

プログラミング言語の ゲーム意味論

——線形近似の視点から——

塚田武志（東京大学）

2015年9月8日
PPLサマースクール

はじめに

目標

PCF[Plotkin 77, Scott 99]のゲーム意味論[Hyland&Ong 2000]の理解

- いわゆる Hyland-Ong model
- ゲーム意味論の最も基本的な結果・モデル

そのために次の2つの視点を用いる

- インタラクションの理論としてのゲーム意味論
- 線形近似の理論としてのゲーム意味論

ゲーム意味論とは

プログラミング言語の意味論の分野のひとつ

- 型 : ゲームのルール
- プログラム : ゲームの戦略

90年代に始まった分野

- 線形論理のモデル [Blass 92] [Abramsky & Jagadeesan 94]
- プログラミング言語 PCF のモデル
 - 構文独立 かつ "ゴミ" がない 結果自体は90年代前半
 - HO model [Hyland & Ong 2000]
 - AJM model [Abramsky, Jagadeesan & Malacaria 2000]

ゲーム意味論 における ゲーム

二人制の対話型ゲーム

- cf. 将棋、囲碁、オセロ など
- ゲームの進行は 互いの手の列 で表現

将棋の例

▲ 7 六歩 ・ ▼ 8 四歩 ・ ▲ 2 六歩

プレイヤーは プログラム と 環境 (≡ 評価文脈)

勝者 でなく 最終局面 に興味がある

- 最終局面 ≡ 評価結果

アウトライン

対象言語

ゲーム

線形資源計算

Affine λ

{

$\mathcal{G}_{\multimap}$

簡単だが
fully definable でない

$\mathcal{G}_{\multimap}$

$\cong$

$\partial_0 \lambda_{\multimap}$

$\lambda_{\rightarrow}$

$\mathcal{G}$

$\cong$

$\partial_0 \lambda_{\rightarrow}$

PCF

$\mathcal{G}^{\text{QA}}$

$\cong$

$\partial_0 \text{PCF}$

注意

紹介する構成 (導入順序、定義) は標準とは異なる

例

- Innocent 戦略の定義は、同値な別のもの
- P-visibility や play といった概念は導入しない

目指したもの

- 天下りの的ではない 動機を理解できる概念の導入
- HO モデルの理解のための最短路
 - HO 論文ではなく

準備： λ 計算・PCF

λ計算

常に単純型付を仮定

- 陽には書かなくとも、暗黙に型が付いている
- 基底型は o ただひとつ

$$T, U, S ::= o \mid T \rightarrow U$$

発散を意味する特殊な定数 Ω を入れる

$$M, N, L ::= x \mid \lambda x.M \mid M N \mid \Omega$$

$$\overline{\vdash \Omega : o}$$

標準形

$\beta\eta$ 標準形のことを、単に標準形という

$$\check{M} ::= \lambda x_1 x_2 \dots x_n. y \check{M}_1 \dots \check{M}_k$$

$$\left(\begin{array}{l} n \geq 0 \\ k \geq 0 \\ y \check{M}_1 \dots \check{M}_k : o \end{array} \right)$$

$\beta\eta$ 同値関係

$$\frac{}{(\lambda x.M) N =_{\beta\eta} M[N/x]} \qquad \frac{x \notin \text{fv}(M)}{M =_{\beta\eta} \lambda x.M x}$$
$$\frac{M =_{\beta\eta} M'}{\lambda x.M =_{\beta\eta} \lambda x.M'} \qquad \frac{M =_{\beta\eta} M'}{M N =_{\beta\eta} M' N} \qquad \frac{N =_{\beta\eta} N'}{M N =_{\beta\eta} M N'}$$

※ 反射律、推移律、対称律は省略

事実 任意の項 M に対して、 $M =_{\beta\eta} N$ となる標準形の項 N が (α 同値を除いて) ちょうど一つ存在する

Affine λ 計算

次の条件を満たす λ 項を、**affine λ 項**と呼ぶ

任意の部分項 M と変数 x について
 x は M 中で 高々一度しか (自由に) 出現しない
各変数は高々一度しか使わない

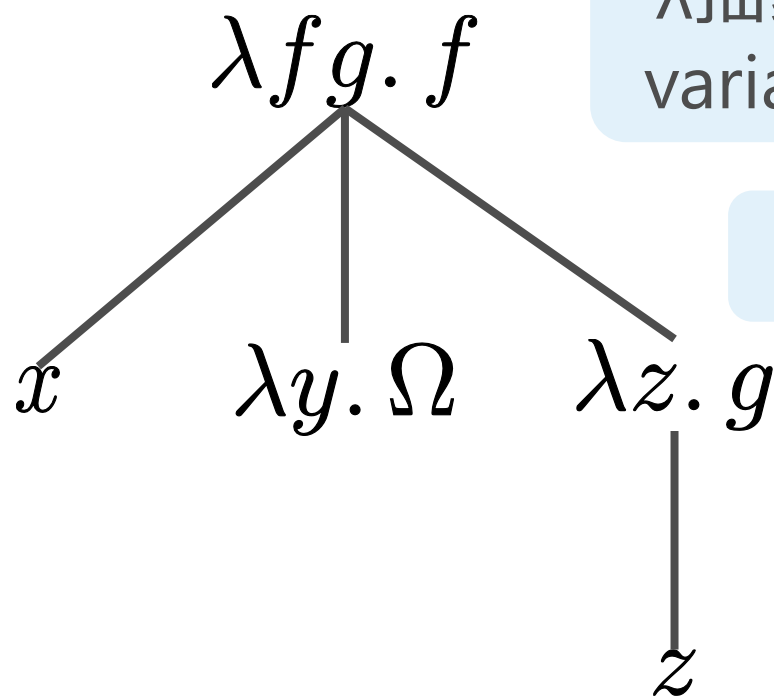
Affine λ 計算の関数型は $T \multimap U$ と書く

<u>例</u>	affine	$\lambda f.f y x$
	not affine	$\lambda f x y.f (f x)$

標準形の項の木表現 (cf. [Stirling 05] [Ong 06])

例 $\lambda f g. f x (\lambda y. \Omega) (\lambda z. g z)$

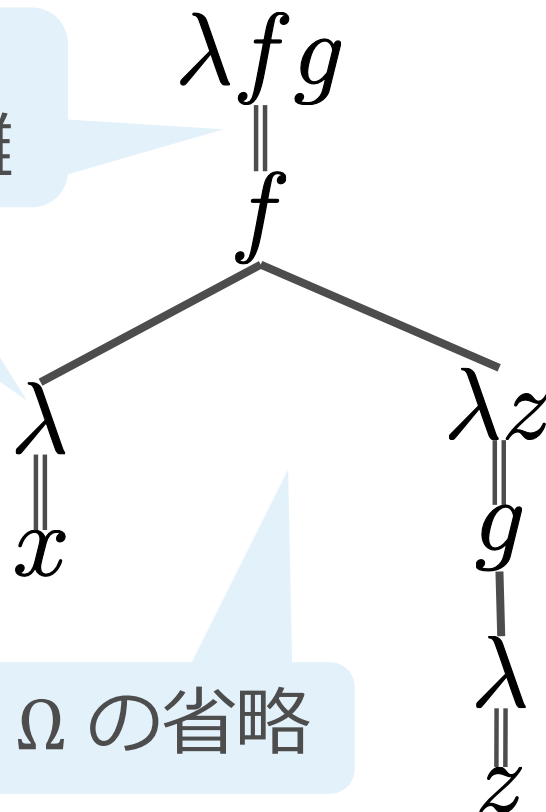
良くある表現



今回用いる表現

λ抽象と head-variable を分離

empty λ



PCF [Plotkin 77, Scott 99]

真偽値、自然数、再帰のある 単純型付きの関数型言語

- 評価戦略は 名前呼び

$$T ::= \mathbf{nat} \mid \mathbf{bool} \mid T \rightarrow T$$

$$\begin{aligned} M ::= & x \mid \lambda x.M \mid M N \mid \mathbf{Y} \\ & \mid \mathbf{if} \ M \ \mathbf{then} \ M \ \mathbf{else} \ M \mid \mathbf{t} \mid \mathbf{f} \\ & \mid \mathbf{iszero} \mid \mathbf{succ} \mid \mathbf{pred} \mid 0 \mid 1 \mid \dots \end{aligned}$$

Affine λ 計算のゲーム意味論 I

<u>対象言語</u>	<u>ゲーム</u>	<u>線形資源計算</u>
Affine λ	$\mathcal{G}_{\rightarrow}^-$	
	$\mathcal{G}_{\rightarrow}$	$\cong \partial_0 \lambda_{\rightarrow}$
$\lambda_{\rightarrow}$	$\mathcal{G}$	$\cong \partial_0 \lambda_{\rightarrow}$
PCF	$\mathcal{G}^{\text{QA}}$	$\cong \partial_0 \text{PCF}$

アウトライン

1. 単純なゲーム $\mathcal{G}_2$.
2. $\mathcal{G}_2$ による affine λ 計算の解釈
 - 項の構造についての帰納的解釈
 - 標準形の項の直接的解釈
3. 問題点: 項に対応しない戦略がある
 - 変数スコープの機構がないため

アウトライン

1. 単純なゲーム $\mathcal{G}_2$ 。

- ゲームと戦略
- ゲームの構成
- 戦略の合成

2. $\mathcal{G}_2$ による affine λ 計算の解釈

3. 問題点: 項に対応しない戦略がある

$\mathcal{G}_2$ のゲーム

定義 $\mathcal{G}_2$ の **ゲーム** とは集合の組 $A = (M_A^O, M_A^P)$ である

- M_A^O の要素を **O-move** という (相手の手)
- M_A^P の要素を **P-move** という (自分の手)

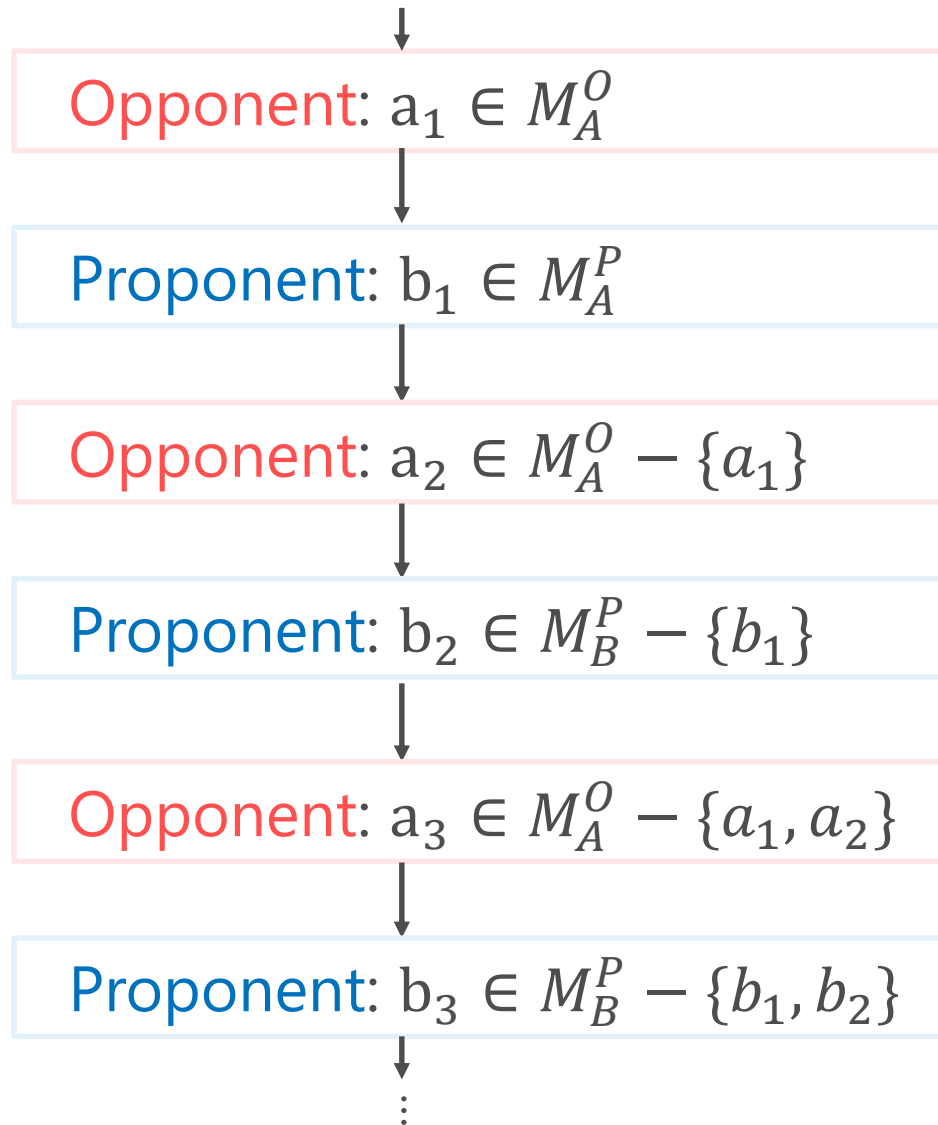
定義 ゲーム $A = (M_A^O, M_A^P)$ の **affine 交代列** とは、

