

プログラム意味論とトポロジー  
— 再帰，相互作用，結び目 —

長谷川 真人  
京都大学数理解析研究所

日本数学会秋季総合分科会  
2010年9月24日

プログラム意味論と低次元トポロジー  
— 再帰，相互作用，結び目 —

長谷川 真人  
京都大学数理解析研究所

日本数学会秋季総合分科会  
2010年9月24日

プログラム意味論と量子トポロジー  
— 再帰，相互作用，結び目 —

長谷川 真人  
京都大学数理解析研究所

日本数学会秋季総合分科会  
2010年9月24日

# I. 序論と概要

## II. 再帰プログラムの不動点意味論

### III. 再帰プログラムの幾何

### IV. 相互作用の幾何

### V. まとめと展望

## 背景：数学 vs 計算機科学

“Computer Science Contradicts Mathematics” (P. Freyd)

数学の世界：

- 主流は線形，連続，可換
- いたるところに対称性や双対性が
- 自己言及は避ける

計算の世界：

- 本質的に非線形，離散，非可換
- たいてい非対称
- 自己言及（再帰，フィードバック）は基本的

## 背景：数学 vs 計算機科学

実際には，計算機科学も，主流派の「筋の良い」数学から，多大な影響と恩恵を受けている：

- 離散から連続へ：計算可能性を連続性で捉える  
(領域理論, D. Scott, 1970's- )
- 非線形から線形へ：非線形な計算や推論を線形なものに分解  
(線形論理, J.-Y. Girard, 1987 )
- 非対称から対称へ：(自己)双対性を持つ計算の世界の構成  
(たとえば Chu spaces, V. Pratt, 1995- )

圏論はこれらの発展において非常に有効に用いられてきた

## プログラミング言語の意味論

### 目的

複雑なプログラムの背後にある数学的な構造をつきとめ、分析することによって、プログラムに関して成り立つ本質的な原理を導く

### 応用

- プログラムを見通し良く **設計**
- プログラムが仕様どおりに動くことを **検証**
- プログラムをより効率の良いもの・わかりやすいものに **改良**

### 方針

- 適切な数学モデルを用いてプログラムの構造を **抽象化**
- 基礎理論研究の立場：本質，普遍性，簡潔性を重視

## プログラムにどう立ち向かうか

プログラム — コンピュータが行なう仕事（コンピューテーション）を直接的・具体的に記述する最強・最良の手段の一つ

プログラムの制御構造やデータ構造は  
コンピューテーションに明確な「かたち」を与える

けれども，コンピューテーションそのものについて考えるためには，  
プログラムに直接立ち向かうことは常にベストの方法とはいえない

- プログラムはたいてい複雑で，巨大である
- プログラムは，たいてい冗長である
- プログラムは，本質的に理解困難である



## プログラミング言語の意味論

プログラムのあらゆるコンピュテーションの本質的な意味を  
システムティックに解釈する理論

プログラムを直接読んだり実行したりするかわりに、  
適切な**抽象化**によって得られた、より扱いやすい情報を用いて、  
そのふるまいを理解する

適切な抽象化は、問題の本質を浮き彫りにする

## 他分野からの引用

「幾何学は与えられた変換群に属する任意の変換で不変な  
図形の性質を研究する学問である」  
(クライン)

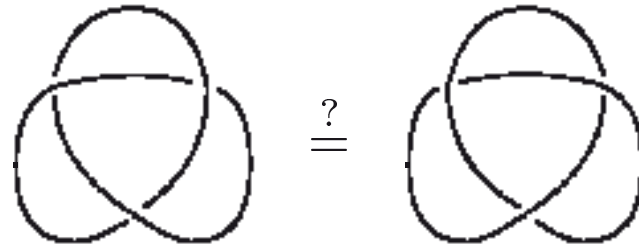
「物理学の学習はありのままに世界をみない手法を  
身につけるための修行である」  
(佐藤文隆)

プログラミング言語の意味論においても、同様のことがいえるはず

## 結び目の理論との比較

数学者の問題：

「与えられたふたつの結び目は同じものだろうか？」



基本方針：結び目の，イソトピーに対して不変な数学的な量を考察  
(結び目の不変量)

計算機科学者の問題：

「与えられたふたつのプログラムは同じ働きをするだろうか？」

基本方針：プログラムの実行に関して不変な数学的な量を考察  
(プログラミング言語の表示的意味論)

## 結び目の理論との比較

タングルの不変量：

$\text{Tangle} \xrightarrow{\text{テンソル関手}} \text{適当なりボン圏 (ex. 量子群の表現の圏)}$

ここで  $\text{Tangle}$  は ( 枠付き ) タングルの圏

ラムダ計算の表示的意味論：

$\text{Lambda} \xrightarrow{\text{カルテシアン閉関手}} \text{適当なカルテジアン閉圏}$   
( ex. 完備半順序集合の圏, 後述 )

ここで  $\text{Lambda}$  は ( 型付き ) ラムダ計算の型と項 ( の  $\beta\eta$  同値類 ) の圏

不変量  $\approx$  意味論  $\approx$  何らかの構造を保存する関手を与えること

## 結び目の理論とプログラム意味論のただならぬ関係

量子トポロジーの発展と並行して、リボン圏やモジュラー圏をはじめとする圏論の枠組みが整備された（80年代中頃～）。

一見、それらは、プログラム意味論で登場する圏とは似ても似つかない。カルテジアン閉なりボン圏は自明な圏だけ – そもそも両立不可能。

ところが、プログラム意味論においても、リボン圏の充満部分モノイダル圏が、昔から盛んに（誰にも知られることなく）用いられていたことが、90年代半ばに発覚した：

- ・ 再帰プログラムの不動点意味論（70年代～）
- ・ 相互作用の幾何（Girard, 80年代後半～）

それ以来、これら量子トポロジーに由来するモノイダル圏は、プログラム意味論においてとても大切な存在となった

## この講演の目的

プログラミング言語の主要な制御構造である再帰の意味論，および双方向の情報の流れを伴う計算を扱う相互作用の幾何において  
トレース付きモノイダル圏（ $\approx$  リボン圏の充満部分モノイダル圏）が果たしている役割について紹介します．

- I 序論と概要
- II 再帰プログラムの不動点意味論（背景）
- III 再帰プログラムの幾何
- IV 相互作用の幾何
- V まとめと展望

- I. 序論と概要
- II. 再帰プログラムの不動点意味論
- III. 再帰プログラムの幾何
- IV. 相互作用の幾何
- V. まとめと展望

## 再帰プログラムの不動点意味論

再帰プログラム, たとえば階乗を計算するプログラム  $\text{fact} : \text{int} \rightarrow \text{int}$

$$\text{fact}(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } x \times \text{fact}(x-1)$$

は, 以下の汎関数  $F : (\text{int} \rightarrow \text{int}) \rightarrow (\text{int} \rightarrow \text{int})$  の不動点として理解することができる:

$$F(f)(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } x \times f(x-1)$$

すなわち  $\text{fact} = F(\text{fact})$  となる.

## 不動点意味論

再帰プログラムを, 適切な数学構造の上の写像の不動点として抽象化して解釈 (表示的意味論・領域理論, 1970ごろ~)



## 再帰プログラムの例

McCarthy の ' 91 関数 '

$$f(x) \equiv \text{if } x > 100 \text{ then } x - 10 \text{ else } f(f(x + 11))$$

竹内郁雄の ' たらいまわし関数 '

$$\begin{aligned} t(x, y, z) \equiv \\ \text{if } x \leq y \text{ then } y \\ \text{else } t(t(x-1, y, z), t(y-1, z, x), t(z-1, x, y)) \end{aligned}$$

( cf. 数理解析研究所の大学院入試問題 , 2005 )

$$\begin{aligned}
& \text{fact}(4) \\
(\text{fact} = F(\text{fact})) &= F(\text{fact})(4) \\
(F \text{ の定義}) &= \text{if } 4 = 0 \text{ then } 1 \text{ else } 4 \times \text{fact}(4-1) \\
(4 \neq 0) &= 4 \times \text{fact}(3) \\
(\text{fact} = F(\text{fact})) &= 4 \times F(\text{fact})(3) \\
(F \text{ の定義}) &= 4 \times (\text{if } 3 = 0 \text{ then } 1 \text{ else } 3 \times \text{fact}(3-1)) \\
(3 \neq 0) &= 4 \times (3 \times \text{fact}(2)) \\
&\dots \\
&= 4 \times (3 \times (2 \times (1 \times 1))) \\
&= 24
\end{aligned}$$

## 領域理論

領域理論 (domain theory) では

- データの集まりを  
適当な条件 (完備性) を満たす順序集合 (データ領域) として
- プログラムを  
領域のあいだの適当な条件 (連続性) を満たす関数として

抽象化して表現する .

イメージ :

計算可能な関数全体  $\subset$  連続な関数全体  $\subset$  関数全体

( 離散的な複雑な現象を , 連続的な世界で筋良くとらえる )

## 完備半順序集合

定義 半順序集合  $(X, \sqsubseteq)$  が最小の要素を持ち,  $X$  の任意の可算単調列

$$x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq x_3 \sqsubseteq \dots$$

が上限  $\bigsqcup x_i$  を持つとき,  $(X, \sqsubseteq)$  は完備半順序集合  
(complete partial order, CPO) であるという.

CPO  $(X, \sqsubseteq)$  の最小元を  $\perp_X$  または  $\perp$  であらわす.

直観的には,  $\perp$  は, 「未定義」「止まらない計算」をあらわしている.

## 完備半順序集合の例

整数上の部分関数（結果が未定義かも知れない関数）全体の集合  $[\mathbf{Z} \rightarrow \mathbf{Z}]$  に，以下のような順序  $\sqsubseteq$  を定める：

$$f \sqsubseteq g \iff \begin{array}{l} \text{任意の } x \in \mathbf{Z} \text{ について} \\ f(x) \text{ が定義されているならば } g(x) \text{ も定義されていて} \\ \text{しかも } f(x) = g(x) \end{array}$$

この順序に関して  $[\mathbf{Z} \rightarrow \mathbf{Z}]$  は完備半順序集合になる．最小元は，任意の  $x \in \mathbf{Z}$  について対応する値が未定義であるような部分関数である．また， $f_0 \sqsubseteq f_1 \sqsubseteq f_2 \sqsubseteq \dots$  について，

$$\left( \bigsqcup f_i \right)(n) = \begin{cases} f_k(n) & \text{ある } k \text{ について } f_k(n) \text{ が定義されているとき} \\ \text{未定義} & \text{その他の場合} \end{cases}$$

## 連続関数

CPO  $(X, \sqsubseteq)$  と  $(Y, \sqsubseteq)$  について, 関数  $f : X \rightarrow Y$  が連続であるとは,

1.  $x_1 \sqsubseteq x_2$  なら  $f(x_1) \sqsubseteq f(x_2)$
2. 可算単調列  $x_0 \sqsubseteq x_1 \sqsubseteq x_2 \sqsubseteq \dots$  について  $f(\bigsqcup x_i) = \bigsqcup f(x_i)$

が成り立つことをいう.

