* Either by ω-sets (intuitive) or by PERs (tech. convenient)

Could as well be a **partial combinatory algebra**. Its examples:

* $\mathbb{N}$ with n·m = comp(n,m)
* { closed λ-terms }

**Defn.**
Given an LCA $A$, an $\omega$-*set* is a pair

$$(\,S,\quad r : S \to \mathcal{P}_{+}(A)\,)$$

where

- $S$ is a set;
- for each $x \in S$, the nonempty subset $r(x) \subseteq A$ is the set of *realizers*.

a ∈ r(x) :

* "realizes" x, or
* "witnesses existence of" x



# Realizability

**Defn.**
A *partial equivalence relation (PER)* $X$ is a transitive and symmetric relation on $A$.

$$\begin{aligned}|X| &:= \{a \mid (a,a) \in X\}\\ &= \{a \mid \exists b.\,(a,b) \in X\}\\ &= \{a \mid \exists b.\,(b,a) \in X\}\end{aligned}$$

is the *domain* of $X$.

* PER = eq. rel. − refl.
* An eq. rel. when restricted to $|X|$
* PER to ω-set:

$$\big(\,|X|/X,\quad |X|/X \xrightarrow{r} \mathcal{P}_{+}(A)\,\big)$$
with $\quad [a] \overset{r}{\longmapsto} \{b \mid (a,b) \in X\}$

* Also: ω-set to PER



# PER$_A$: The Category of PERs

* **Obj.** A PER $X$ on $A$
* **Arr.** The homset is

All the valid **codes** c (well-dfd?)

Modulo "the same function"

$$\mathbf{PER}_A(X,Y) = \Big\{c \in A \,\Big|\, (x,x') \in X \Longrightarrow (cx, cx') \in Y\Big\}$$

$$\Big\{(c,c') \,\Big|\, \forall x \in |X|.\ (cx, c'x) \in Y\Big\}$$

* Thus: $[c] : X \longrightarrow Y$ (with $c \in A$)
* Often put: $\mathbf{PER}_A(X,Y) = \Big\{(c,c') \,\Big|\, (x,x') \in X \Longrightarrow (cx, c'x') \in Y\Big\}$



# Type Constructors in PER$_A$

with full K: K$xy$=$x$

**Thm.** ([AL05])
If $A$ is an affine LCA, then $\mathbf{PER}_A$ is a linear category. Furthermore, $\mathbf{PER}_A$ has finite products and coproducts.

* Linear category [Benton&Wadler,LICS'96][Bierman,TLCA'95]
  * Categorical model of linear logic/linear λ, with
    * Monoidal closed with $\boxtimes, \mathbf{I}, \multimap$
    * Linear exponential comonad **!**

Not ⊗, for distinction

# Type Constructors in $\mathbf{PER}_A$

* How to get operators $\boxtimes, \times, +, \ldots$
  * Like "programming in untyped λ"!

| | | |
|---|---|---|
| $\underline{n} := \lambda fx.f(f\cdots(fx)\cdots)$ | | Church numeral |
| $\overline{\mathsf{K}} := \mathsf{KI}$ | | |
| $\mathsf{P} := \lambda xyz.zxy$ | | Paring |
| $\mathsf{P_l} := \lambda w.w\mathsf{K}$ | | Left projection |
| $\mathsf{P_l} := \lambda w.w\overline{\mathsf{K}}$ | | Right projection |

$\mathsf{P_l}(\mathsf{P}xy) = x$
$\mathsf{P_r}(\mathsf{P}xy) = y$

* Cf. Combinaroty completeness



# Type Constructors in $\mathbf{PER}_A$

$$\frac{X \in \mathbf{PER}_A}{X \subseteq A \times A,\ \text{sym., trans.}}$$

multiplicative and

$$X \boxtimes Y := \left\{ (\mathsf{P}xy, \mathsf{P}x'y') \;\middle|\; (x,x') \in X \wedge (y,y') \in Y \right\}$$

$$X \times Y := \Big\{ \big(\mathsf{P}k_1(\mathsf{P}k_2u),\ \mathsf{P}k'_1(\mathsf{P}k'_2u')\big) \;\Big|\; (k_1u, k'_1u') \in X \wedge (k_2u, k'_2u') \in Y \Big\}$$

additive and

CPS-style. $k_1$, $k_2$: "access methods"

$$!X := \left\{ (!x, !x') \;\middle|\; (x,x') \in X \right\}$$

$$X + Y := \left\{ (\mathsf{PK}x, \mathsf{PK}x') \;\middle|\; (x,x') \in X \right\} \cup \left\{ (\mathsf{PK}y, \mathsf{PK}y') \;\middle|\; (y,y') \in Y \right\}$$

$$X \multimap Y := \left\{ (c,c') \;\middle|\; (x,x') \in X \Longrightarrow (cx, c'x') \in Y \right\}$$