列 $s: \{1, 2, \dots, n\} \rightarrow (M_A^O + M_A^P)$ であって、

- $s(1) \in M_A^O$
- $\forall i. s(i) \in M_A^O \Leftrightarrow s(i+1) \in M_A^P$ (alternating)
- $\forall i, j. i \neq j \Rightarrow s(i) \neq s(j)$ (affine)

を満たすものを言う。affine 交代列の集合を $\hat{P}_A$ と書く

ゲームのルールを図解



各 move は
高々一度しか使えない
= affineness

$\mathcal{G}_2$ の戦略

相手の手から適切な応手への関数

定義 ゲーム $A = (M_A^O, M_A^P)$ の **戦略** とは、部分関数

$$f : M_A^O \rightarrow M_A^P$$

であって、**単射**であるものを言う。

※ 単射でないとルール違反の手を打ってしまう。

$f(a_1) = f(a_2) = b$ ($a_1 \neq a_2$) とすると、

$$a_1 \cdot b \cdot a_2 \cdot b$$

と進んだ時にルール違反 ($a_1 \cdot b \cdot a_2 \cdot b \notin \hat{P}_A$)

補足：より一般的な戦略

先ほどの戦略は **history-free 戦略** [Abramsky+ 94] と呼ばれ、大雑把に言えば、参照のない計算に対応する。

次の定義は、より一般的な戦略の概念を与える。

定義 ゲーム A の **history-sensitive 戦略** とは、

$$f : \hat{P}_A^{odd} \multimap M_A^P$$

であって、以下を満たすものと言う

$$\forall s \in \text{dom}(f). s \cdot f(s) \in \hat{P}_A$$

ただし $\hat{P}_A^{odd} := \{ s \in \hat{P} \mid \text{length}(s) \text{ is odd} \}$

ゲームと戦略の例

例 1 (ブール値)

$$B = (\{*\}, \{\mathbf{t}, \mathbf{f}\})$$

とすると、 B の戦略は

$$f : \{*\} \rightarrow \{\mathbf{t}, \mathbf{f}\}$$

という単射部分関数。つまり、次の三つ。

ボトム

$$f(*) \uparrow$$

未定義

真

$$f(*) = \mathbf{t}$$

偽

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

ゲームと戦略の例

例 2 (自然数)

$$N = (\{*\}, \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \dots\})$$

とすると、 N の戦略は

$$f : \{*\} \rightarrow \{\mathbf{0}, \mathbf{1}, \mathbf{2}, \mathbf{3}, \dots\}$$

という単射部分関数。つまり、次のいずれか。

ボトム

$$f(*) \uparrow$$

自然数 n

$$f(*) = \mathbf{n}$$

アウトライン

1. 単純なゲーム $\mathcal{G}$ 。

- ゲームと戦略
- ゲームの構成
- 戦略の合成

2. $\mathcal{G}$ による affine λ 計算の解釈

3. 問題点: 項に対応しない戦略がある

ゲームの構成

与えられたゲーム A と B から、新たなゲームを構成

1. 組ゲーム $A \otimes B$

- A と B の後手番を同時にプレイ
- affine λ 計算の型環境の解釈に使う

2. 関数ゲーム $A \multimap B$

- A の先手番と B の後手番を同時にプレイ
- affine λ 計算の関数型の解釈に使う

組ゲーム

A と B の後手番を同時にプレイ

定義 ゲーム $A = (M_A^O, M_A^P)$ と $B = (M_B^O, M_B^P)$ について

$$(A \otimes B) := (\underbrace{M_A^O + M_B^O}_{M_{A \otimes B}^O}, \underbrace{M_A^P + M_B^P}_{M_{A \otimes B}^P})$$

と定義する。これを組ゲームと呼ぶ。

関数ゲーム $A \multimap B$

A の先手番と B の後手番を同時にプレイ

定義 ゲーム $A = (M_A^O, M_A^P)$ と $B = (M_B^O, M_B^P)$ について

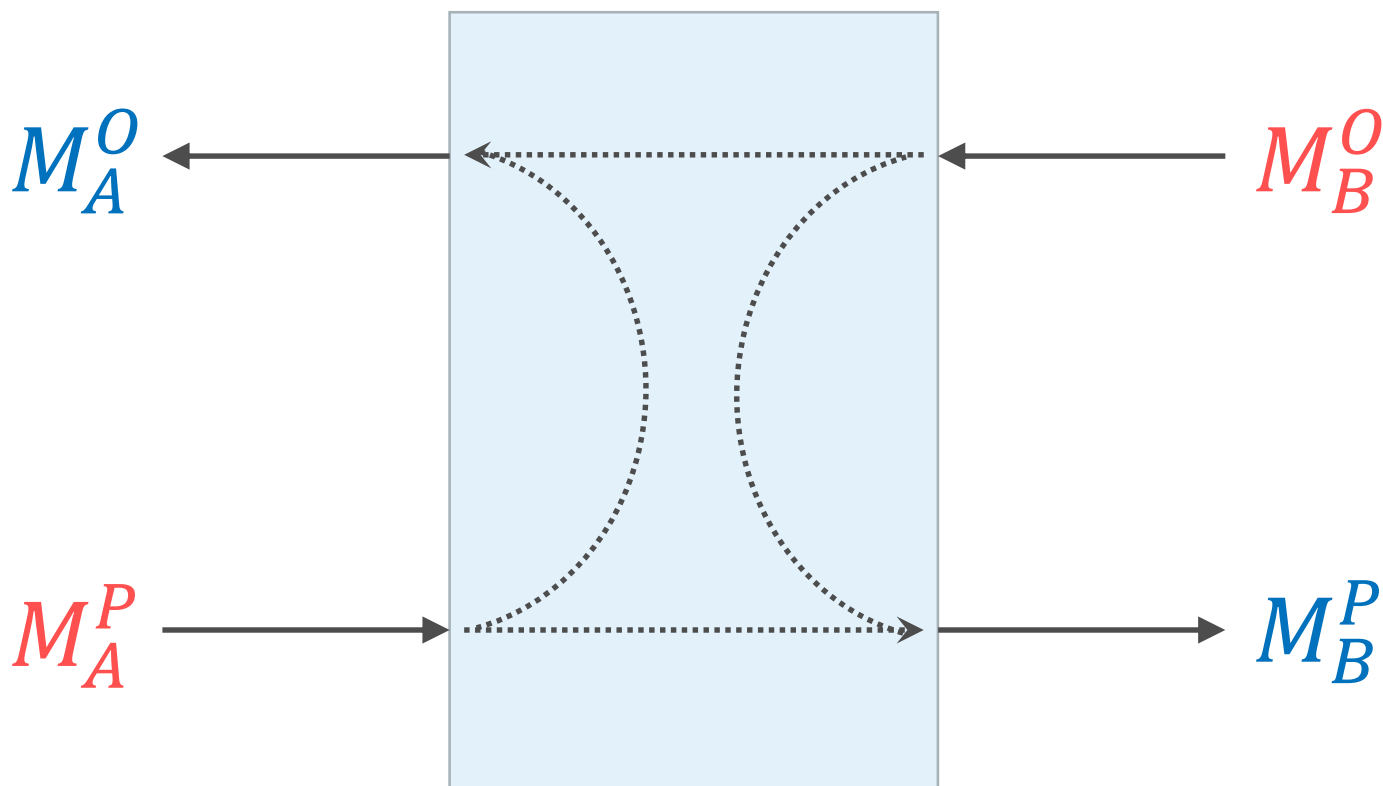
$$(A \multimap B) := (\underbrace{M_A^P + M_B^O}_{M_{A \multimap B}^O}, \underbrace{M_A^O + M_B^P}_{M_{A \multimap B}^P})$$

と定義する。これを関数ゲームと呼ぶ。

A の O と P が反転していることに注意。

関数ゲームの図示

$$A \multimap B$$



関数ゲームの例

例 1 (単射部分関数)