(CPO を適当に位相空間とみなした場合の関数の連続性と同値)

## 最小不動点定理

定理 CPO  $(X, \sqsubseteq)$  上の連続関数  $f : X \rightarrow X$  について,  $f(x) = x$  となる  $x \in X$  で最小のものが存在し, それは

$$\perp \sqsubseteq f(\perp) \sqsubseteq f^2(\perp) \sqsubseteq f^3(\perp) \sqsubseteq \dots$$

の上限  $\bigsqcup f^n(\perp)$  で与えられる.

証明  $f$  の連続性から

$$f(\bigsqcup f^n(\perp)) = \bigsqcup f^{n+1}(\perp) = \bigsqcup f^n(\perp)$$

また,  $f(x) = x$  とすると,  $\perp \sqsubseteq x$  より  $f^n(\perp) \sqsubseteq f^n(x) = x$  なので,  $\bigsqcup f^n(\perp) \sqsubseteq x$  である.

## 最小不動点による再帰プログラムの解釈

$[\mathbf{Z} \rightarrow \mathbf{Z}]$  上の連続関数  $F$  を

$$F(f)(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } x \times f(x-1)$$

で定めると,  $F^n(\perp) : \mathbf{Z} \rightarrow \mathbf{Z}$  は

$$F^n(\perp)(x) = \begin{cases} x \cdot (x-1) \cdot \dots \cdot 2 \cdot 1 & x < n \\ \text{未定義} & x \geq n \end{cases}$$

である. したがって,  $F$  の最小不動点  $\sqcup F^n(\perp)$  は, 階乗を計算する関数に他ならない. つまり, 再帰プログラム

$$\text{fact}(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } x \times \text{fact}(x-1)$$

の解釈を与えている.



## 不動点意味論に関する補足

CPO と連続関数の圏は，再帰プログラムのみならず，プログラミング言語の多くの側面を解釈するために十分な構造を持っている．

- データの組をひとつのデータとして扱うために必要な有限直積
- プログラムをデータとして扱うために必要な冪対象
- 再帰的に定義されたデータ構造を扱うのに必要な，ある種の（非常識なくらい強い）閉包性・完備性

冒頭で引用した Freyd の言葉” Computer Science contradicts Math” は，特にこの最後の性質を指したもの．Freyd は，この非常識な性質を簡潔な圏論の言葉で定式化し，この分野のその後の進展に大きな影響を与えた．

実は，その Freyd の条件を満たす圏は，必ず適当なりボン圏の充滿部分モノイダル圏になっている．

- I. 序論と概要
- II. 再帰プログラムの不動点意味論
- III. 再帰プログラムの幾何
- IV. 相互作用の幾何
- V. まとめと展望

## 背景（１） 不動点意味論への不満

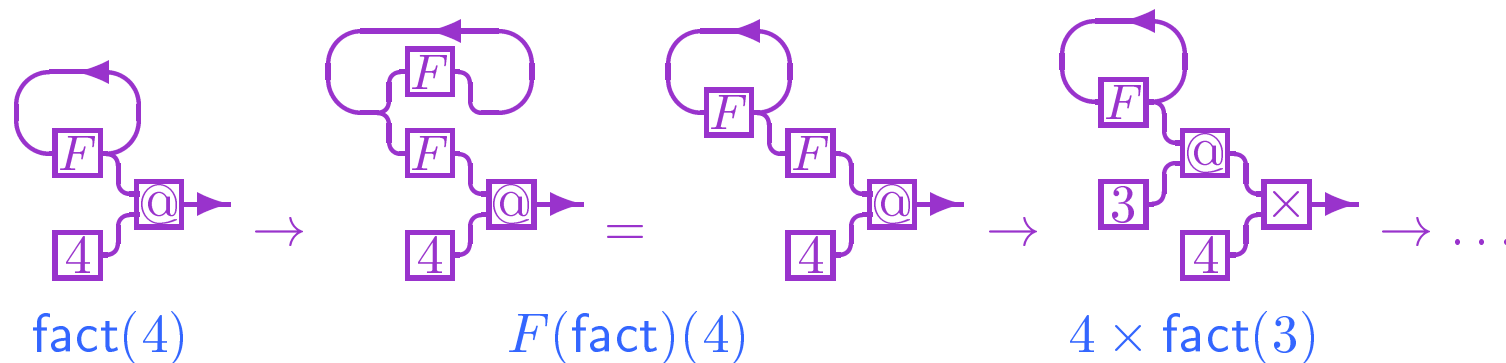
不動点意味論はシンプルでエレガントな数学理論だが、  
計算の重要な側面を過度に単純化している

- どのように再帰計算が生み出されるか説明できない
- 再帰と、プログラミング言語の他の機能との相互作用が説明できない

必要なもの：これらの問題について調べるための適度な抽象化

## 巡回構造から生み出される再帰

再帰プログラム  $\text{fact}(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } x \times \text{fact}(x-1)$  は、  
汎関数  $F(f)(x) \equiv \text{if } x = 0 \text{ then } 1 \text{ else } x \times f(x-1)$  の不動点  
( $\text{fact} = F(\text{fact})$ ) として理解することができたが、これを巡回構造から  
生み出される再帰として捉えると以下のようなになる。

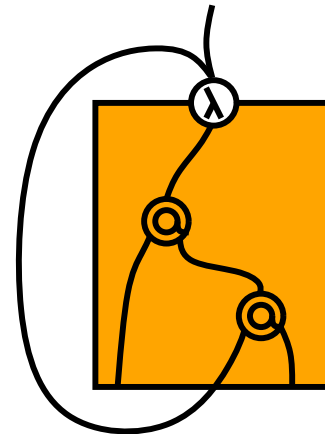
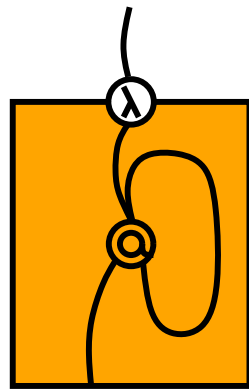


再帰呼び出し = 巡回的に共有された資源のコピー

必要なもの：再帰計算と巡回構造の関係を正確に説明できる意味論

## 例：巡回的ラムダ計算

巡回的ラムダ計算における再帰演算子のふたつの実装



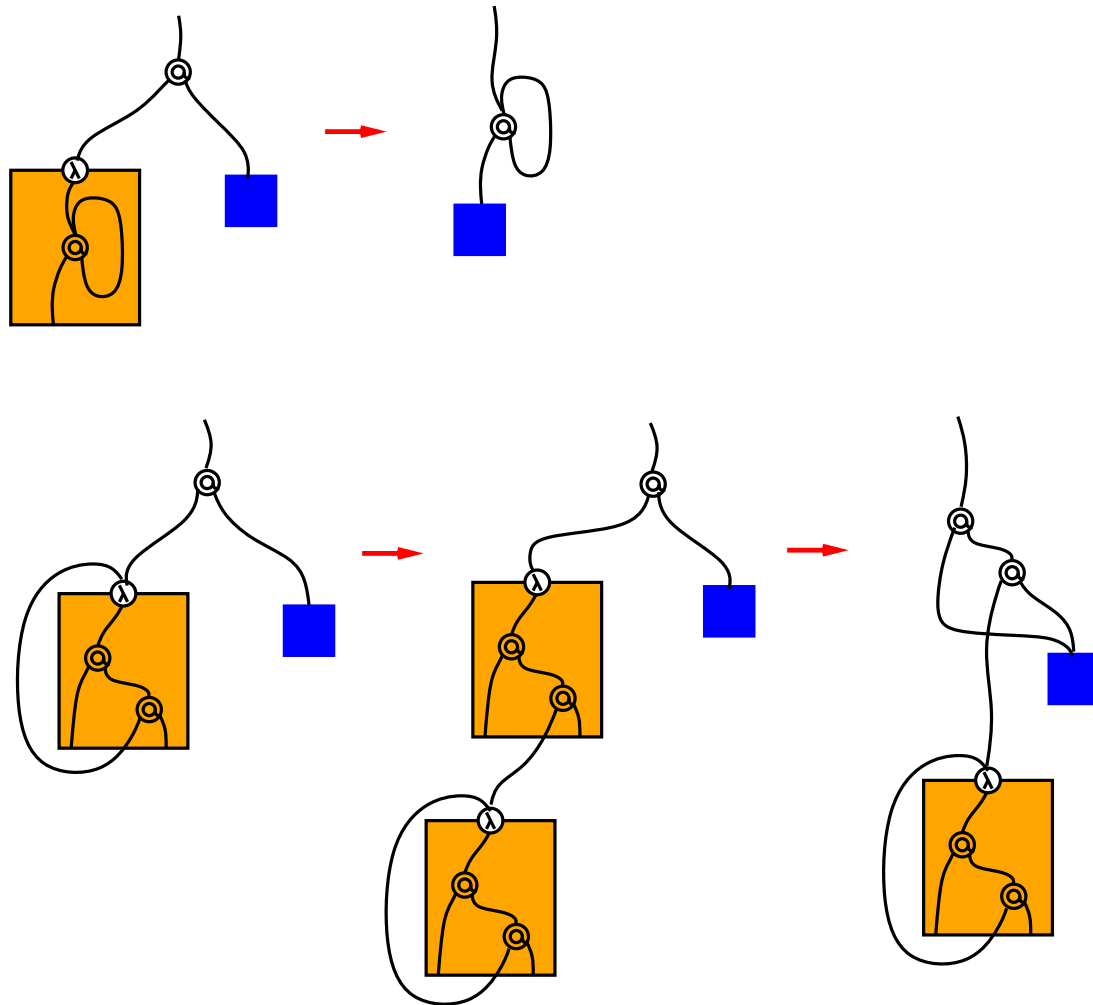
$$\lambda f.\text{letrec } x \text{ be } fx \text{ in } x \quad \stackrel{?}{=} \quad \text{letrec } F \text{ be } \lambda f.f(Ff) \text{ in } F$$

カリ-の再帰演算子

チューリングの再帰演算子

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

$$\Theta = Y(\lambda F.\lambda f.f(Ff))$$



## 背景 ( 2 ) 再帰プログラムの「幾何」?

再帰プログラムについて成り立つ「法則」

相互再帰 ( dinaturality ):

$$\text{letrec } x=g(f(x)) \text{ in } x = \text{letrec } y=f(g(y)) \text{ in } g(y)$$

対角化 ( diagonal property ):

$$\text{letrec } x=\{\text{letrec } y=h(x,y) \text{ in } y\} \text{ in } x = \text{letrec } z = h(z,z) \text{ in } z$$

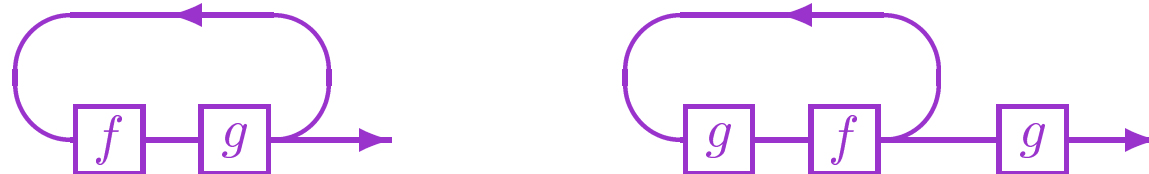
同時再帰 ( Bekič property ):

$$\text{letrec } x=f(x,y), y=g(x,y) \text{ in } x = \text{letrec } x=f(x,\{\text{letrec } y=g(x,y) \text{ in } y\}) \text{ in } x$$