# Summary: Realizability

**Affine LCA** $A$

$a \cdot b, \quad !a, \quad \mathsf{B}, \mathsf{C}, \mathsf{I}, \ldots$

$\longmapsto$

**Linear category** $\mathbf{PER}_A$

* 
$$\begin{array}{ccc} X & \xrightarrow{[c]} & Y \\ [a] & \longmapsto & [c \cdot a] \end{array} \qquad (a, c \in A)$$

* Type constructors via "programming in untyped λ"
  * Symmetric monoidal closed $\boxtimes, \mathbf{I}, \multimap$
  * Finite product, coproduct

Not ⊗, for distinction



# The Categorical GoI Workflow

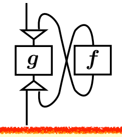
Branching monad B

Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs → "GoI situation" [AHS02]

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

$g \cdot f =$ 

Categorical GoI [AHS02]

Linear combinatory algebra

* Applicative str. + combinators
* Model of untyped calculus

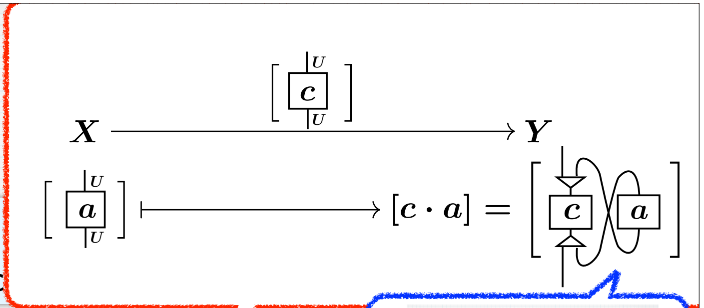
Realizability

Linear category

Model of typed calculus

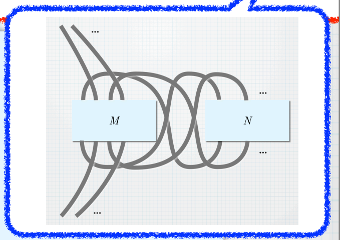
## Affine LCA

$a \cdot b, \quad !a, \quad$ B, C




## Linear category $\mathbf{PER}_A$

* $X \xrightarrow{[c]} Y$, $[a] \longmapsto [c \cdot a]$ $(a,c \in A)$




* Type constructors via "programming in untyped λ"
  * Symmetric monoidal closed $\boxtimes, \mathbf{I}, \multimap$
  * Finite product, coproduct

Not ⊗,
for distinction

suo (Tokyo)



# Time to Wake Up!!

It's time to save them. WWF



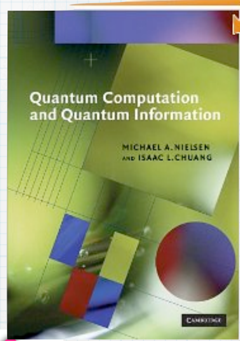
# Part 3

## Quantum Computation in 5 min.

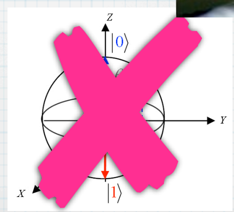


# What You Need to Know

Click to LOOK INSIDE!




* Not much, really!
* Our principal reference:
  * M.A. Nielsen and I.L. Chuang. **Quantum Computation and Quantum Information**. CUP, 2000
  * Its Chap. 3 & Chap. 8
    * Hilbert space formulation
    * Quantum operation formalism (Kraus)
  * No need for the Bloch sphere




Hasuo (Tokyo)

# Some Principles


- A state of a 1-qubit system = a normalized vector

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \in \mathbb{C}^2$$

- with $\| |\varphi\rangle \|^2 = |\alpha|^2 + |\beta|^2 = 1$
- Various notations for base:

$\{\, |0\rangle, |1\rangle \,\}, \ \{\, |+\rangle, |-\rangle \,\}, \ \{\, |\uparrow\rangle, |\downarrow\rangle \,\}, \ldots$