A と B を Opponent 部分が空なゲーム、つまり、

$$A = (\emptyset, M_A^P) \quad B = (\emptyset, M_B^P)$$

とする。このとき、

$$A \multimap B = (M_A^P, M_B^P)$$

であって、その上の戦略は次の単射部分関数である

$$f : M_A^P \rightarrowtail M_B^P$$

関数ゲームの例

例 2

ゲーム I を次のように定義する。

$$I := (\emptyset, \emptyset)$$

このとき、任意のゲーム A について、

$$I \multimap A = (\emptyset + M_A^O, \emptyset + M_A^P) \cong (M_A^O, M_A^P)$$

であり、 A と $I \multimap A$ は同じゲームである。

アウトライン

1. 単純なゲーム $\mathcal{G}_2$ 。

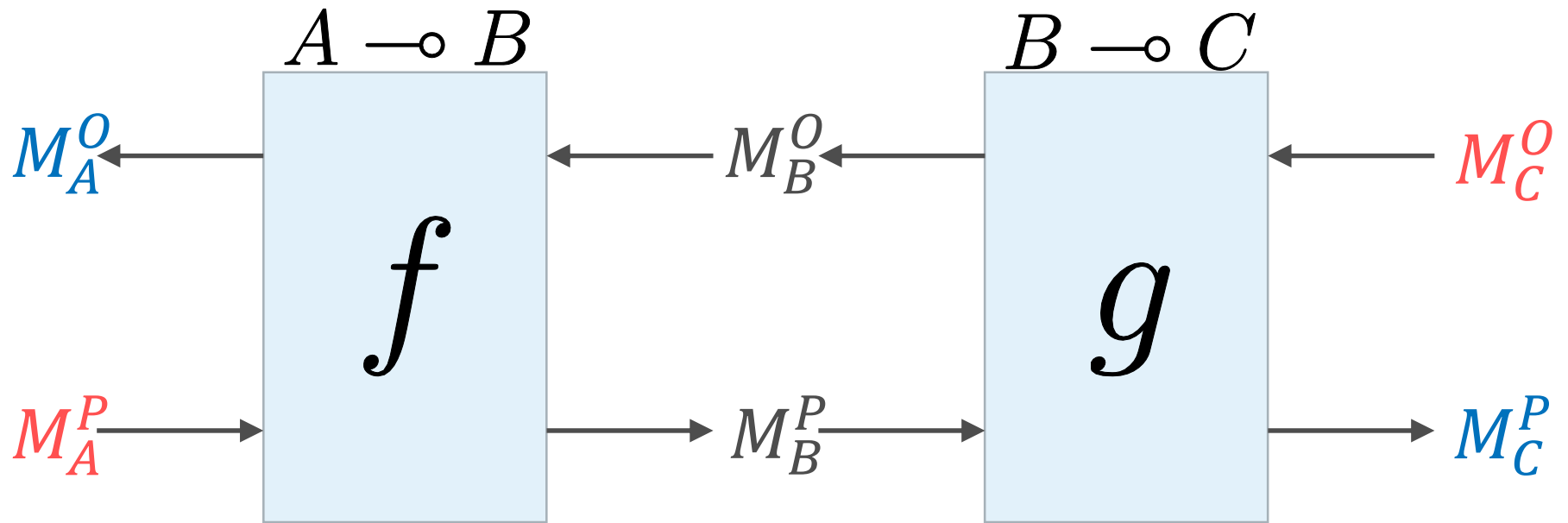
- ゲームと戦略
- ゲームの構成
- 戦略の合成

2. $\mathcal{G}_2$ による affine λ 計算の解釈

3. 問題点: 項に対応しない戦略がある

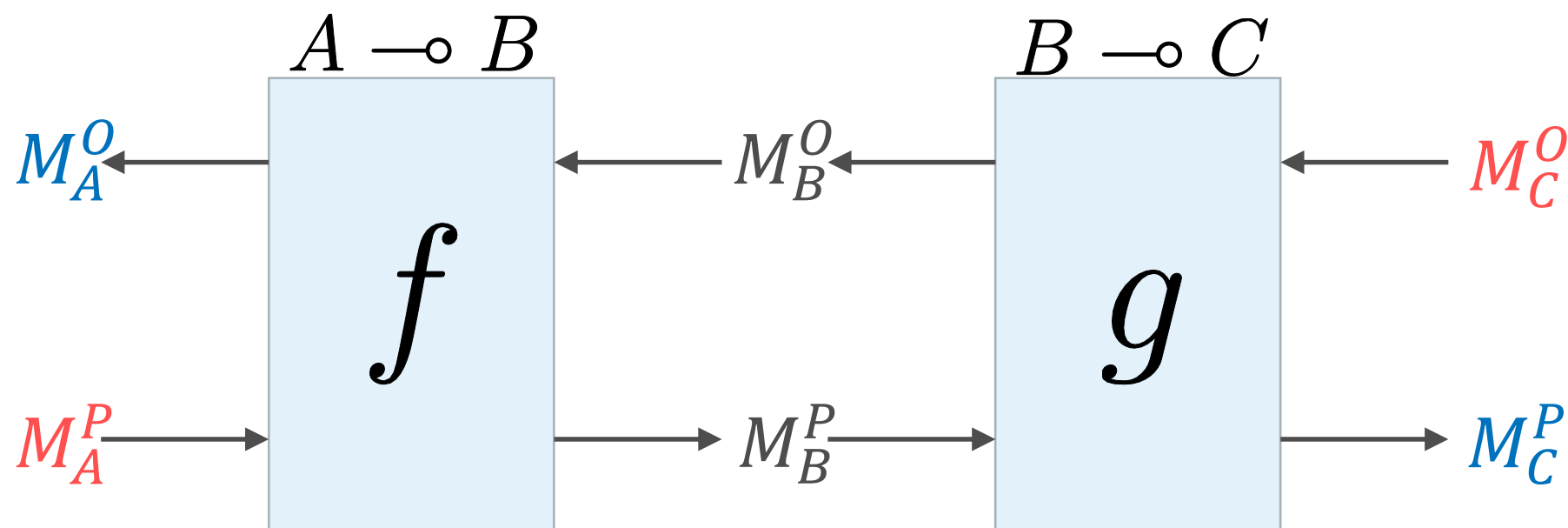
戦略の合成

$A \multimap B$ と $B \multimap C$ の戦略を合成し、 $A \multimap C$ の戦略を作る



戦略の合成

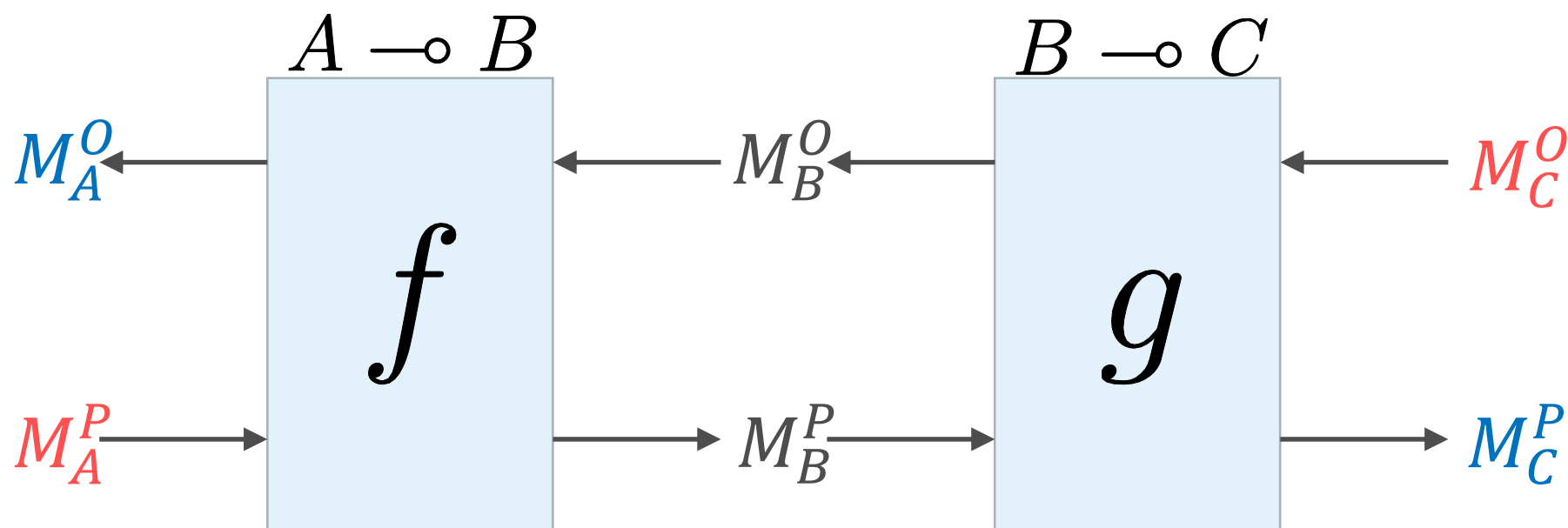
$A \multimap B$ と $B \multimap C$ の戦略を合成し、 $A \multimap C$ の戦略を作る



$$M_A^P \ni a \xrightarrow{f} a' \in M_A^O \quad \text{ならば} \quad (g \circ f)(a) = a'$$

戦略の合成

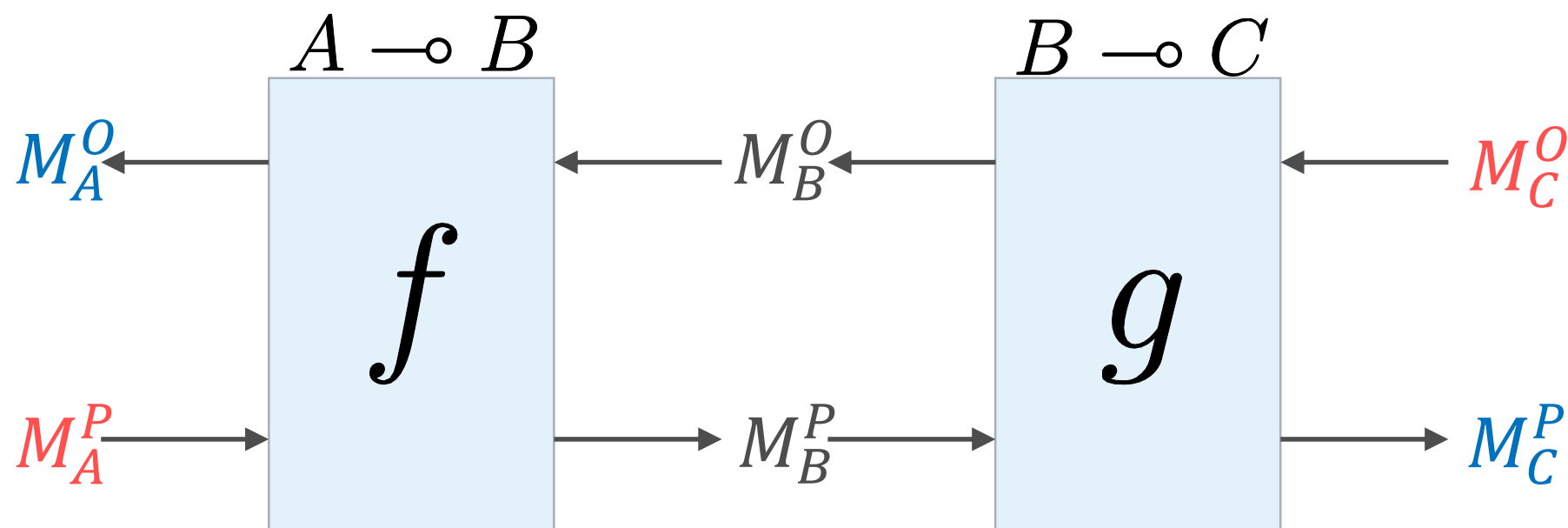
$A \multimap B$ と $B \multimap C$ の戦略を合成し、 $A \multimap C$ の戦略を作る



$$M_A^P \ni a \xrightarrow{f} b \xrightarrow{g} c \in M_C^P \quad \text{ならば } (g \circ f)(a) = c$$

戦略の合成

$A \multimap B$ と $B \multimap C$ の戦略を合成し、 $A \multimap C$ の戦略を作る

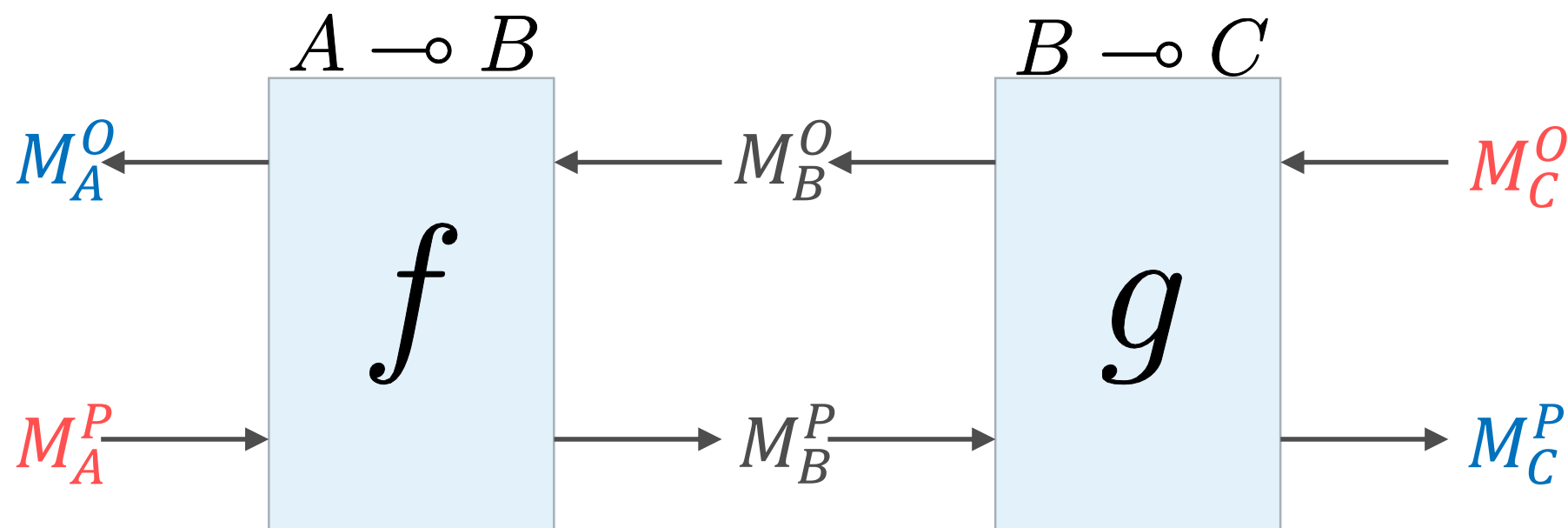


$$M_A^P \ni a \xrightarrow{f} b \xrightarrow{g} b' \xrightarrow{f} a' \in M_A^O$$