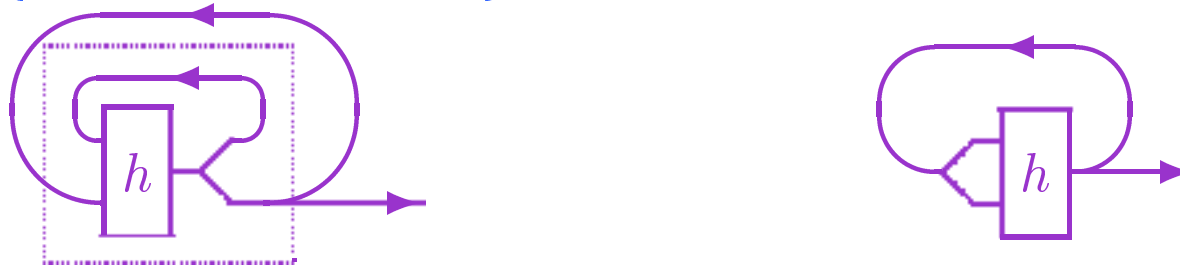
(dinaturality + diagonal  $\implies$  Bekič)

これらの法則は，幾何的な直観に対応している：

$$\text{letrec } x=g(f(x)) \text{ in } x \quad = \quad \text{letrec } y=f(g(y)) \text{ in } g(y)$$

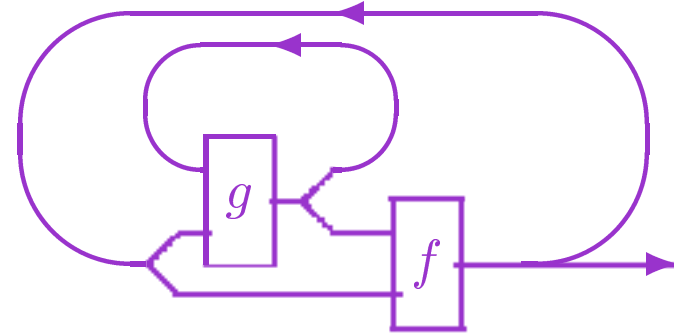
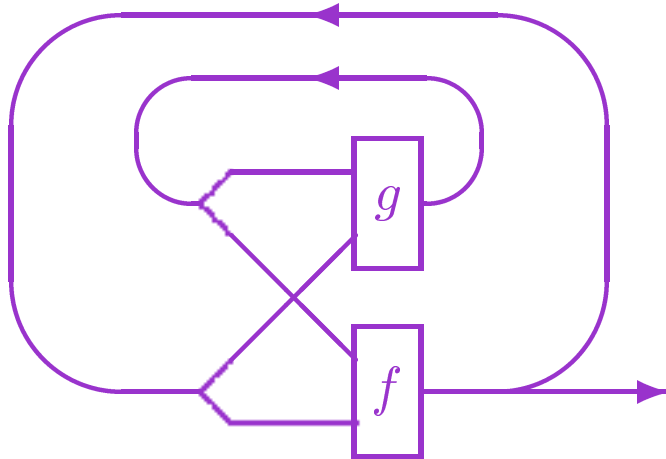


$$\text{letrec } x=\{\text{letrec } y=h(x,y) \text{ in } y\} \text{ in } x \quad = \quad \text{letrec } z = h(z,z) \text{ in } z$$



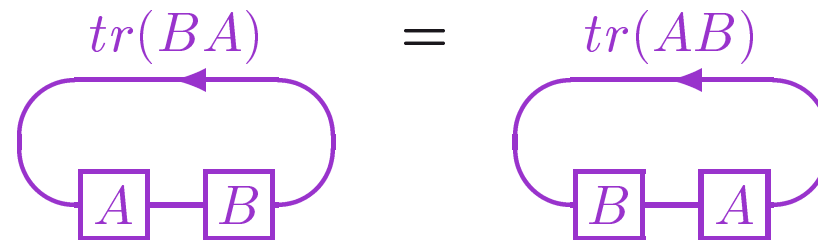


letrec  $x=f(x,y), y=g(x,y)$  in  $x$  = letrec  $x=f(x, \{\text{letrec } y=g(x,y) \text{ in } y\})$  in  $x$



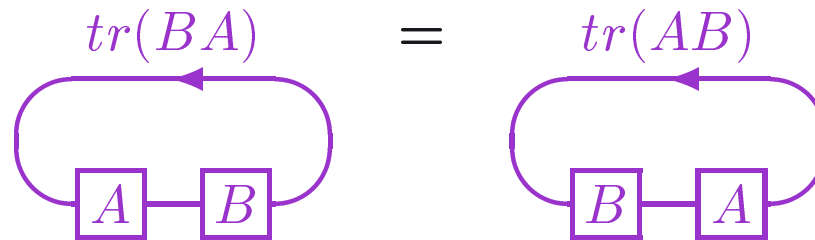
## 線形代数との比較

$m \times n$ -行列  $A$  と  $n \times m$ -行列  $B$  について  $tr(BA) = tr(AB)$  がなりたつ

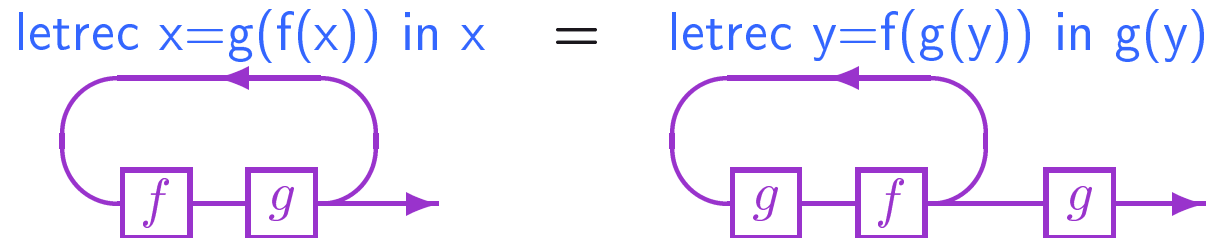


## 線形代数との比較

$m \times n$ -行列  $A$  と  $n \times m$ -行列  $B$  について  $tr(BA) = tr(AB)$  がなりたつ



これは前述の再帰プログラムの「法則」と良く似ている：

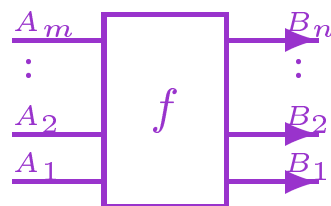


再帰を，適切に一般化されたトレースとして理解できないか？

## モノイダル圏の幾何 (Geometry of Tensor Calculus, Joyal and Street)

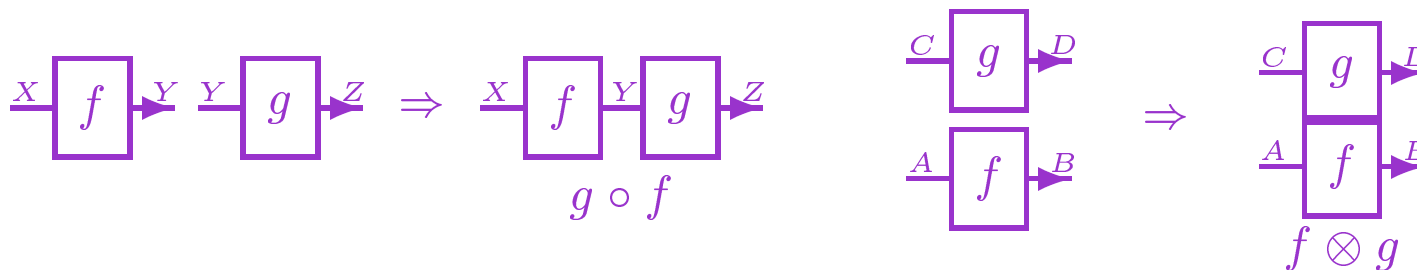
モノイダル圏 (テンソル圏): テンソル積  $\otimes$  を持つ圏

モノイダル圏の射  $f : A_1 \otimes A_2 \otimes \dots \otimes A_m \rightarrow B_1 \otimes B_2 \otimes \dots \otimes B_n$  は



と図示できる

射の合成は逐次合成で, またテンソル積は平行合成で表現できる



## トレース付きモノイダル圏 ( Joyal, Street, Veiry 1996 )

結び目のような巡回的な構造を表現できる代数構造

以下のような「トレース演算子」をもつ ( バランス付き ) モノイダル圏

$$\frac{f : A \otimes X \rightarrow B \otimes X}{\text{Tr}_{A,B}^X(f) : A \rightarrow B}$$

古典的な例：体  $K$  上の有限次元ベクトル空間と線型写像の圏

線型写像  $f : U \otimes_K W \rightarrow V \otimes_K W$  について，そのトレース

$\text{Tr}_{U,V}^W(f) : U \rightarrow V$  は以下の式で与えられる

$$(\text{Tr}_{U,V}^W(f))_{i,j} = \sum_k f_{i \otimes k, j \otimes k}$$

より一般に，任意の **リボン圏** はトレース付きモノイダル圏である

\*

## モノイダル圏

モノイダル圏 (テンソル圏) は, 圏  $\mathcal{C}$ , 関手  $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ , 対象  $I \in \mathcal{C}$  および自然同型  $a_{A,B,C} : (A \otimes B) \otimes C \xrightarrow{\sim} A \otimes (B \otimes C)$ ,  $l_A : I \otimes A \xrightarrow{\sim} A$  と  $r_A : A \otimes I \xrightarrow{\sim} A$  の組  $\mathcal{C} = (\mathcal{C}, \otimes, I, a, l, r)$  で, 以下の図式が可換になるもの:

$$\begin{array}{ccc} ((A \otimes B) \otimes C) \otimes D & \xrightarrow{a} & (A \otimes B) \otimes (C \otimes D) \\ \begin{array}{c} a \otimes D \downarrow \\ (A \otimes (B \otimes C)) \otimes D \\ a \downarrow \end{array} & & \downarrow a \\ A \otimes ((B \otimes C) \otimes D) & \xrightarrow{A \otimes a} & A \otimes (B \otimes (C \otimes D)) \end{array}$$
$$\begin{array}{ccc} (A \otimes I) \otimes B & \xrightarrow{a} & A \otimes (I \otimes B) \\ \begin{array}{c} r \otimes B \searrow \\ A \otimes B \end{array} & & \begin{array}{c} A \otimes l \swarrow \\ A \otimes B \end{array} \end{array}$$

実用上は,  $a, l, r$  を無視して  $(A \otimes B) \otimes C$  と  $A \otimes (B \otimes C)$  を同一視したり,  $I \otimes A, A \otimes I$  と  $A$  を同一視したりしても問題ない (コヒーレンス定理)