Hasuo (Tokyo)



# Some Principles


- Composed system: $\otimes$, not $\times$.
  - not $\mathbb{C}^2 \times \mathbb{C}^2 \times \mathbb{C}^2 \cong \mathbb{C}^6$, with base $\left\{ \begin{matrix} |0_1\rangle & |0_2\rangle & |0_3\rangle \\ |1_1\rangle & |1_2\rangle & |1_3\rangle \end{matrix} \right\}$
  - but $\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^8$, with base $\left\{ \begin{matrix} |000\rangle & |001\rangle & |010\rangle & |011\rangle \\ |100\rangle & |101\rangle & |110\rangle & |111\rangle \end{matrix} \right\}$

Hasuo (Tokyo)



# Some Principles

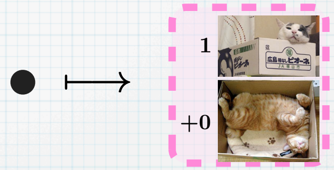

- Composed system: $\otimes$, not $\times$.
  - Source of power of quantum comp./comm.
    - N-qubit → $2^N$-dim (not 2N-dim)
    - Entanglement; superposition

Hasuo (Tokyo)

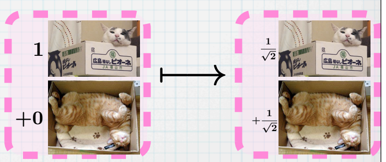


# Three Quantum Primitives

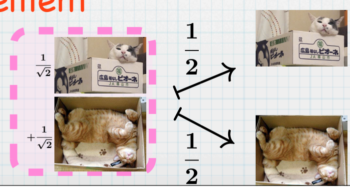
- Preparation




- Unitary transformation




- Measurement




Hasuo (Tokyo)

# Three Quantum Primitives


- Preparation
  - Creates/"prepares" a quantum state (typically $|0\rangle$)



# Three Quantum Primitives

- Unitary transformation



$$\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \quad \stackrel{U}{\longmapsto} \quad U\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

- Unitary matrix: $UU^\dagger = U^\dagger U = \mathcal{I}$
  - Invertible. "Rotation"
- Also for N-dim systems (of course)