ならば $(g \circ f)(a) = a'$

戦略の合成

$A \multimap B$ と $B \multimap C$ の戦略を合成し、 $A \multimap C$ の戦略を作る

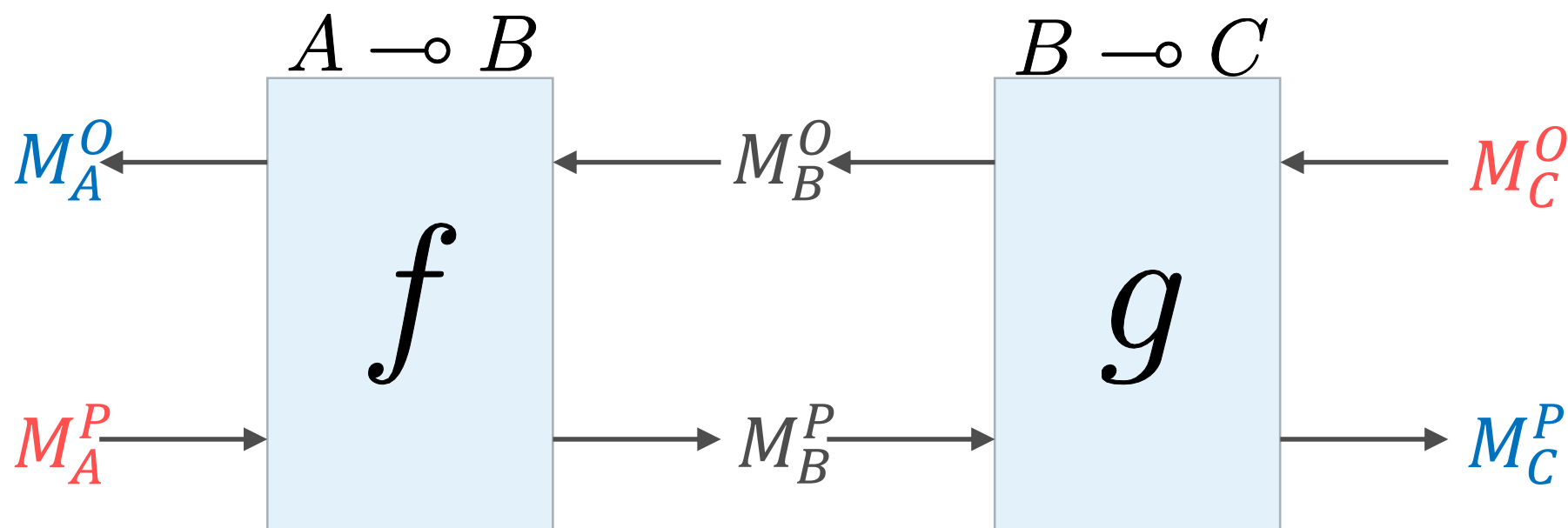


$$M_A^P \ni a \xrightarrow{f} b \xrightarrow{g} b' \xrightarrow{f} b'' \xrightarrow{g} c \in M_C^P$$

ならば $(g \circ f)(a) = c$

戦略の合成

$A \multimap B$ と $B \multimap C$ の戦略を合成し、 $A \multimap C$ の戦略を作る

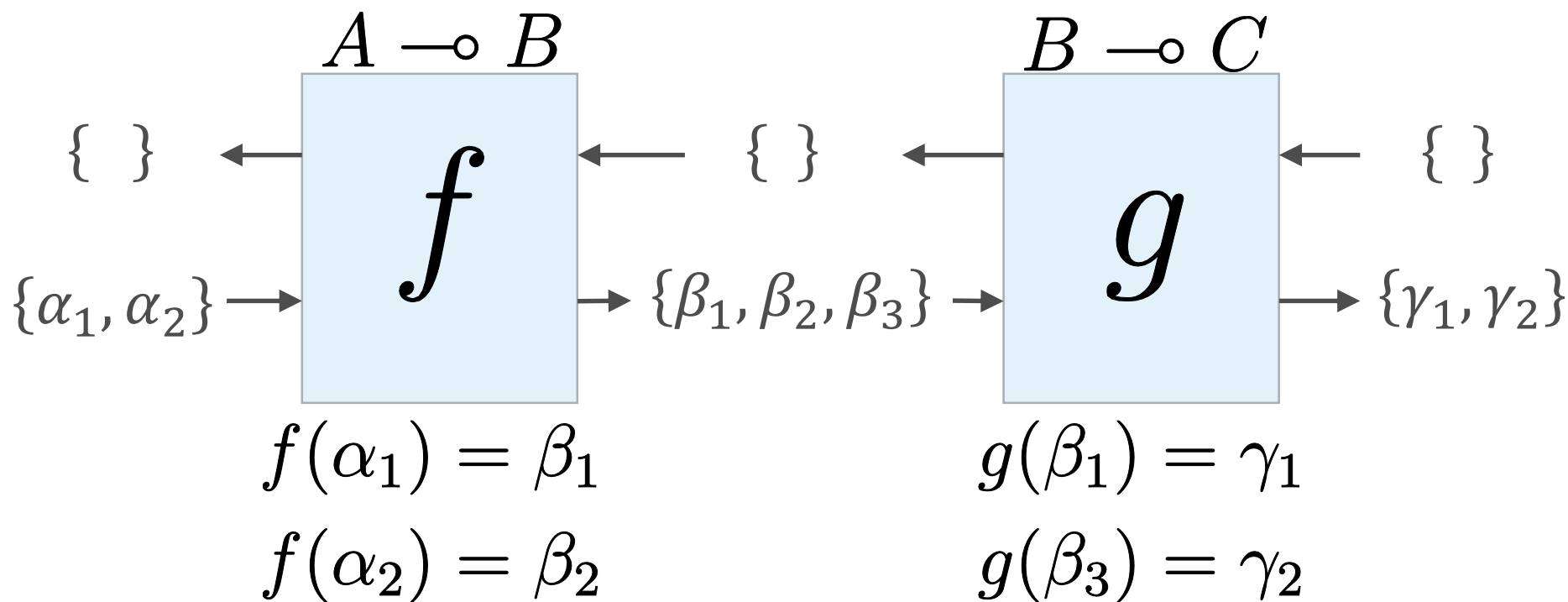


$$M_A^P \ni a \xrightarrow{f} b \xrightarrow{g} b' \xrightarrow{f} b'' \xrightarrow{g} b''' \xrightarrow{f} a' \in M_A^O$$

$$\text{ならば } (g \circ f)(a) = a'$$

合成の例

$$A = (\emptyset, \{\alpha_1, \alpha_2\}) \quad B = (\emptyset, \{\beta_1, \beta_2, \beta_3\}) \quad C = (\emptyset, \{\gamma_1, \gamma_2\})$$

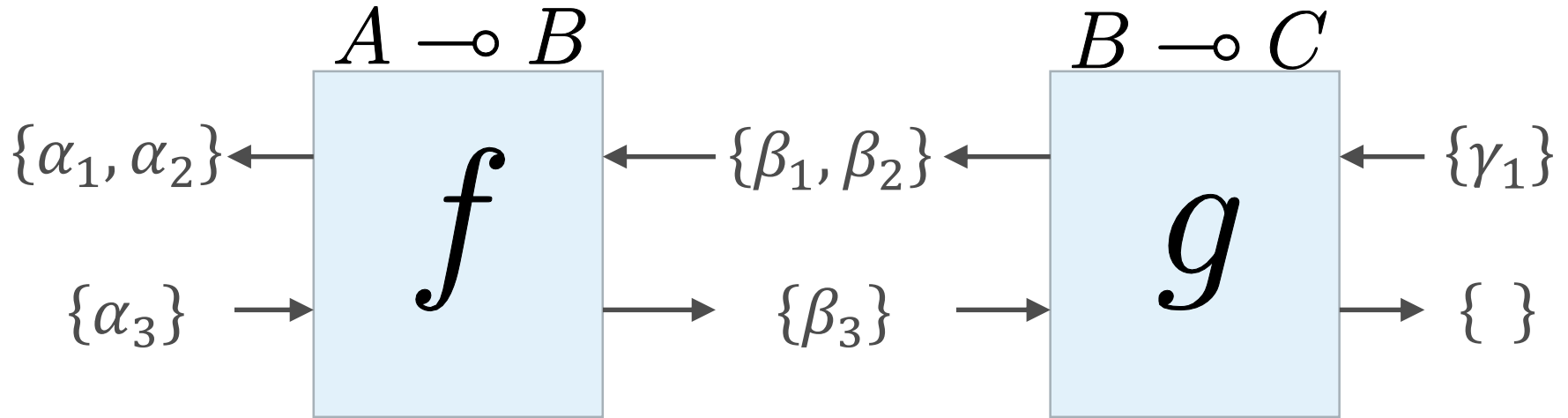


$$(g \circ f)(\alpha_1) = \gamma_1$$
$$(g \circ f)(\alpha_2) \uparrow$$

部分関数の
合成と同じ

合成の例

$$A = (\{\alpha_1, \alpha_2\}, \{\alpha_3\}) \quad B = (\{\beta_1, \beta_2\}, \{\beta_3\}) \quad C = (\{\gamma_1\}, \emptyset)$$



$$f(\alpha_3) = \alpha_2$$

$$f(\beta_1) = \beta_3$$

$$f(\beta_2) = \alpha_1$$

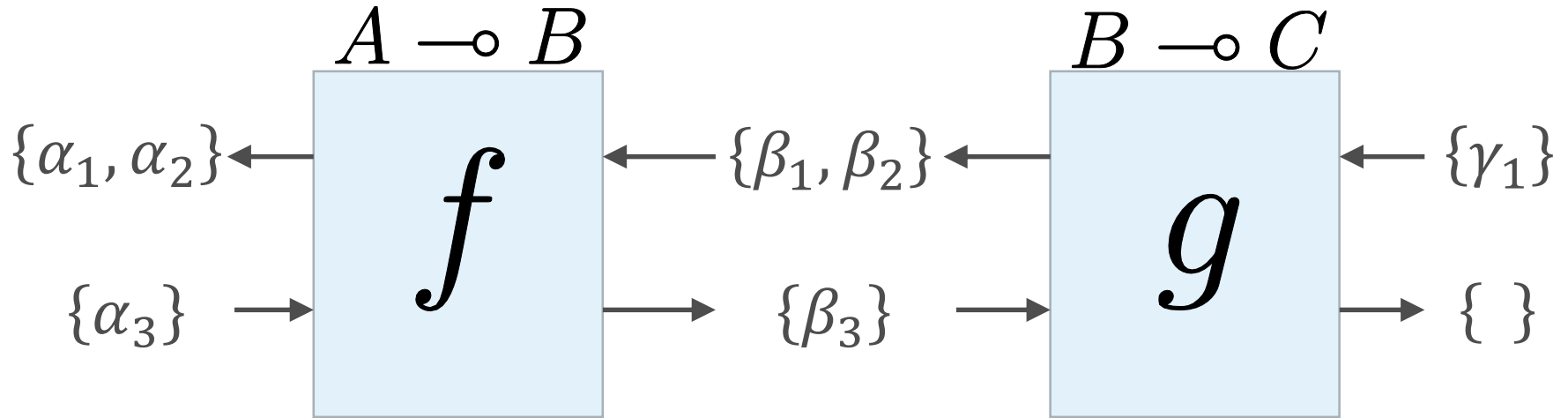
$$g(\beta_3) = \beta_2$$

$$g(\gamma_1) = \beta_1$$

$$\alpha_3 \xrightarrow{f} \alpha_2$$

合成の例

$$A = (\{\alpha_1, \alpha_2\}, \{\alpha_3\}) \quad B = (\{\beta_1, \beta_2\}, \{\beta_3\}) \quad C = (\{\gamma_1\}, \emptyset)$$



$$f(\alpha_3) = \alpha_2$$

$$f(\beta_1) = \beta_3$$

$$f(\beta_2) = \alpha_1$$

$$g(\beta_3) = \beta_2$$

$$g(\gamma_1) = \beta_1$$

$$\gamma_1 \xrightarrow{g} \beta_1 \xrightarrow{f} \beta_3 \xrightarrow{g} \beta_2 \xrightarrow{f} \alpha_1$$