## ブレイド付きモノイダル圏

モノイダル圏における**ブレイド**は，自然同型  $c_{A,B} : A \otimes B \xrightarrow{\sim} B \otimes A$  で  $c$  と  $c^{-1}$  の両方が以下の“双線形性”を満たすもの ( $c^{-1}$  のほうは略):

$$\begin{array}{ccccc}
 (A \otimes B) \otimes C & \xrightarrow{a} & A \otimes (B \otimes C) & \xrightarrow{c} & (B \otimes C) \otimes A \\
 \downarrow c \otimes C & & & & \downarrow a \\
 (B \otimes A) \otimes C & \xrightarrow{a} & B \otimes (A \otimes C) & \xrightarrow{B \otimes c} & B \otimes (C \otimes A)
 \end{array}$$

図示すると

$$c = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} c^{-1} = \begin{array}{c} \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array}$$

$$\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array} = \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \end{array}$$

ブレイドを持つモノイダル圏をブレイド付きモノイダル圏と呼ぶ

## バランス付きモノイダル圏

ブレイド付きモノイダル圏における**バランス**（ツイスト）とは，自然な同型射  $\theta_A : A \xrightarrow{\sim} A$  で  $\theta_I = id_I$  と  $\theta_{A \otimes B} = c_{B,A} \circ (\theta_B \otimes \theta_A) \circ c_{A,B}$  を満たすもの

バランスを持つブレイド付きモノイダル圏は，**バランス付きモノイダル圏**と呼ばれる

図示すると  $\theta = \text{twist}$      $\theta^{-1} = \text{twist}$

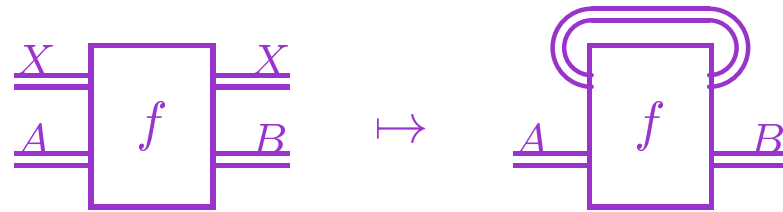




## トレース付きモノイダル圏

トレース付きモノイダル圏は、バランス付きモノイダル圏  $\mathcal{C}$  で、以下のトレースとよばれる演算をそなえたもの

$$Tr_{A,B}^X : \mathcal{C}(A \otimes X, B \otimes X) \longrightarrow \mathcal{C}(A, B)$$



トレースが満たすべき条件は以下のとおり：

Tightening:  $Tr_{A',B'}^X((k \otimes id_X) \circ f \circ (h \otimes id_X)) = k \circ Tr_{A,B}^X(f) \circ h$

Yanking:  $Tr_{X,X}^X(c_{X,X}) \circ \theta_X^{-1} = id_X = Tr_{X,X}^X(c_{X,X}^{-1}) \circ \theta_X$

Superposing:  $Tr_{C \otimes A, C \otimes B}^X(id_C \otimes f) = id_C \otimes Tr_{A,B}^X(f)$

Exchange:  $Tr_{A,B}^X(Tr_{A \otimes X, B \otimes X}^Y(f)) =$   
 $Tr_{A,B}^Y(Tr_{A \otimes Y, B \otimes Y}^X((id_B \otimes c_{Y,X}) \circ f \circ (id_A \otimes c_{Y,X}^{-1})))$

## トレースの公理 (1/2)

Tightening (Naturality)

$$\text{Tr}_{A',B'}^X((k \otimes id_X) \circ f \circ (h \otimes id_X)) = k \circ \text{Tr}_{A,B}^X(f) \circ h$$



Yanking

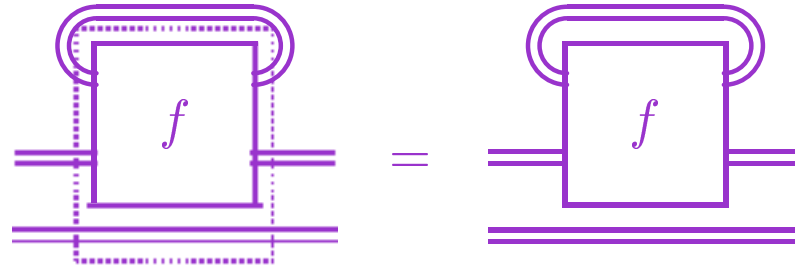
$$\text{Tr}_{X,X}^X(c_{X,X}) \circ \theta_X^{-1} = id_X = \text{Tr}_{X,X}^X(c_{X,X}^{-1}) \circ \theta_X$$



## トレースの公理 (2/2)

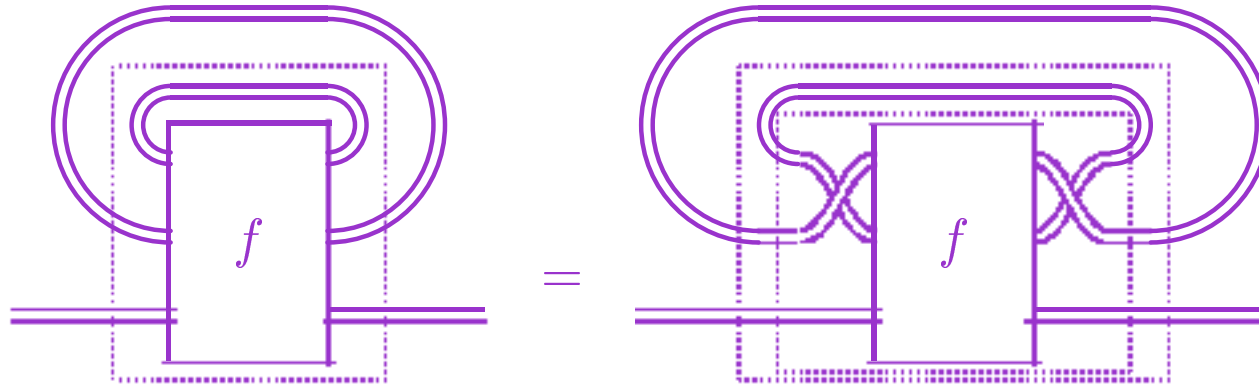
Superposing

$$\text{Tr}_{C \otimes A, C \otimes B}^X(\text{id}_C \otimes f) = \text{id}_C \otimes \text{Tr}_{A, B}^X(f)$$



Exchange

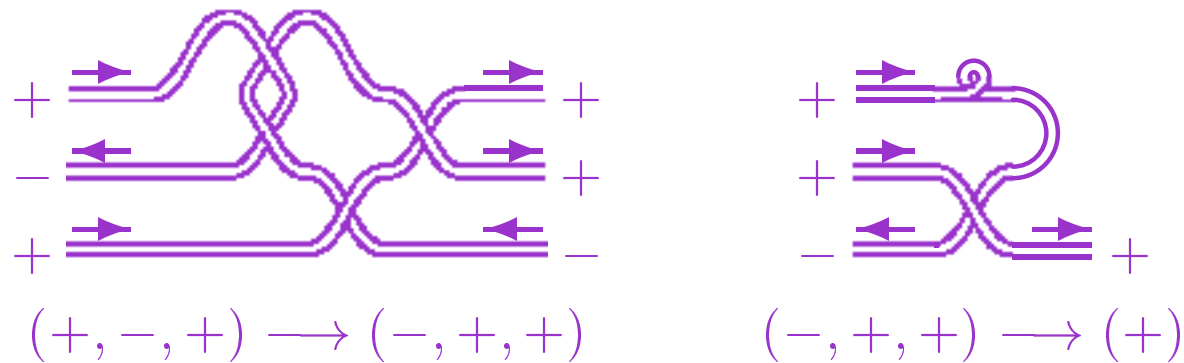
$$\begin{aligned} \text{Tr}_{A, B}^X(\text{Tr}_{A \otimes X, B \otimes X}^Y(f)) = \\ \text{Tr}_{A, B}^Y(\text{Tr}_{A \otimes Y, B \otimes Y}^X((\text{id}_B \otimes c_{Y, X}) \circ f \circ (\text{id}_A \otimes c_{Y, X}^{-1}))) \end{aligned}$$





## リボン圏の例

- 有限次元ベクトル空間と線形写像の圏  $\text{Vect}_K^{\text{fin}}$
- 量子群（リボン Hopf 代数）の表現の圏
- 枠付きタングルの圏 **Tangle** : 対象は  $+$  と  $-$  の有限列 ,  
射は適切な端点を持つ向きづけられた枠付きタングル