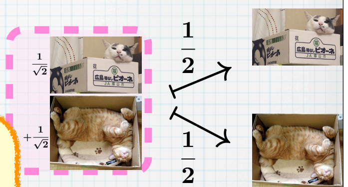
# Three Quantum Primitives

- Measurement

When one measures

$$\alpha|0\rangle + \beta|1\rangle$$

state reduction

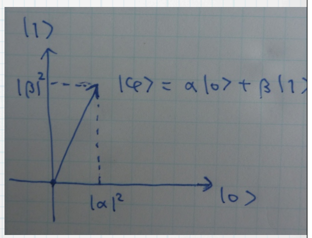



- $|0\rangle$ is observed, and
- the state becomes $|0\rangle$

with prob. $|\alpha|^2$

- $|1\rangle$ is observed, and
- the state becomes $|1\rangle$

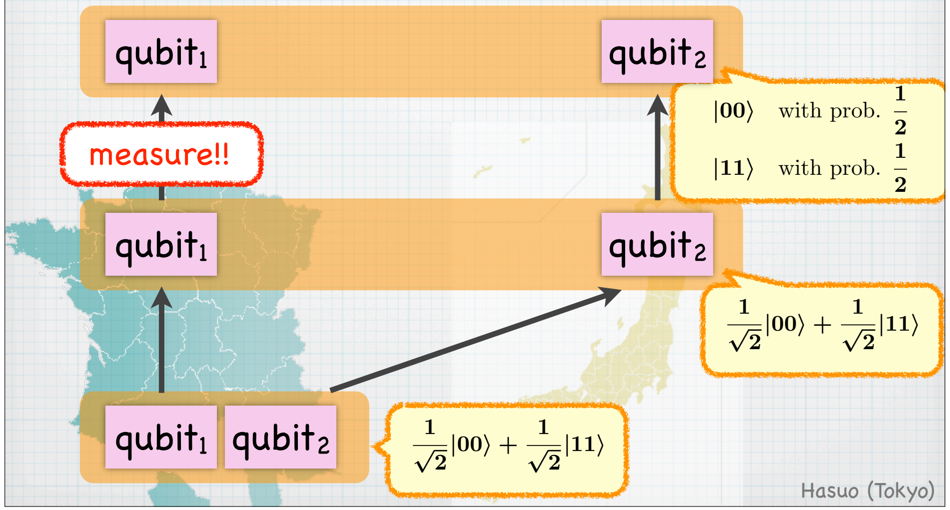
with prob. $|\beta|^2$




Also: for other dimensions, bases



# Entanglement



# Density Matrix, Quantum Operation

* Advanced, mathematically convenient formalisms
* State vector ➜ density matrix
  * Use $|\varphi\rangle\langle\varphi|$ in place of $|\varphi\rangle$
  * Can also represent **mixed states**, e.g. $|00\rangle$ with prob. $\frac{1}{2}$, $|11\rangle$ with prob. $\frac{1}{2}$
* Quantum operation (QO) [Kraus]
  * {QOs} = {any combinations of preparation, Unitary transf., measurement}
  * But **no** classical control (like case-distinction)
  * Used in [Selinger,MSCS'04] and other

Hasuo (Tokyo)



# Density Matrix, Quantum Operation

**Defn.**

- An *$m$-dimensional density matrix* is an $m \times m$ matrix $\rho \in \mathbb{C}^{m\times m}$ which is positive and satisfies $\mathbf{tr}(\rho) \in [0,1]$.
  – Notation: $D_m = \{m\text{-dim. density matrices}\}$
- A *quantum operation (QO)* is a mapping $\mathcal{E} : D_m \to D_n$ subject to the following axioms.
  1. (Trace condition) $\mathbf{tr}[\mathcal{E}(\rho)] \in [0,1]$ for any $\rho \in D_m$.
  2. (Linearity) Let $(\rho_i)_{i\in I}$ be a family of $m$-dim. density matrices; and $(p_i)_{i\in I}$ be a probability subdistribution (meaning $\sum_i p_i \le 1$). Then: $\mathcal{E}(\sum_{i\in I} p_i\rho_i) = \sum_{i\in I} p_i\mathcal{E}(\rho_i)$ .
  3. (Complete positivity) An arbitrary "extension" of $\mathcal{E}$ of the form $\mathcal{I}_k \otimes \mathcal{E} : M_k \otimes M_m \to M_k \otimes M_n$ carries a positive matrix to a positive one.

  – Notation: $\mathbf{QO}_{m,n} = \{\text{QOs from } m\text{-dim. to } n\text{-dim.}\}$

* For specialists: we allow trace ≤ 1
* So that **probabilities** are implicitly carried by density matrices

Hasuo (Tokyo)



# Quantum Computation: Summary

* A quantum state = a vector $|\varphi\rangle$

Generalized to density matrix

* Composition by ⊗
  ➜ Dimension grows exponentially
* Three primitives:
  * Preparation
  * Unitary transformation
  * Measurement (➜ st. reduction)

Unified to quantum operation (QO)

Hasuo (Tokyo)



# Part 4

# Quantum GoI

# The Language $q\lambda_\ell$

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

Hasuo (Tokyo)

# The Language

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

2-qbit ≅ qbit ⊗ qbit

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

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

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

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

Recursion

Quantum primitives

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

Hasuo (Tokyo)

Implicit linearity tracking via subtyping <:
e.g. !A <: A, !A <: !!A
(following [Selinger-Valiron'09])

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

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

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

Measurements

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

Bookkeeping (due to ⊗ vs. ⊠)

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

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

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

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

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

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

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

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

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

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

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

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

Hasuo (Tokyo)