合成の形式的定義

定義 f を $A \multimap B$ の戦略、 g を $B \multimap C$ の戦略とし、
 $M = M_A^O + M_A^P + M_B^O + M_B^P + M_C^O + M_C^P$ とする。

単射部分関数 $\mu: M \multimap M$ を次のように定義する

$$\mu(x) = \begin{cases} f(x) & (\text{if } x \in M_A^P + M_B^O) \\ g(x) & (\text{if } x \in M_B^P + M_C^O) \\ \text{undefined} & (\text{otherwise}) \end{cases}$$

合成 $g \circ f$ は $A \multimap C$ の戦略であって、次で定義される

$$M_A^O + M_C^P \ni x \xrightarrow{g \circ f} y \in M_A^P + M_C^O$$

$$\Leftrightarrow \exists n \in \{1, 2, 3, \dots\}. \mu^n(x) = y$$

補足：Int 構成 [Joyal+ 96, Abramsky+ 02]

$\mathcal{G}_{\sqcup}$ の構成は Int 構成の例

- 集合と単射部分関数の圏を $PInj$ とすると、 $PInj$ は traced symmetric monoidal
 - モノイド積は集合の直和
- $\mathcal{G}_{\sqcup}$ は $\text{Int}(PInj)$

GoI 解釈 [Girard 89, 88, 95] と深く関わる

アウトライン

1. 単純なゲーム G_2 。

2. G_2 による affine λ 計算の解釈

- 型の解釈
- 項の解釈（帰納的）
- 項の解釈（直接的）

3. 問題点: 項に対応しない戦略がある

型・型判断の解釈

型の構造に関する帰納法による

- 基底型 : $\llbracket o \rrbracket := (\{\star\}, \emptyset)$

※ 左の集合の要素は適宜名前を変える

- 関数型 : $\llbracket T \multimap U \rrbracket := \llbracket T \rrbracket \multimap \llbracket U \rrbracket$

- 型環境 : $\llbracket x_1 : T_1, \dots, x_n : T_n \rrbracket$
 $= \llbracket T_1 \rrbracket \otimes \dots \otimes \llbracket T_n \rrbracket$

- 型判断 $\llbracket \Gamma \vdash T \rrbracket = \llbracket \Gamma \rrbracket \multimap \llbracket T \rrbracket$

型の解釈の例

基底型の出現の区別するため、異なる名前を付ける

$$o \multimap o \multimap o \quad \longmapsto \quad \gamma \multimap \beta \multimap \alpha$$

例

$$\begin{aligned} \llbracket \beta \multimap \alpha \rrbracket &= \llbracket \beta \rrbracket \multimap \llbracket \alpha \rrbracket \\ &= (\{\beta\}, \emptyset) \multimap (\{\alpha\}, \emptyset) \\ &= (\emptyset + \{\alpha\}, \{\beta\} + \emptyset) \\ &= (\{\alpha\}, \{\beta\}) \end{aligned}$$

型の解釈の例

基底型の出現の区別するため、異なる名前を付ける

$$o \multimap o \multimap o \quad \longmapsto \quad \gamma \multimap \beta \multimap \alpha$$

例

$$\llbracket \beta \multimap \alpha \rrbracket = (\{\alpha\}, \{\beta\})$$

$$\begin{aligned} \llbracket \gamma \multimap \beta \multimap \alpha \rrbracket &= \llbracket \gamma \rrbracket \multimap \llbracket \beta \multimap \alpha \rrbracket \\ &= (\{\gamma\}, \emptyset) \multimap (\{\alpha\}, \{\beta\}) \\ &= (\emptyset + \{\alpha\}, \{\gamma\} + \{\beta\}) \\ &= (\{\alpha\}, \{\gamma, \beta\}) \end{aligned}$$

型の解釈の例

基底型の出現の区別するため、異なる名前を付ける

$$o \multimap o \multimap o \quad \longmapsto \quad \gamma \multimap \beta \multimap \alpha$$

例

$$\llbracket \beta \multimap \alpha \rrbracket = (\{\alpha\}, \{\beta\})$$

$$\begin{aligned} \llbracket (\beta \multimap \alpha) \multimap \gamma \rrbracket &= \llbracket \beta \multimap \alpha \rrbracket \multimap \llbracket \gamma \rrbracket \\ &= (\{\alpha\}, \{\beta\}) \multimap (\{\gamma\}, \emptyset) \\ &= (\{\beta\} + \{\gamma\}, \{\alpha\} + \emptyset) \\ &= (\{\beta, \gamma\}, \{\alpha\}) \end{aligned}$$

型の解釈の例

基底型の出現の区別するため、異なる名前を付ける

$$o \multimap o \multimap o \quad \longmapsto \quad \gamma \multimap \beta \multimap \alpha$$

例

$$\llbracket (\beta \multimap \alpha) \multimap \gamma \rrbracket = (\{\beta, \gamma\}, \{\alpha\})$$

$$\begin{aligned} \llbracket ((\beta \multimap \alpha) \multimap \gamma) \multimap \delta \rrbracket &= \llbracket \beta \multimap \alpha \rrbracket \multimap \llbracket \gamma \rrbracket \multimap \llbracket \delta \rrbracket \\ &= (\{\beta, \gamma\}, \{\alpha\}) \multimap (\{\delta\}, \emptyset) \\ &= (\{\alpha\} + \{\delta\}, \{\beta, \gamma\} + \emptyset) \\ &= (\{\alpha, \delta\}, \{\beta, \gamma\}) \end{aligned}$$

型の直接的解釈

$$\llbracket T \rrbracket = (\underbrace{\text{共変位置の基底型}}_{M_{\llbracket T \rrbracket}^O}, \underbrace{\text{反変位置の基底型}}_{M_{\llbracket T \rrbracket}^P})$$

例

$$\llbracket \beta \multimap \alpha \rrbracket = (\{\alpha\}, \{\beta\})$$

$$\llbracket \gamma \multimap \beta \multimap \alpha \rrbracket = (\{\alpha\}, \{\gamma, \beta\})$$

$$\llbracket (\gamma \multimap \beta) \multimap \alpha \rrbracket = (\{\alpha, \gamma\}, \{\beta\})$$

$$\llbracket ((\delta \multimap \gamma) \multimap \beta) \multimap \alpha \rrbracket = (\{\alpha, \gamma\}, \{\beta, \delta\})$$

アウトライン

1. 単純なゲーム G_2 。

2. G_2 による affine λ 計算の解釈

- 型の解釈
- 項の解釈（帰納的）
- 項の解釈（直接的）

3. 問題点: 項に対応しない戦略がある

項の解釈

項 $\Gamma \vdash M : T$ の解釈はゲーム $\llbracket \Gamma \vdash T \rrbracket$ の戦略

$$\llbracket x_1 : T_1, \dots, x_n : T_n \vdash U \rrbracket = (\llbracket T_1 \rrbracket \otimes \dots \otimes \llbracket T_n \rrbracket) \multimap \llbracket U \rrbracket$$

- 型判断の導出の構造に関する帰納法で定義

$$\left[\frac{}{x : T \vdash x : T} \right] = \text{id}_{\llbracket T \rrbracket}$$

$$\left[\frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x. M : T \multimap U} \right] = \Lambda(\llbracket \Gamma, x : T \vdash M : U \rrbracket)$$

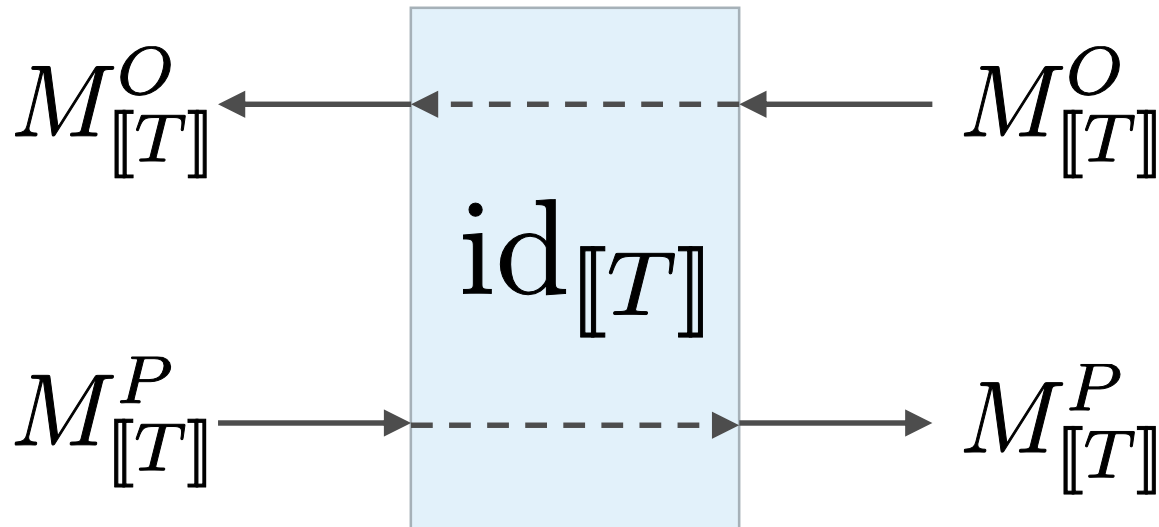
$$\begin{aligned} \left[\frac{\Gamma \vdash M : T \multimap U \quad \Delta \vdash N : T}{\Gamma, \Delta \vdash M N : U} \right] \\ = \mathbf{ev} \circ (\llbracket \Gamma \vdash M : T \multimap U \rrbracket \otimes \llbracket \Delta \vdash N : U \rrbracket) \end{aligned}$$

変数の解釈

$$\llbracket T \rrbracket = (M_{\llbracket T \rrbracket}^O, M_{\llbracket T \rrbracket}^P) \text{ とすると } \llbracket x : T \vdash x : T \rrbracket = \text{id}_{\llbracket T \rrbracket}$$

$$\llbracket x : T \vdash T \rrbracket = (M_{\llbracket T \rrbracket}^P + M_{\llbracket T \rrbracket}^O, M_{\llbracket T \rrbracket}^O + M_{\llbracket T \rrbracket}^P)$$

$$\text{id}_{\llbracket T \rrbracket}(a) = a$$



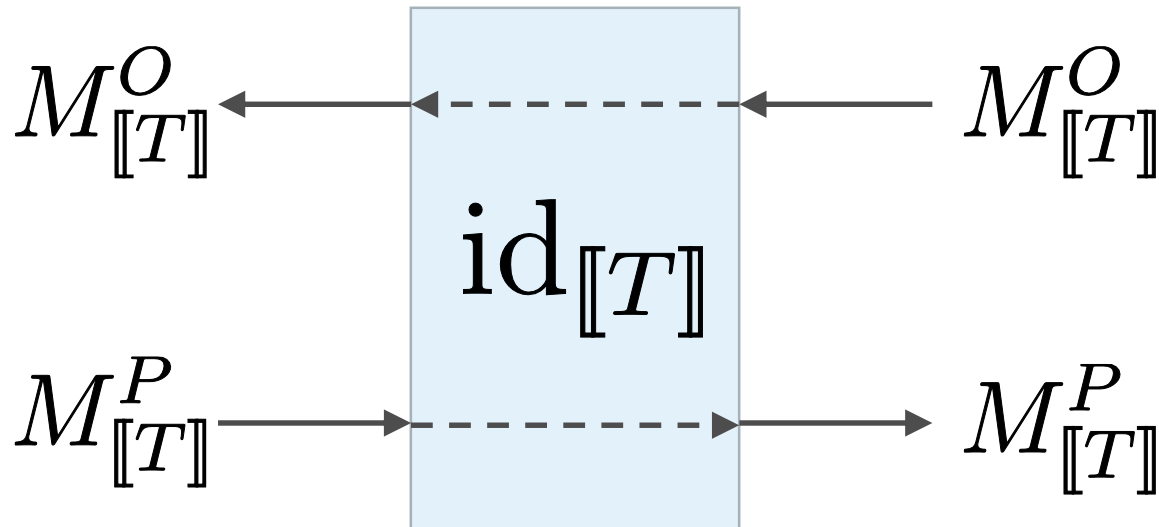
変数の解釈

$$\llbracket T \rrbracket = (M_{\llbracket T \rrbracket}^O, M_{\llbracket T \rrbracket}^P) \text{ とすると } \llbracket x : T \vdash x : T \rrbracket = \text{id}_{\llbracket T \rrbracket}$$