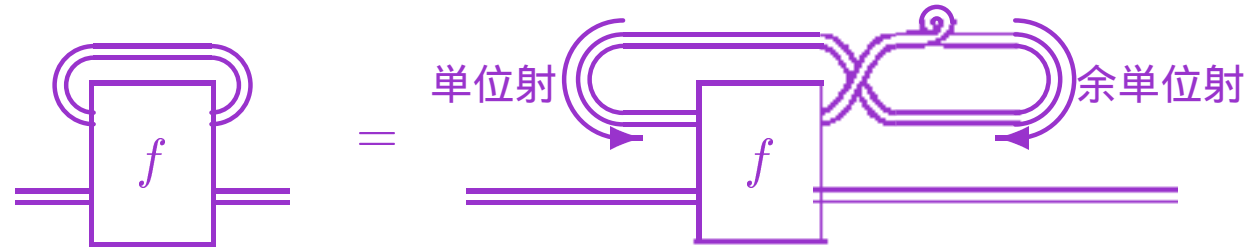


定理 (Shum). 単一の対象から自由生成されるリボン圏は枠付きタングルの圏と（リボン圏として）同値である

リボン圏（とそのひとつの対象）を与えることは，枠付きタングルの不変量を与えることに他ならない

## トレース付きモノイダル圏とリボン圏

リボン圏はトレース付きモノイダル圏でもある：



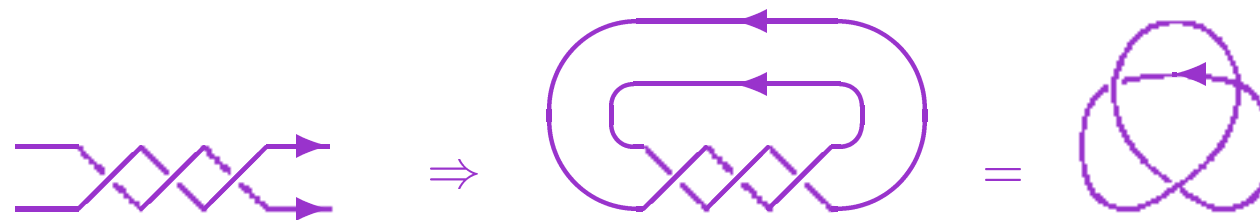
系：リボン圏の充満部分モノイダル圏はトレース付きモノイダル圏

例：Tangle の対象を  $+$  の列のみに制限した充満部分圏  $\text{Tangle}_+$

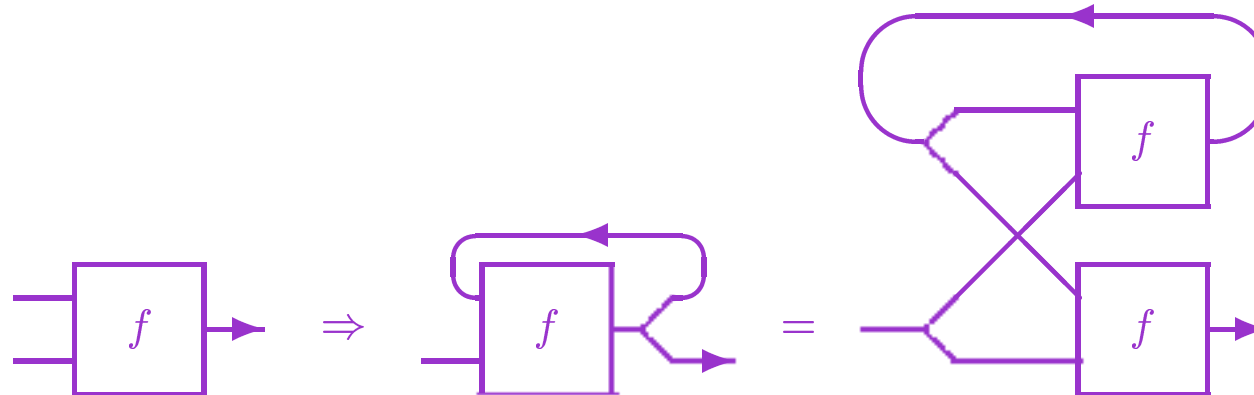
実は、任意のトレース付きモノイダル圏はあるリボン圏のモノイダル充満部分圏になっている（構造定理）。この話題にはあとで「相互作用の幾何」との関係で触れる。

結び目も再帰もトレースを用いて構成できる

トレース（ブレイド閉包）を用いた三つ葉結び目の構成：



トレース（巡回共有構造）を用いた再帰計算：



## 再帰プログラムの幾何，古典版

適当な（古典的な）条件のもとでは，トレースによる意味論は不動点意味論と完全に一致する：

定理（Hyland / 長谷川） テンソル積  $\otimes$  が直積（カルテジアン積）であるモノイダル圏において，トレースと，Bekič property を満たす不動点演算子との間に，一対一対応が存在する

例：領域理論

完備半順序集合と連続関数の圏は，直積をテンソル積とするモノイダル圏であり，しかも最小不動点から定まるトレースを持つ

連続関数  $f : A \times X \rightarrow B \times X$  について， $f(a, x) = (g(a, x), h(a, x))$  ( $g : A \times X \rightarrow B$  ,  $h : A \times X \rightarrow X$ ) とすれば， $Tr_{A,B}^X(f) : A \rightarrow B$  は

$$Tr_{A,B}^X(a) = g(a, \bigsqcup_i h_a^i(\perp)) \quad (\text{ただし } h_a(x) = h(a, x))$$



## 再帰プログラムの幾何，拡張版

反例：巡回的ラムダ計算（値しかコピーできない $\Rightarrow$  テンソル積は直積ではない）— 以下の一般化でカバーされる：

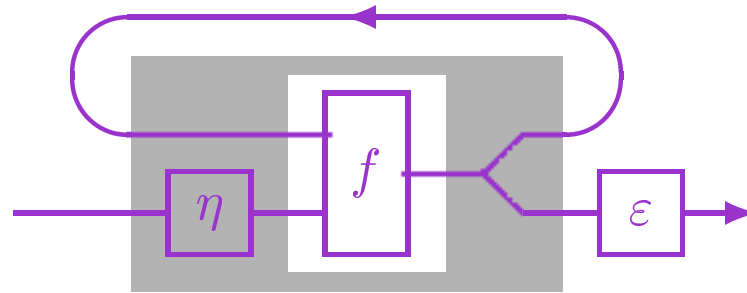
定理（長谷川）有限直積を持つ圏  $C$  と，トレースつきモノイダル圏  $D$ ，および左随伴モノイダル関手  $F : C \rightarrow D$  を考える．このとき， $D$  の射  $f : FA \otimes X \rightarrow X$  について，

$$f^\dagger = f \circ (1_{FA} \otimes f^\dagger) \circ n \circ F\Delta_A$$

をみたす  $f^\dagger : FA \rightarrow X$  が存在する

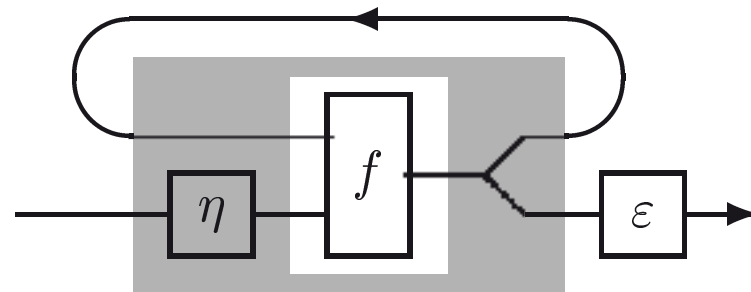
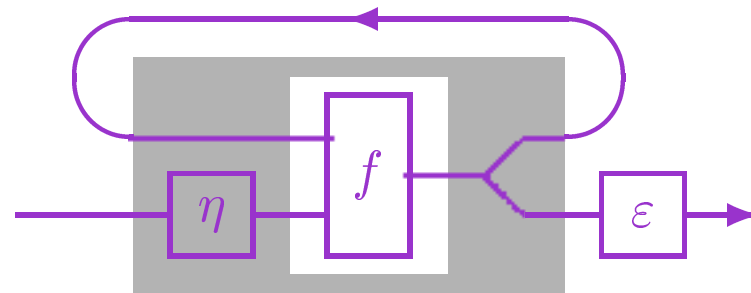
## 再帰プログラムの幾何，拡張版

テクニカルな言葉や式で書くと大変そうに思われるかも知れないが，  
図示してしまえば実はそれほど難解なことをいっているわけではない

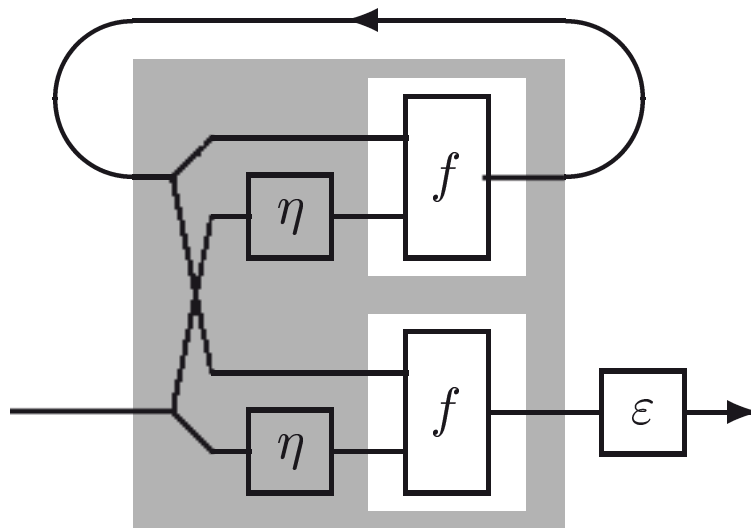
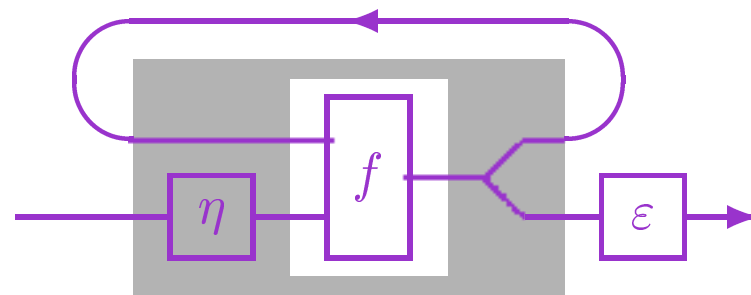


$$f^\dagger = \varepsilon \circ \text{Tr}^{FUX} (m^{F^{-1}} \circ F(\Delta \circ U f \circ m^U \circ (\eta \times id)) \circ m^F)$$

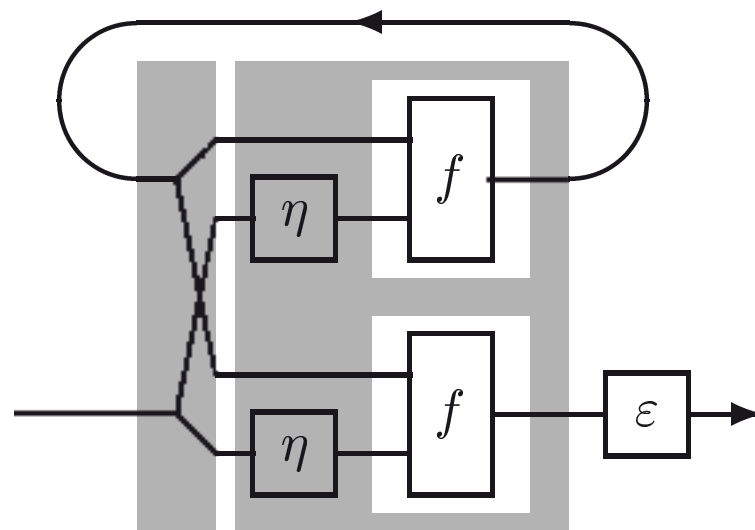
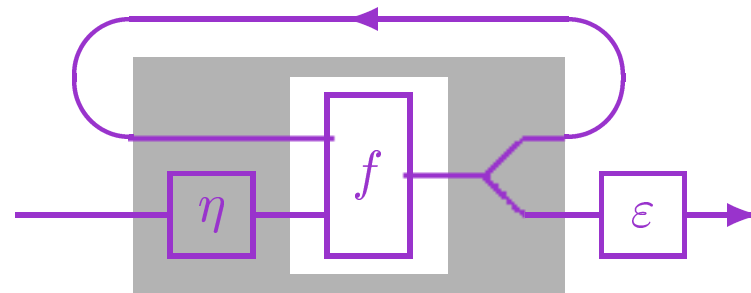
1/9