# Operational Semantics

$$\begin{aligned}
& E[\,(\lambda x^A.M)V\,] \to_1 E[\,M[V/x]\,] \\
& E[\,\mathtt{let}\,\langle x^A, y^B\rangle = \langle V, W\rangle\,\mathtt{in}\,M\,] \to_1 E[\,M[V/x, W/y]\,] \\
& E[\,\mathtt{let}\,* = *\,\mathtt{in}\,M\,] \to_1 E[\,M\,] \\
& E[\,\mathtt{match}\,(\mathtt{inj}_\ell^B V)\,\mathtt{with}\,(x^{!^n A} \mapsto M \mid y^{!^n B} \mapsto N)\,] \\
& \qquad \to_1 E[\,M[V/x]\,] \\
& E[\,\mathtt{match}\,(\mathtt{inj}_r^A V)\,\mathtt{with}\,(x^{!^n A} \mapsto M \mid y^{!^n B} \mapsto N)\,] \\
& \qquad \to_1 E[\,N[V/y]\,] \\
& E[\,\mathtt{letrec}\,f^{A \multimap B} x = M\,\mathtt{in}\,N\,] \\
& \qquad \to_1 E[\,N[\lambda x^A.\mathtt{letrec}\,f^{A \multimap B} x = M\,\mathtt{in}\,M/f]\,] \\
& E[\,\mathtt{meas}_i^{n+1}(\mathtt{new}\,\rho)\,] \to_1 E[\,\langle\,\mathtt{tt},\,\mathtt{new}\,\langle 0_i|\rho|0_i\rangle\,\rangle\,] \\
& E[\,\mathtt{meas}_i^{n+1}(\mathtt{new}\,\rho)\,] \to_1 E[\,\langle\,\mathtt{ff},\,\mathtt{new}\,\langle 1_i|\rho|1_i\rangle\,\rangle\,] \\
& E[\,\mathtt{meas}_1^1(\mathtt{new}\,\rho)\,] \to_{\langle 0|\rho|0\rangle} E[\,\mathtt{tt}\,] \\
& E[\,\mathtt{meas}_1^1(\mathtt{new}\,\rho)\,] \to_{\langle 1|\rho|1\rangle} E[\,\mathtt{ff}\,] \\
& E[\,U(\mathtt{new}\,\rho)\,] \to_1 E[\,\mathtt{new}\,(U\rho)\,] \\
& E[\,\mathtt{cmp}_{m,n}\langle \mathtt{new}\,\rho, \mathtt{new}\,\sigma\rangle\,] \to_1 E[\,\mathtt{new}\,(\rho \otimes \sigma)\,]
\end{aligned}$$

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

Hasuo (Tokyo)

# The Language qλℓ

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



# The Categorical GoI Workflow

Branching monad B

↓ Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs ➜ "GoI situation" [AHS02]

↓ Categorical GoI [AHS02]

Linear combinatory algebra

↓ Realizability

Linear category

Quantum branching monad → Quantum TSMC → Quantum LCA → Model of quantum language



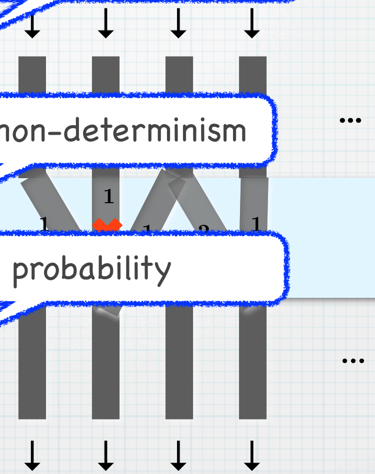
# Different Branching in
# The ... n

- **Pfn** (partial functions)
  - Pipe can be stuck
- **Rel** (relations)
  - Pipe can branch
- **DSRel**
  - Pipe is probabilistically branched

$Kl(\mathcal{L})$, non-termination

$Kl(\mathcal{P})$, non-determinism

$Kl(\mathcal{D})$, probability





# Quan...
# Geometry o...

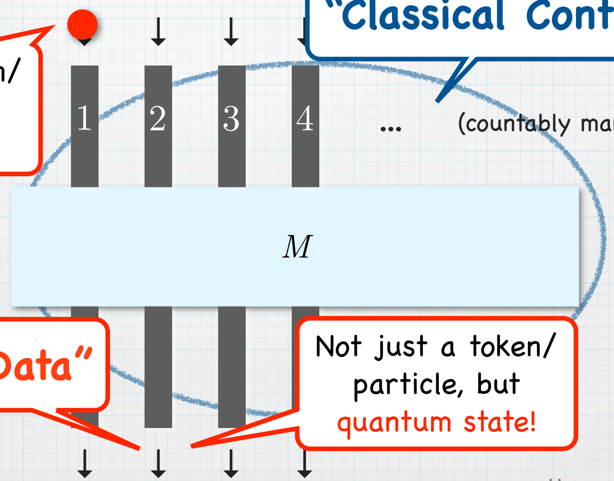
- "in which pipe"
- (measurement ➜ case-distinction) leads a token to different pipes

"Classical Control"

Not just a token/ particle, but quantum state!

(countably many)

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



"Quantum Data"

Not just a token/ particle, but quantum state!