$$\llbracket x : T \vdash T \rrbracket = (M_{\llbracket T \rrbracket}^P + M_{\llbracket T \rrbracket}^O, M_{\llbracket T \rrbracket}^O + M_{\llbracket T \rrbracket}^P)$$

$$\text{id}_{\llbracket T \rrbracket}(\mathbf{inl}(a)) = \mathbf{inr}(a)$$

$$\text{id}_{\llbracket T \rrbracket}(\mathbf{inr}(a)) = \mathbf{inl}(a)$$



λ抽象の解釈（準備）

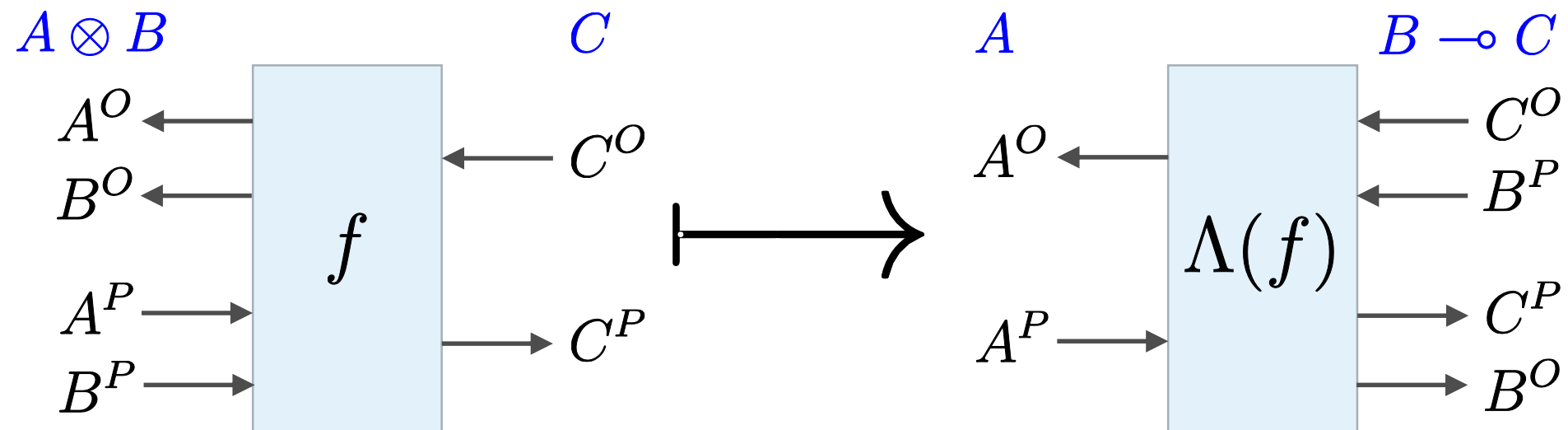
次の一対一対応が存在

$\Lambda : (A \otimes B \multimap C \text{ の戦略}) \rightarrow (A \multimap (B \multimap C) \text{ の戦略})$

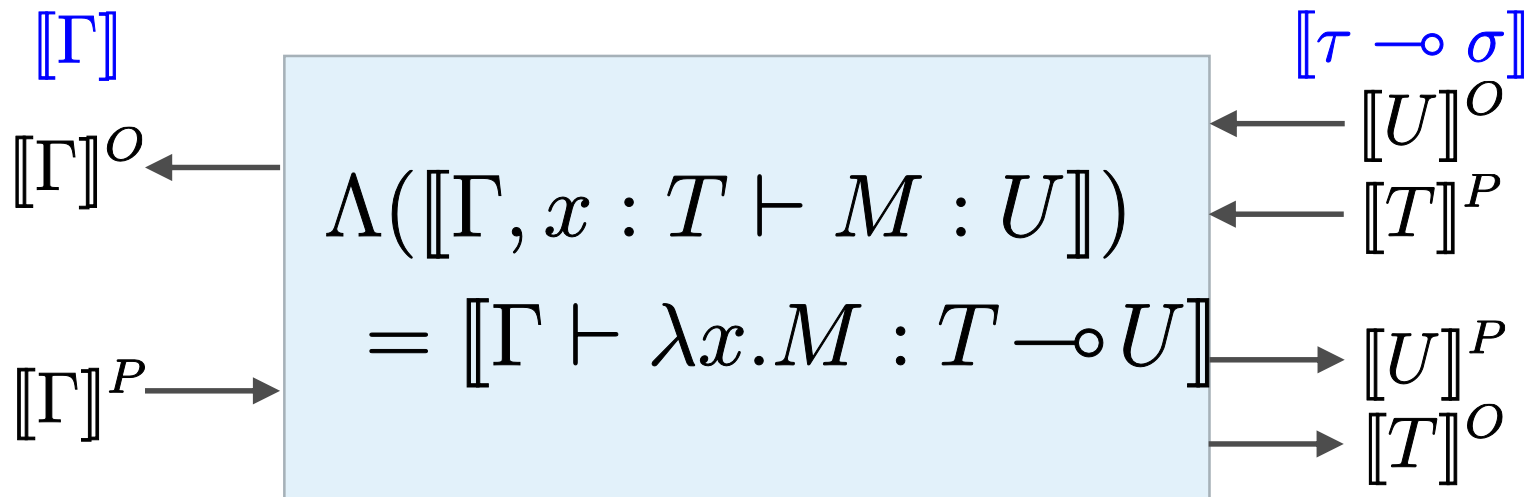
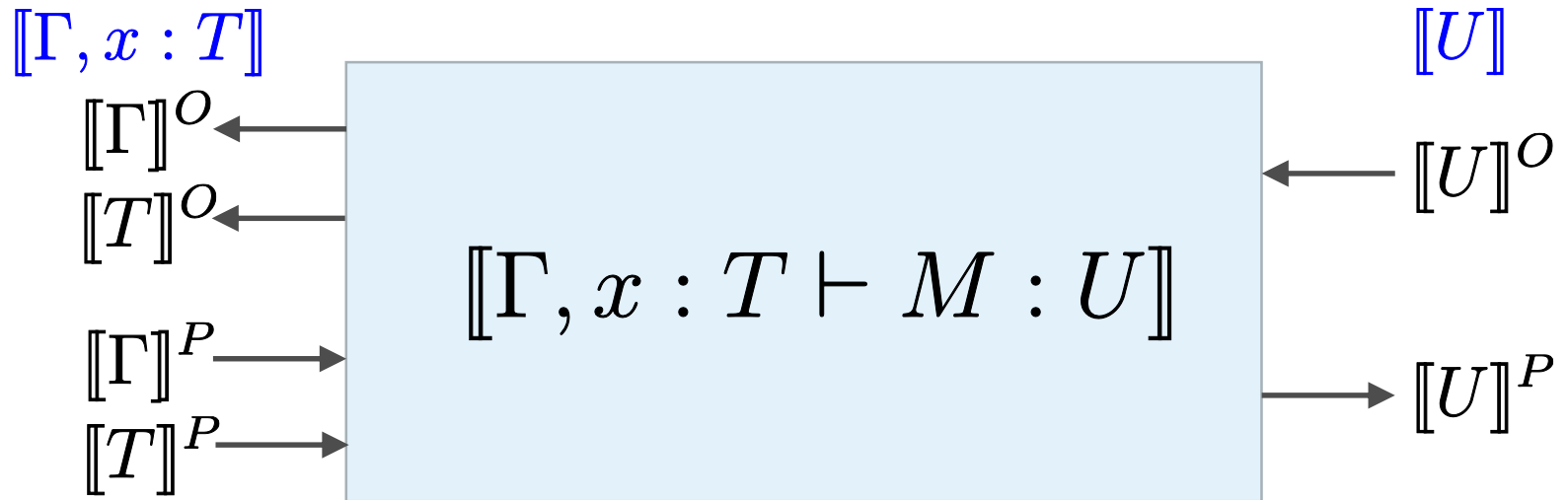
- $A \otimes B \multimap C$ と $A \multimap (B \multimap C)$ は本質的には同じため