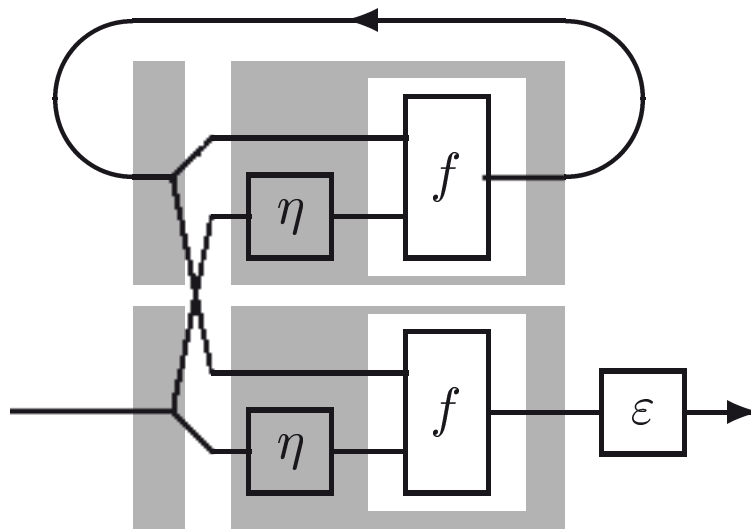
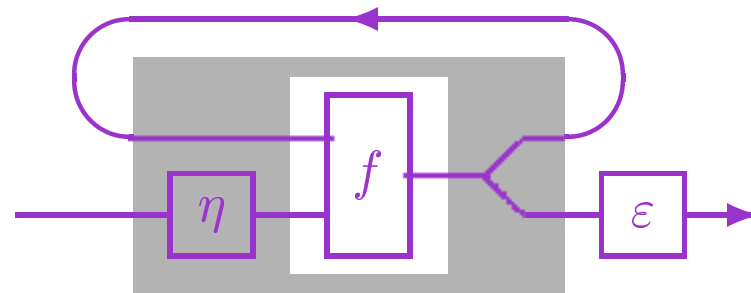
2/9



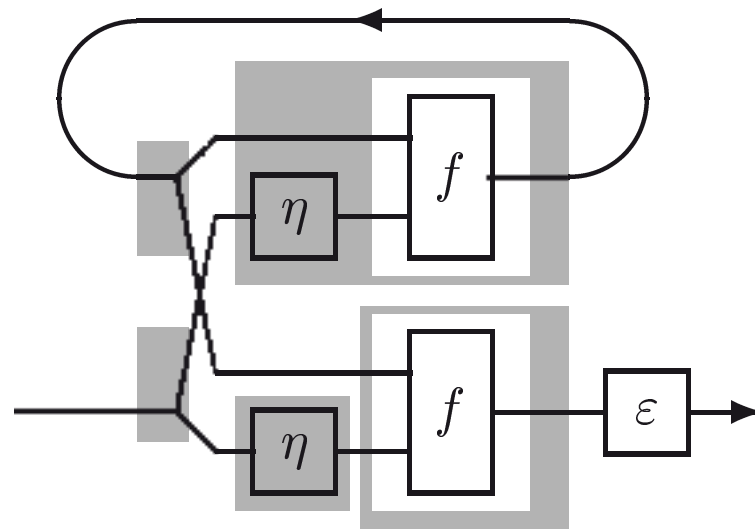
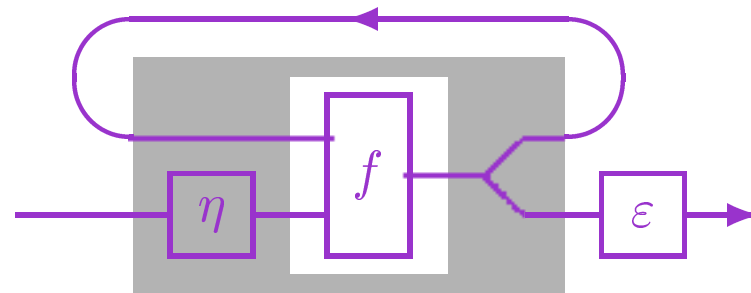
3/9



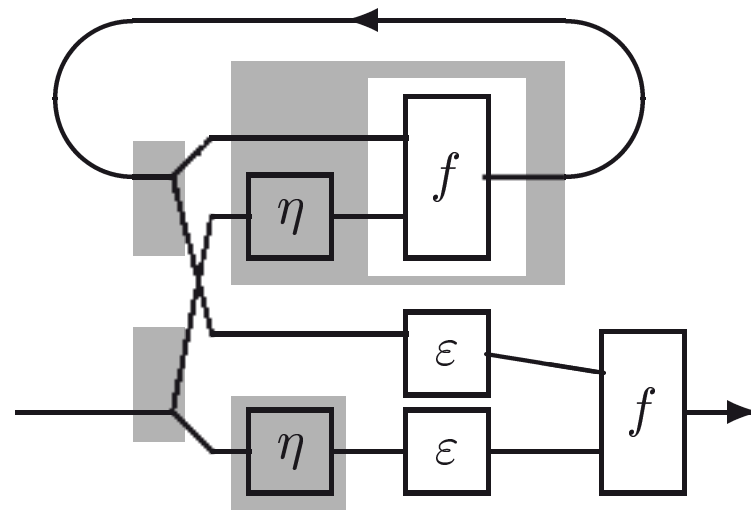
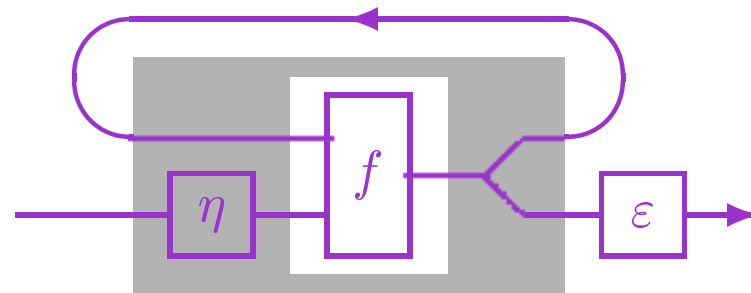
4/9



5/9

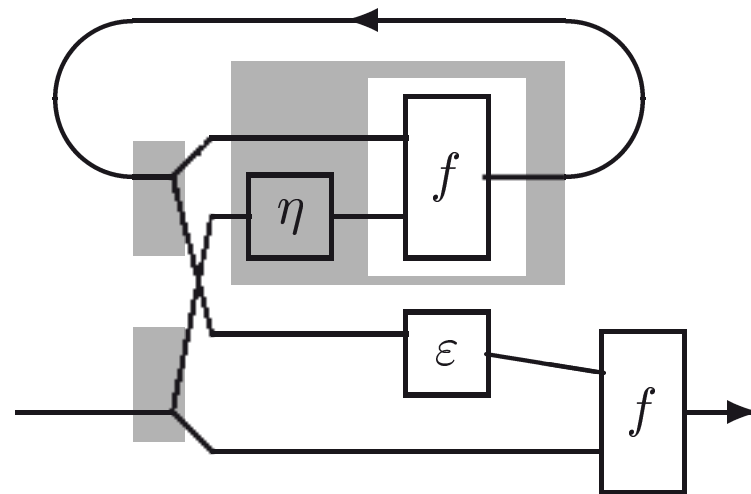
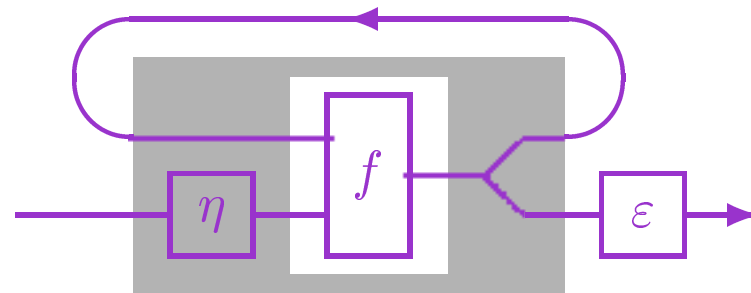


6/9

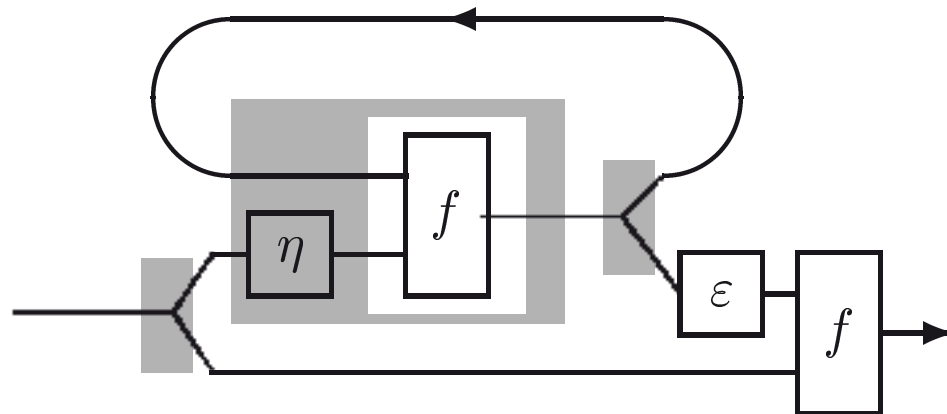
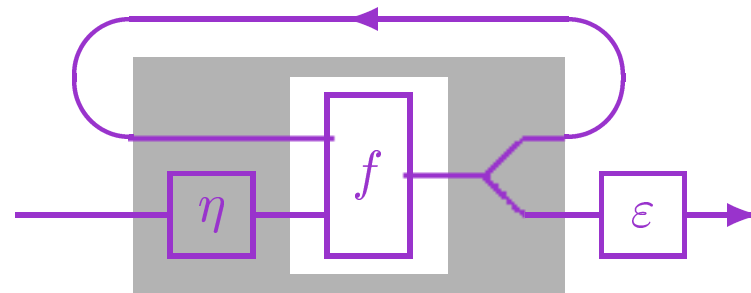




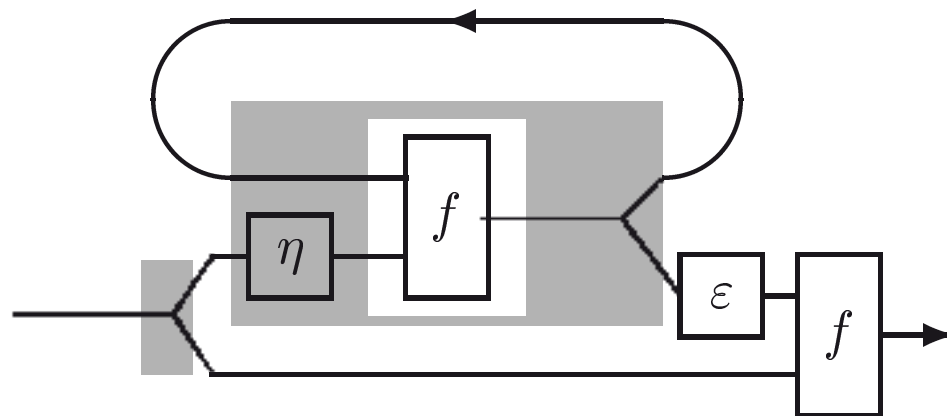
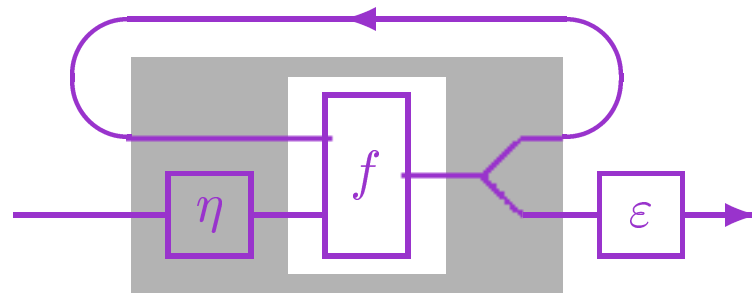
7/9