# The Quantum Branching Monad

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

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

(trace of matrix ≈ probability)

* Compare with

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

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

Hasuo (Tokyo)



# The Quantum Branching Monad

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

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

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

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

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

**Any opr. on quantum data:** combination of

- preparation
- unitary transf.
- measurement

* Subject to the **trace condition**

Hasuo (Tokyo)



# The Quantum Branching Monad

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

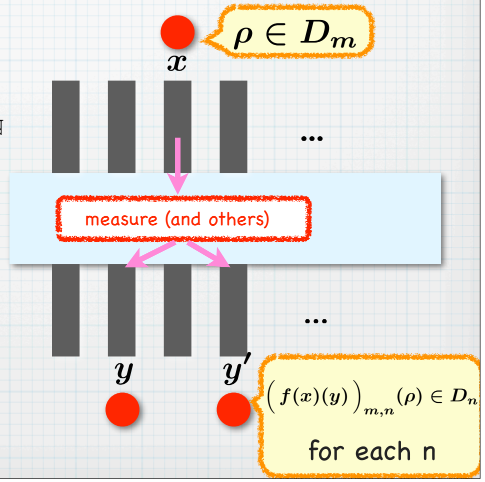
$$\sum_{y\in Y}\sum_{n\in\mathbb{N}} \mathbf{tr}\bigl[(c(y))_{m,n}(\rho)\bigr] \le 1\ , \quad \forall m\in\mathbb{N},\ \forall\rho\in D_m.$$

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

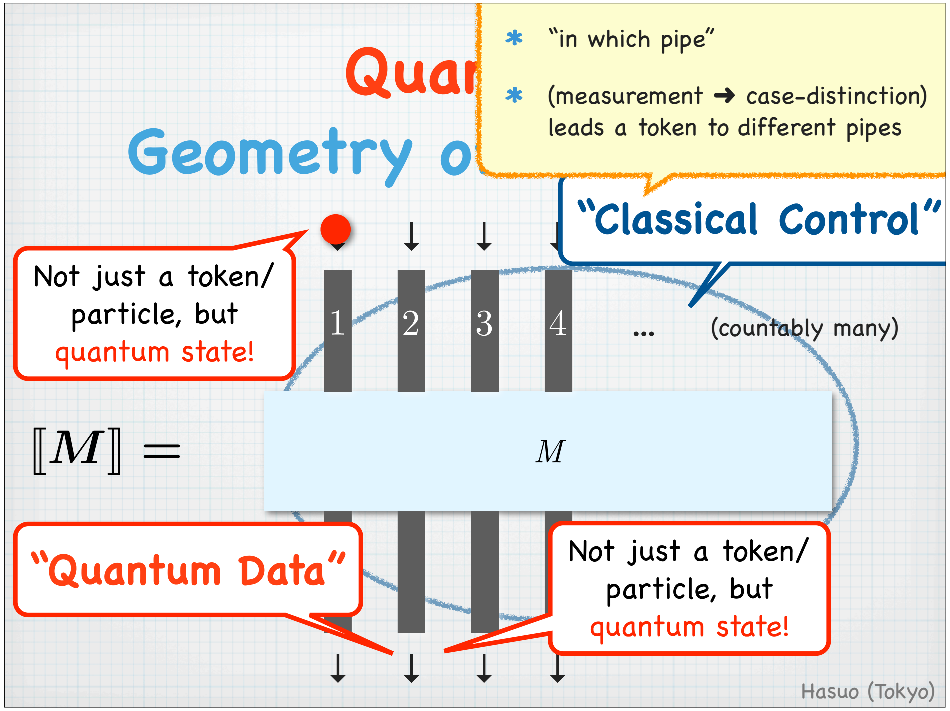
* Given $x\in X$ (entrance), $y\in Y$ (exit), $m\in\mathbb{N}$ (in. dim.), $n\in\mathbb{N}$ (out. dim.) determines a quantum operation $(f(x)(y))_{m,n}$

* **trace cond.:**

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





# Indeed...

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



# End of the Story?

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



## Conclusion: the Cat

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

Branching monad B

↓ Coalgebraic trace semantics

Traced monoidal category $\mathbb{C}$

+ other constructs → "GoI situation" [AHS02]

↓ Categorical GoI [AHS02]

Linear combinatory algebra

↓ Realizability

Linear category

Quantum branching monad

↓

Quantum TSMC

↓

Quantum LCA

↓

Model of quantum language