$$A \otimes B \multimap C = (A^P + B^P + C^O, A^O + B^O + C^P)$$

$$A \multimap (B \multimap C) = (A^P + B^P + C^O, A^O + B^O + C^P)$$

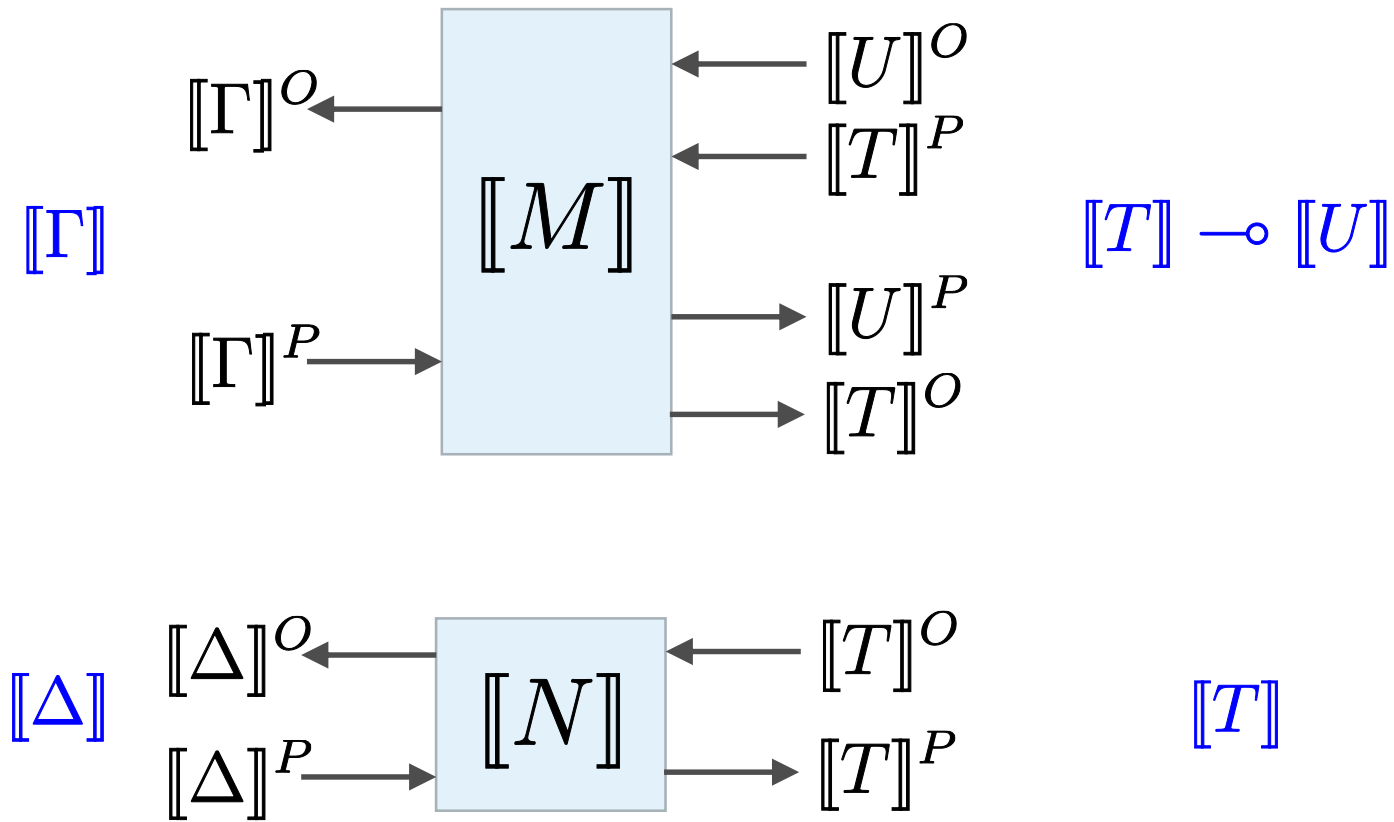


λ抽象の解釈



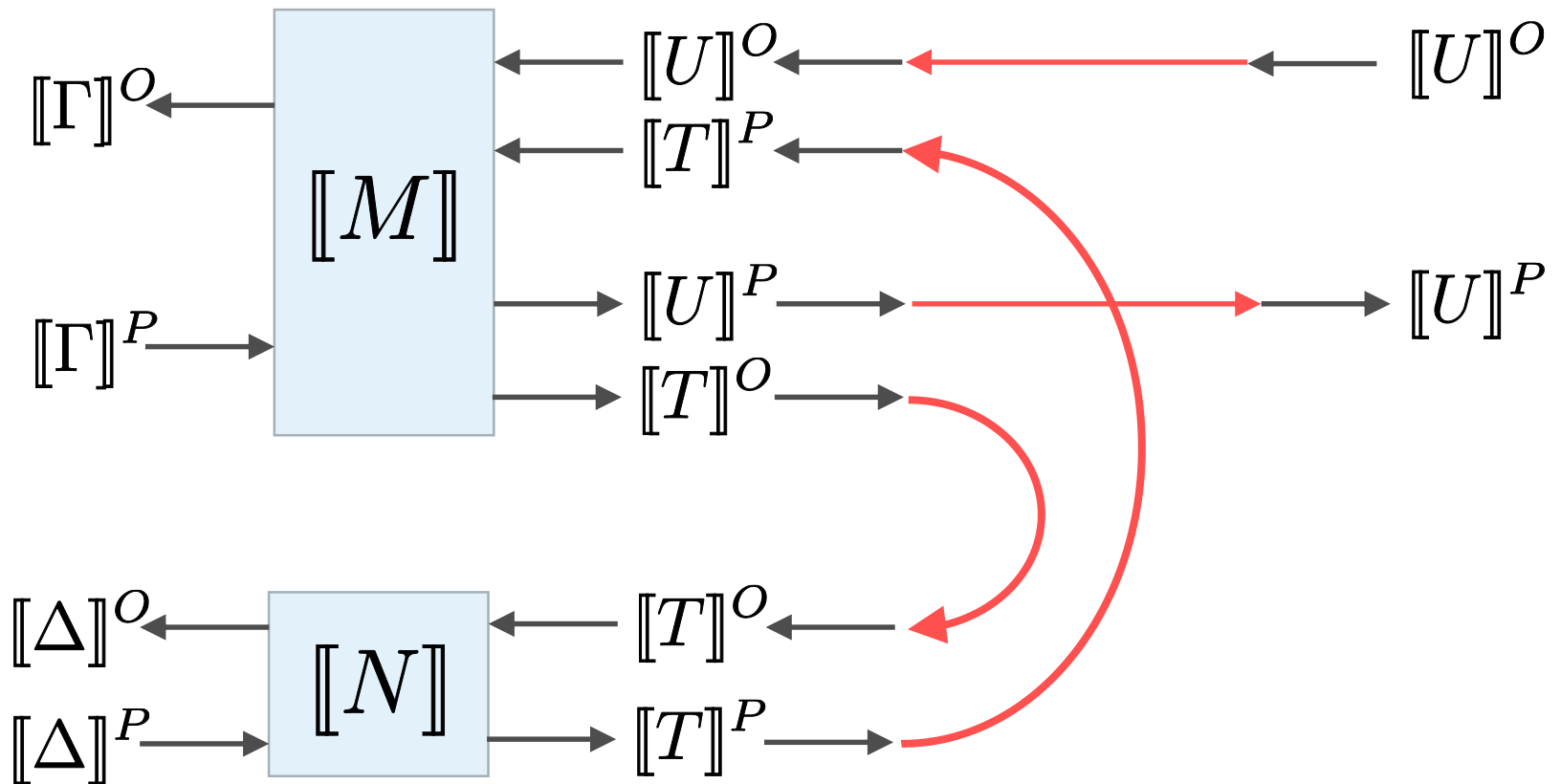
関数適用の解釈

$$\left[\frac{\Gamma \vdash M : T \multimap U \quad \Delta \vdash N : T}{\Gamma, \Delta \vdash M N : U} \right] = \mathbf{ev} \circ ([M] \otimes [N])$$



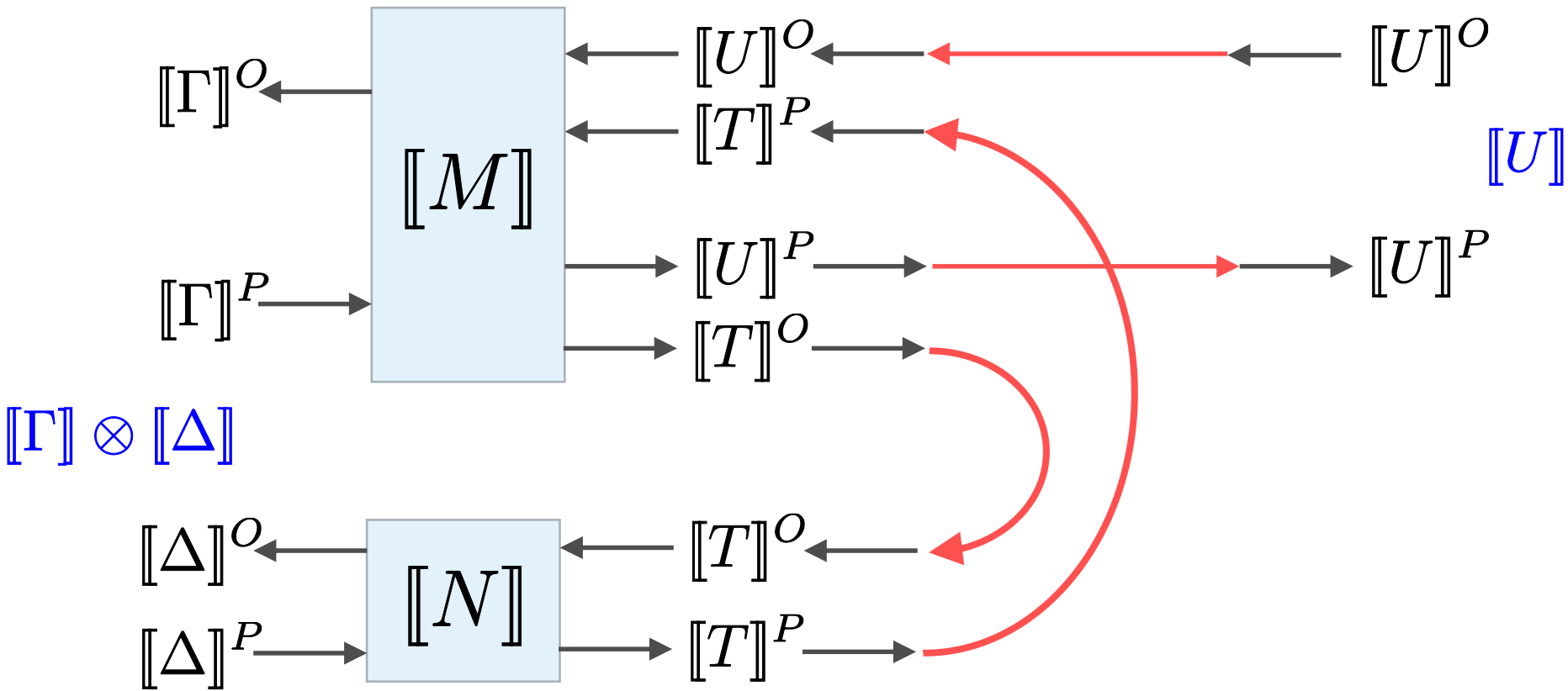
関数適用の解釈

$$\left[\frac{\Gamma \vdash M : T \multimap U \quad \Delta \vdash N : T}{\Gamma, \Delta \vdash M N : U} \right] = \mathbf{ev} \circ ([M] \otimes [N])$$



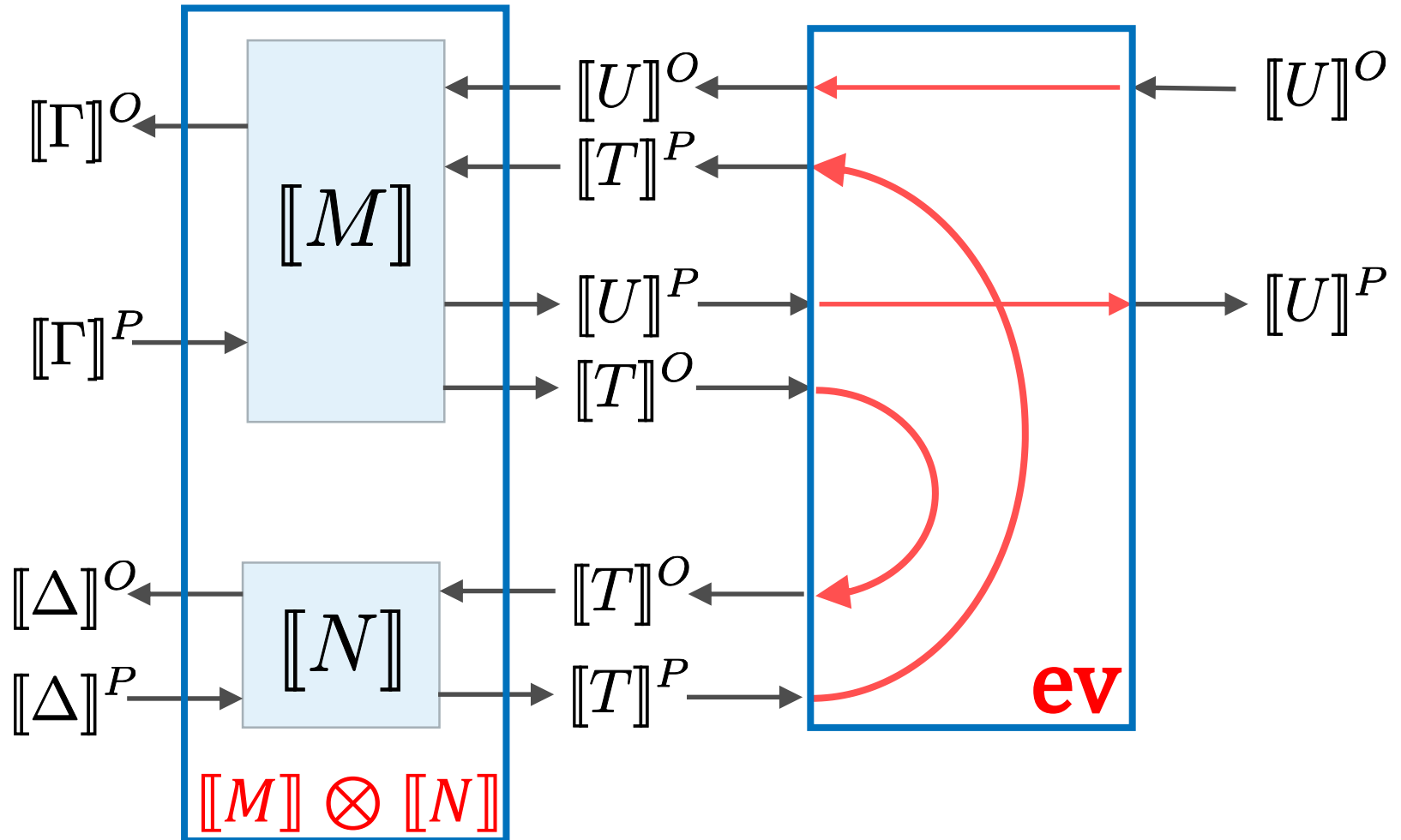
関数適用の解釈

$$\left[\frac{\Gamma \vdash M : T \multimap U \quad \Delta \vdash N : T}{\Gamma, \Delta \vdash M N : U} \right] = \mathbf{ev} \circ ([M] \otimes [N])$$