8/9

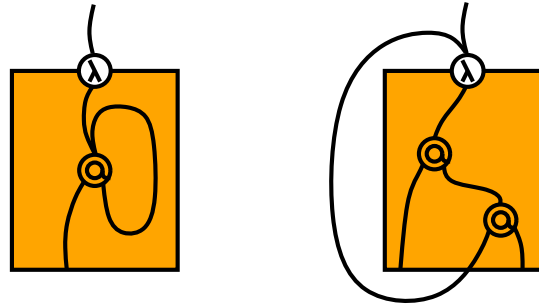


9/9



## 例：巡回的共有構造の解釈

非決定性計算のなすトレース付きモノイダル圏では



$$\text{fix1} = \lambda f. \text{letrec } x \text{ be } f x \text{ in } x$$

$$\text{fix2} = \text{letrec } F \text{ be } \lambda f. f(F f) \text{ in } F$$

$$[[\text{fix1 } M]] = \bigcup_{f \in [M]} \{x \mid (x, x) \in f\} \quad [[\text{fix2 } M]] = \bigcup_{f \in [M]} \bigcup_{f \circ A = A} A$$

これらのふたつの再帰演算子は、この「不変量」によって明確に区別される（非決定性プログラミング言語において異なる振る舞いをする）

$$\text{fix1} (\lambda x. (0 \text{ or } x + 1)) = 0 \text{ or } \perp$$

$$\text{fix2} (\lambda x. (0 \text{ or } x + 1)) = 0 \text{ or } 1 \text{ or } 2 \text{ or } \dots \text{ or } \perp$$

## 「再帰プログラムの幾何」の発展（1997～現在）

この、「巡回構造から生じる再帰計算の不変量」の理論は、従来の「再帰計算の不動点意味論」を真に拡張したものとなっている。逆に、従来の不動点意味論は、トレースによる一般的な意味論の特殊な場合として明快に理解できる。

さらに、トレース付きモノイダル圏をもとにした、新しい意味論が、多くの研究者によって発見され、調べられている

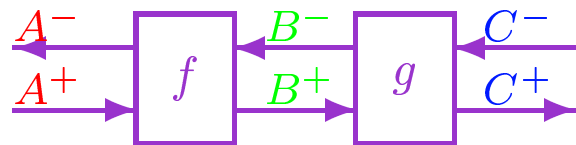
最近の展開：

トレース付きモノイダル圏における一様性原理の発見とその応用，  
公理的領域理論への応用，相互作用の幾何との関係，  
再帰と他の制御構造の関係の分析，線型な型体系における再帰の分析  
など

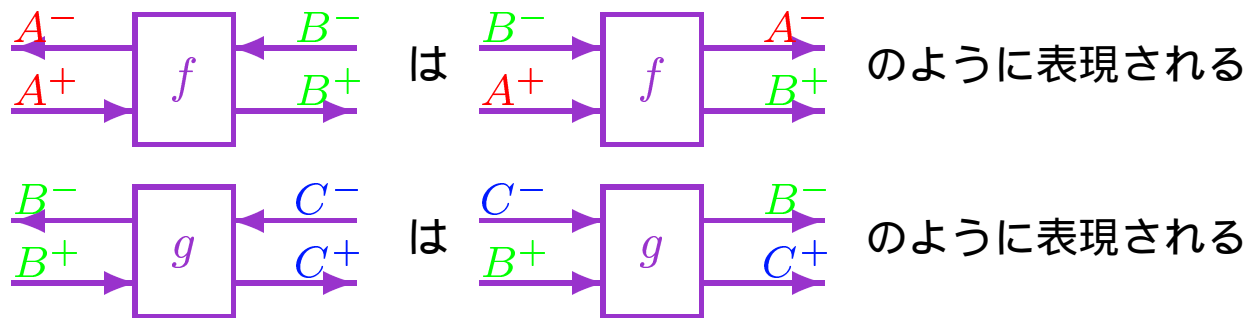
- I. 序論と概要
- II. 再帰プログラムの不動点意味論
- III. 再帰プログラムの幾何
- IV. 相互作用の幾何
- V. まとめと展望

## 双方向のコミュニケーションのモデル

相互作用的な (双方向の) コミュニケーションのモデルを与えることを考える:



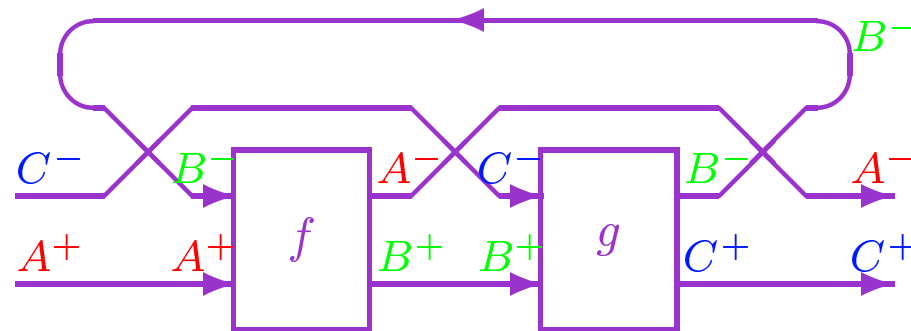
モノイダル圏では:



しかし, これらをどうやって合成したらよいのか?

Geometry of Interaction (相互作用の幾何) (Girard/Abramsky) /  
Int 構成 (Joyal, Street and Verity)

解決策: **トレース**を使えばよい!



これは, **トレース付きモノイダル圏**から**リボン圏**をつくりだす普遍的な構成を与えている

**系 (構造定理):**

**トレース付きモノイダル圏 = リボン圏の充満部分モノイダル圏**

応用: 平行計算および線型論理のさまざまな意味論の構成

最近の応用: プログラム変換における不要な高階中間コードの除去,  
**属性文法**, 非明示的計算量の理論など \*\*




## Int 構成 (Joyal, Street and Verity)

トレース付き圏  $\mathcal{C}$  から以下のようにリボン圏  $\mathbf{Int} \mathcal{C}$  を構成する

(cf. definition of integers as  $\mathbb{N} \times \mathbb{N} / \sim$  where  $(x, x') \sim (y, y')$  iff  $x + y' = y + x'$ )

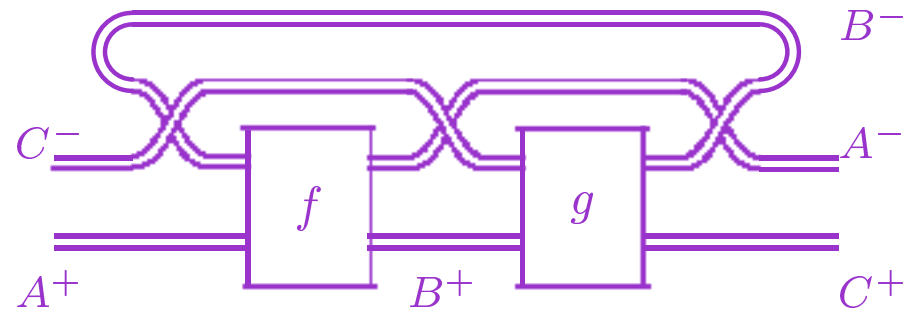
対象 :  $\mathbf{Int} \mathcal{C}$  の対象は  $\mathcal{C}$  の対象のペア

射 :  $\mathbf{Int} \mathcal{C}((A^+, A^-), (B^+, B^-)) = \mathcal{C}(A^+ \otimes B^-, B^+ \otimes A^-)$

$(A^+, A^-)$  上の恒等射は  $id_{A^+} \otimes \theta_{A^-}^{-1} \in \mathcal{C}(A^+ \otimes A^-, A^+ \otimes A^-)$  

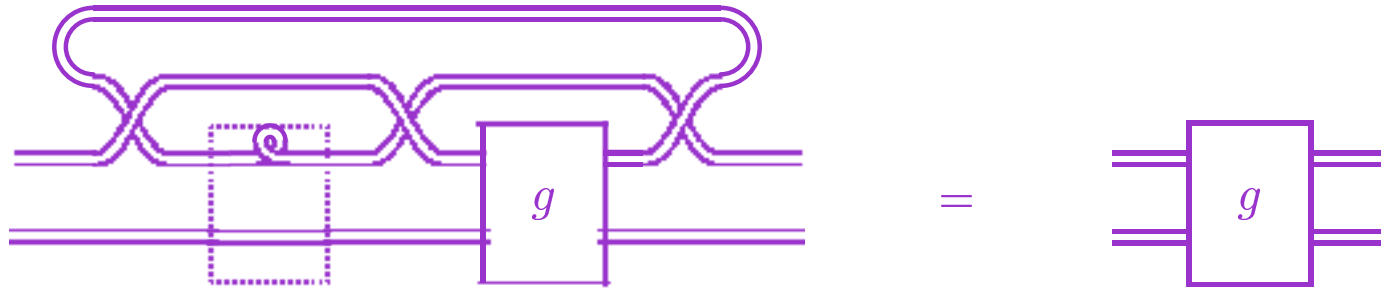
$f \in \mathbf{Int} \mathcal{C}((A^+, A^-), (B^+, B^-)) = \mathcal{C}(A^+ \otimes B^-, B^+ \otimes A^-)$  と

$g \in \mathbf{Int} \mathcal{C}((B^+, B^-), (C^+, C^-)) = \mathcal{C}(B^+ \otimes C^-, C^+ \otimes B^-)$  の合成は

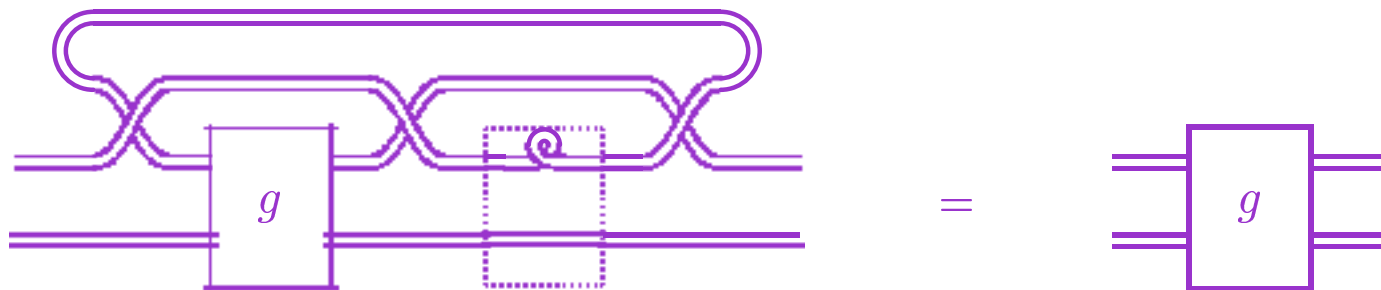


## Int 構成

$$g \circ id = g:$$

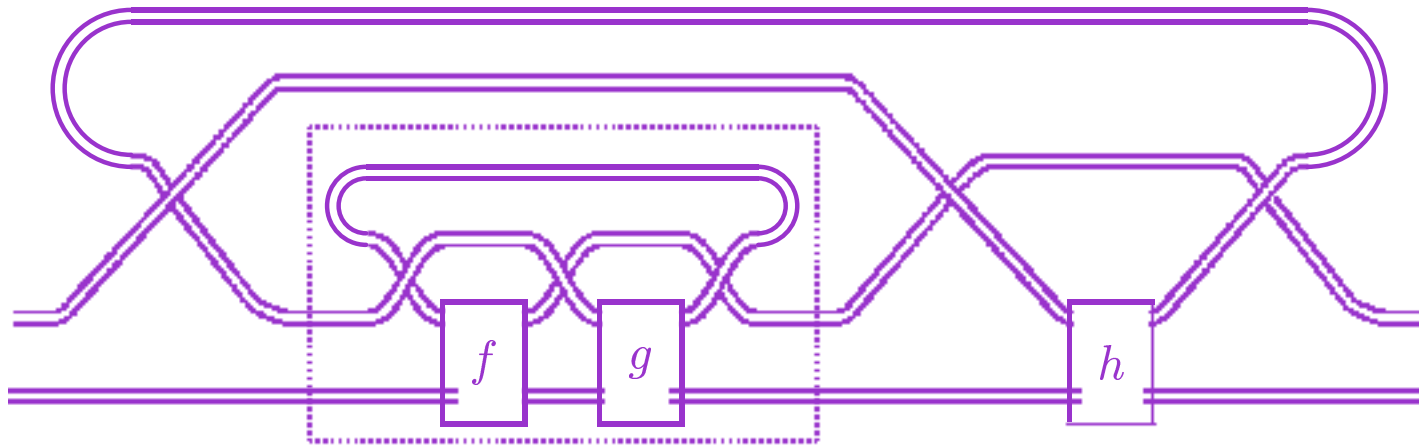


$$id \circ g = g:$$

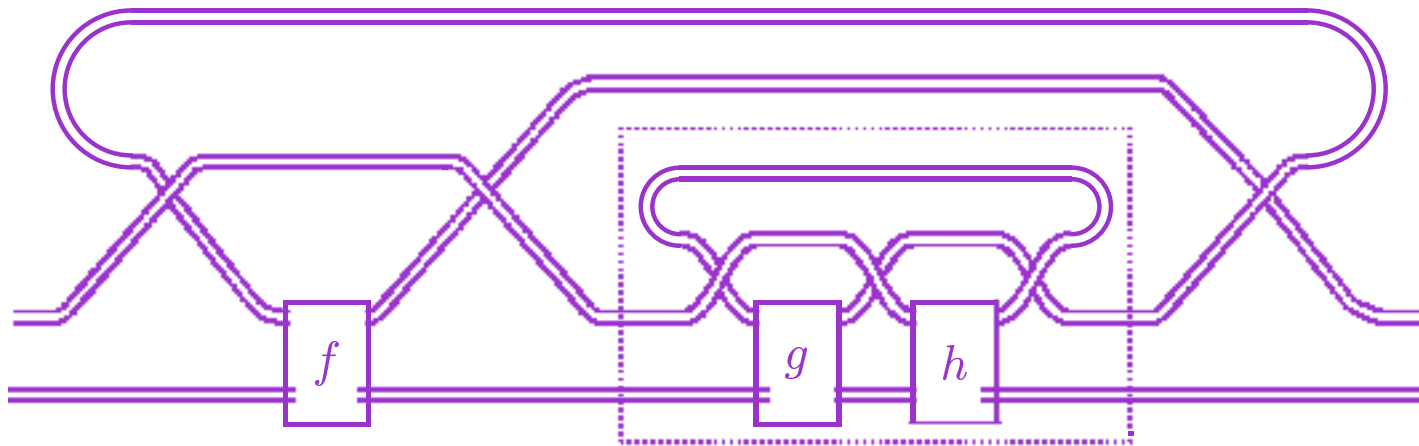


## Int 構成

$$h \circ (g \circ f) = (h \circ g) \circ f:$$



=



## Int 構成

テンソル積 :

$$(A^+, A^-) \otimes (B^+, B^-) = (A^+ \otimes B^+, B^- \otimes A^-), \quad I = (I, I)$$

$f_1 : (A_1^+, A_1^-) \rightarrow (B_1^+, B_1^-)$  と  $f_2 : (A_2^+, A_2^-) \rightarrow (B_2^+, B_2^-)$  について

$$f_1 \otimes f_2 = \text{Diagram}$$


ブレイドとバランス :

$$c = \text{Diagram} \qquad \theta = \text{Diagram}$$


## Int 構成

双対 :  $(A^+, A^-)^* = (A^-, A^+)$

単位射  $I \rightarrow (A^+, A^-) \otimes (A^+, A^-)^*$  は  $id_{A^+} \otimes \theta_{A^-}^{-1}$

余単位射  $(A^+, A^-)^* \otimes (A^+, A^-) \rightarrow I$  は  $id_{A^-} \otimes \theta_{A^+}$

定理 (Joyal, Street and Verity).

任意のトレース付きモノイダル圏  $\mathcal{C}$  について, 以上の定義により

$\text{Int } \mathcal{C}$  はリボン圏の構造を持つ.

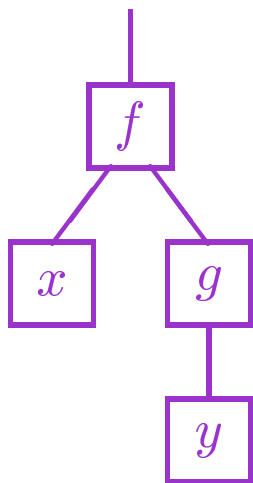
しかも、 $\mathcal{C}$  から  $\text{Int } \mathcal{C}$  への関手  $A \mapsto (A, I)$  は, トレーシング付きモノイダル圏の構造を保存し、かつ完全充満である.

トレース付きモノイダル圏  $\approx$  リボン圏の充満部分モノイダル圏

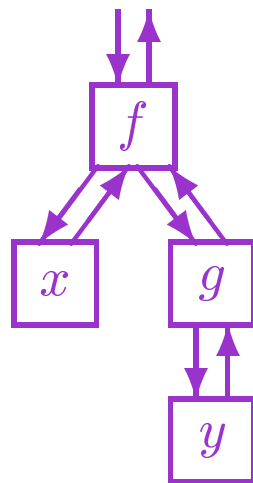
## 相互作用の幾何の具体例：属性文法 (Knuth/勝股)

コンパイラ等で文脈自由言語の意味の記述に用いられる属性文法 (attribute grammars) (Knuth 1968) も, 相互作用の幾何の一例として理解できる (勝股 2008) .

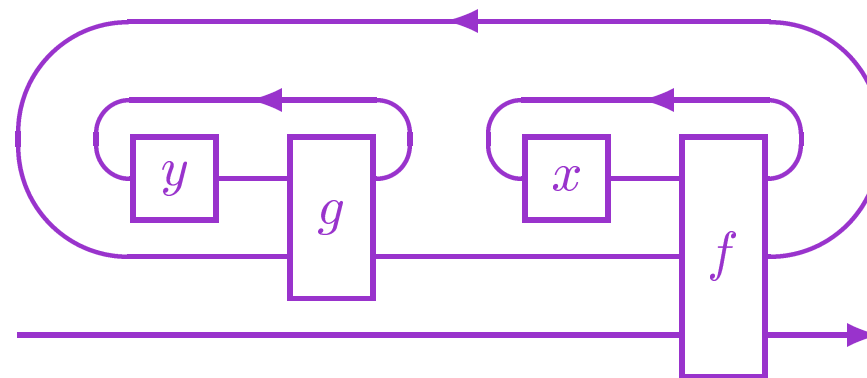
構文木



属性文法



トレースによる解釈



特に, 高階のデータ (プログラム) を属性に用いる属性文法は, 高階の相互作用の幾何の重要な例となっている .

- I. 序論と概要
- II. 再帰プログラムの不動点意味論
- III. 再帰プログラムの幾何
- IV. 相互作用の幾何
- V. **まとめと展望**

## 「再帰プログラムの幾何」のまとめ

- リボン圏から，タングルの不変量を得ることが出来る
- トレース付きモノイダル圏はリボン圏の充満部分モノイダル圏
- 再帰プログラムの意味論は，CPO の圏のような不動点を持つ圏を用いて与えられる（再帰プログラムの不動点意味論）
- テンソル積が直積である場合は，トレースと不動点は一対一に対応する
- テンソル積が直積でない場合も，適当な条件のもとでトレースから不動点を構成できる

これらの事実をあわせることにより，再帰プログラムの意味論（＝再帰プログラムのトレース付きモノイダル圏における解釈の研究）は，結び目の理論の類似とみなすことができる



## 「相互作用の幾何」のまとめ

- トレース付きモノイダル圏から，それを充満部分圏として含むようなリボン圏を構成することができる (Int 構成)
- Int 構成では，トレースを用いて「上り」と「下り」の流れを両立
- 双方向の情報流を伴う状況は，Int 構成で得られたリボン圏で自然に解釈できる

相互作用の幾何 (= Int 構成で得られたリボン圏における双方向の情報流の解釈の研究) も，結び目の理論の類似とみなすことができる

## 結び目の理論とプログラム意味論のすりあわせ？

結び目の理論では、交差  $\times$  と  $\times$  の区別がとても重要だが、プログラム意味論では、これらの違いは無視

一方、プログラム意味論では、データをコピーしたり共有したりできることが必須だが、結び目の理論では枝分かれのようなことを扱うことはあまりない（少なくとも主要な研究対象ではない）

同じトレース付きモノイダル圏・リボン圏の枠組みを用いてはいても、技術的な発展の方向は、現状ではかなり異なる

プログラム意味論と結び目の理論のすりあわせはまだまだこれからの課題 — 目指すは計算の量子不変量？ プログラム意味論の量子化？

（cf. 位相的量子計算）

## 結び目の理論とプログラム意味論のすりあわせ?(小さな試み)

プログラム意味論で用いられている圏で、量子不変量の真似を試みた。  
集合と二項関係の圏 **Rel** は、非決定性計算、平行計算や線形論理の意味論等で用いられているトレース付きモノイダル圏であり、再帰プログラムの意味論にも有用。

任意の群  $G$  は、**Rel** において(群環に相当する) Hopf 代数を与える。  
量子二重化により、**Rel** におけるリボン Hopf 代数が得られる。  
その表現の圏は、交差  $G$  集合を対象に、また交差  $G$  集合の構造を保つ二項関係を射とするリボン圏である。

初歩的ではあるが、これは、再帰プログラムの意味論とタングルの不変量の両方に必要な特徴を備えた、はじめての非自明なりボン圏である。  
ここから何か新しいことをはじめられないか?

## 結び

プログラミング言語の意味論の研究は、ありのままに  
プログラムをみない手法を身につけるための修行である

表層にとらわれないことによって浮き彫りになってくる本質に  
目を凝らすことが大切

ありがとうございました