関数適用の解釈

$$\left[\frac{\Gamma \vdash M : T \multimap U \quad \Delta \vdash N : T}{\Gamma, \Delta \vdash M N : U} \right] = \mathbf{ev} \circ ([M] \otimes [N])$$



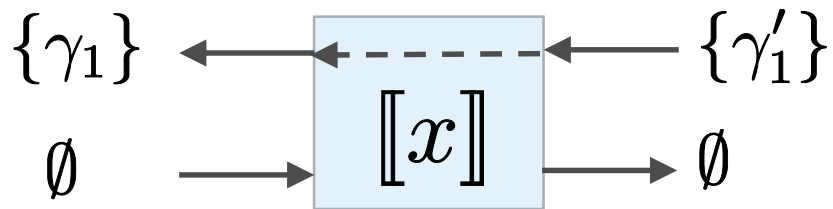
解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f \ x \ g : \delta_1$$

解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f \ x \ g : \delta_1$$

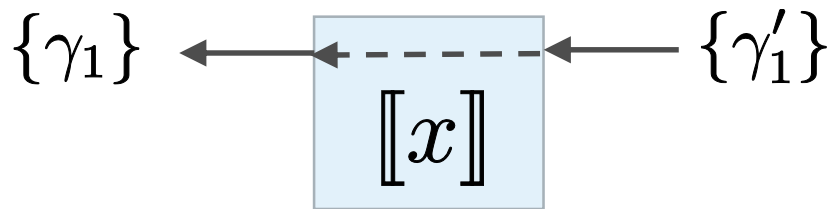
$$\llbracket x : \gamma_1 \vdash x : \gamma'_1 \rrbracket$$



解釈の例

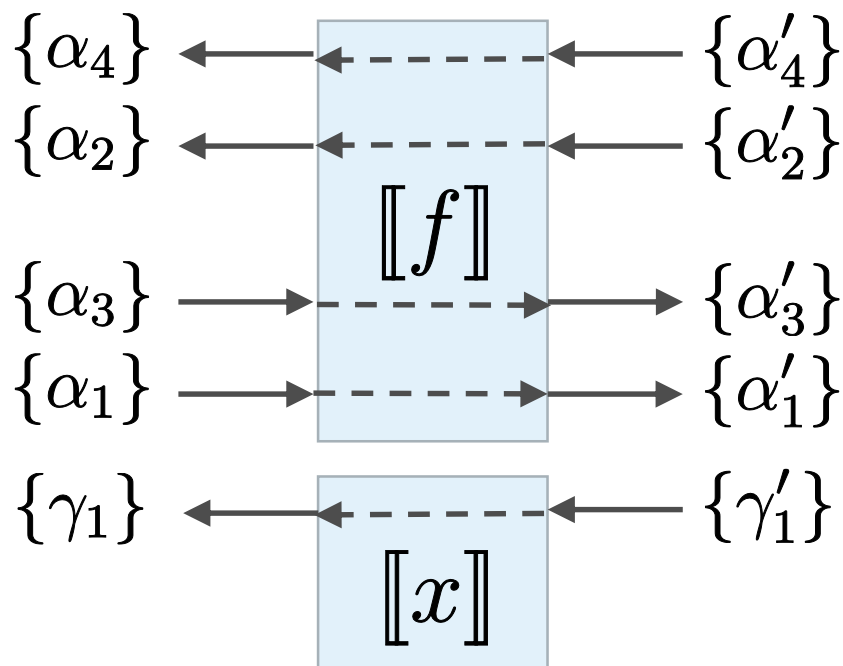
$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$

$$\llbracket x : \gamma_1 \vdash x : \gamma'_1 \rrbracket$$



解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$

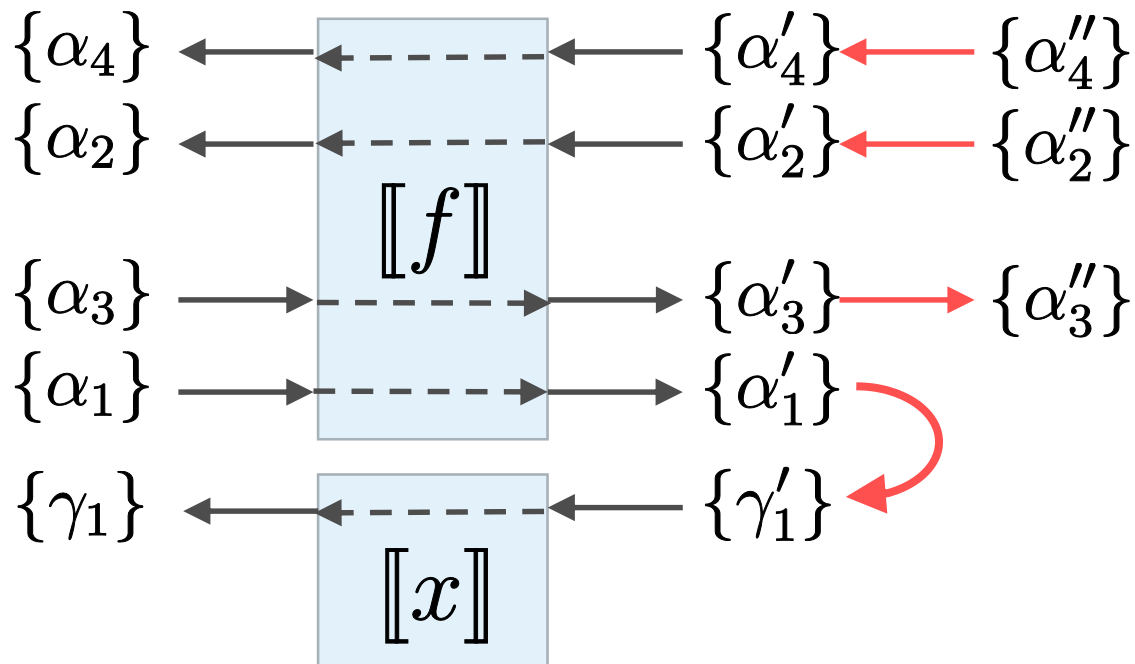


$$\llbracket f : T \vdash f : T' \rrbracket$$

$$T = \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4$$

解釈の例

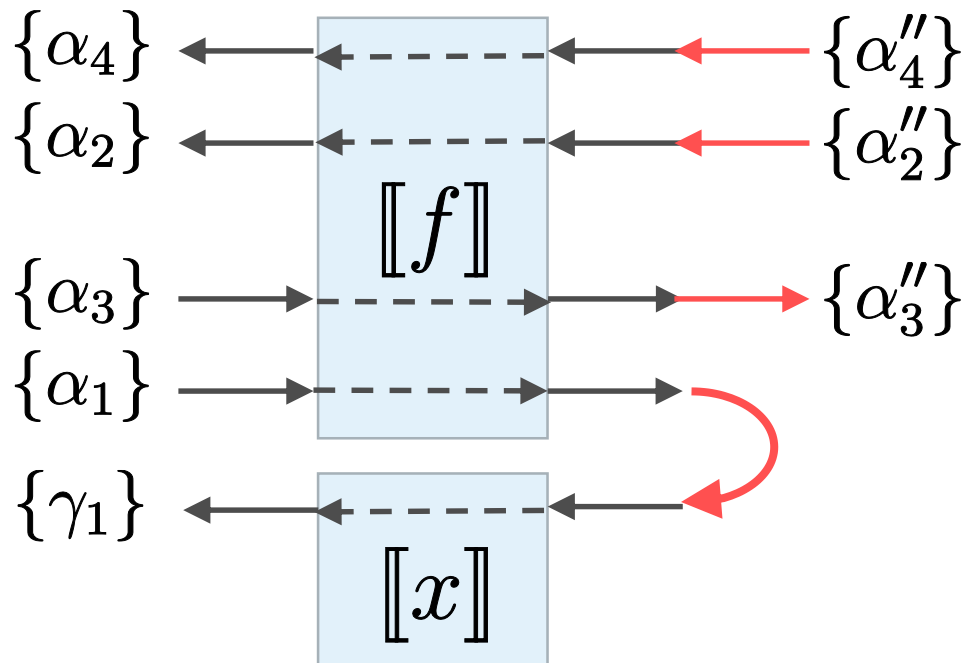
$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$



$$\left[\begin{array}{c} \dots \vdash f : \alpha'_1 \multimap (\alpha'_2 \multimap \alpha'_3) \multimap \alpha'_4 \\ \dots \vdash x : \gamma'_1 \\ \hline \dots \vdash f x : (\alpha''_2 \multimap \alpha''_3) \multimap \alpha''_4 \end{array} \right]$$

解釈の例

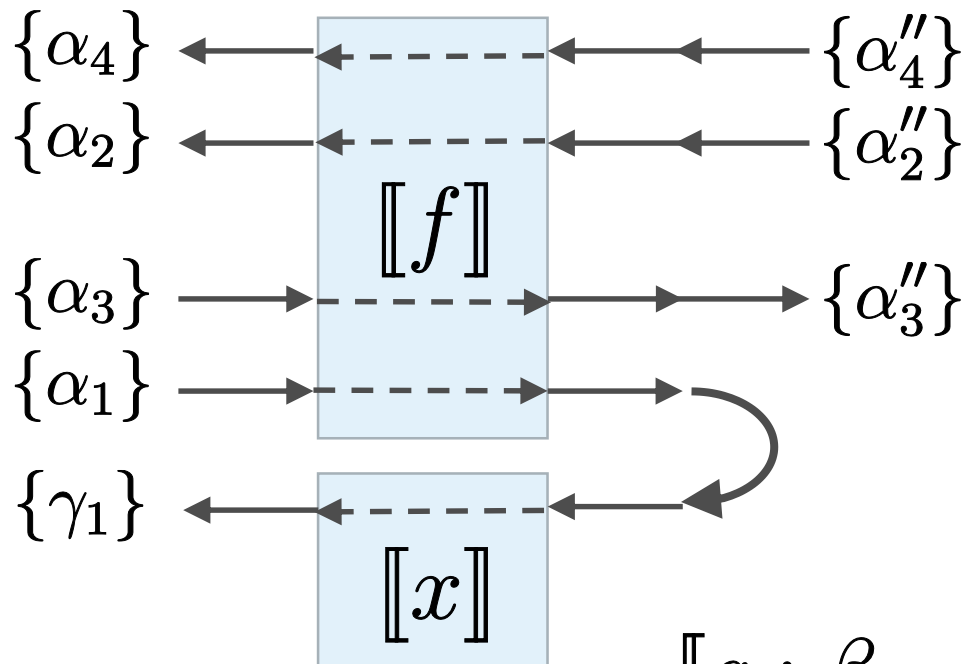
$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$



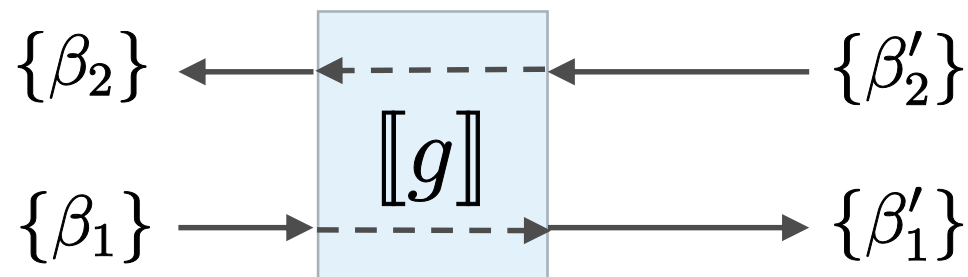
$$\left[\frac{\begin{array}{l} \dots \vdash f : \alpha'_1 \multimap (\alpha'_2 \multimap \alpha'_3) \multimap \alpha'_4 \\ \dots \vdash x : \gamma'_1 \end{array}}{\dots \vdash f x : (\alpha''_2 \multimap \alpha''_3) \multimap \alpha''_4} \right]$$

解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f \ x \ g : \delta_1$$

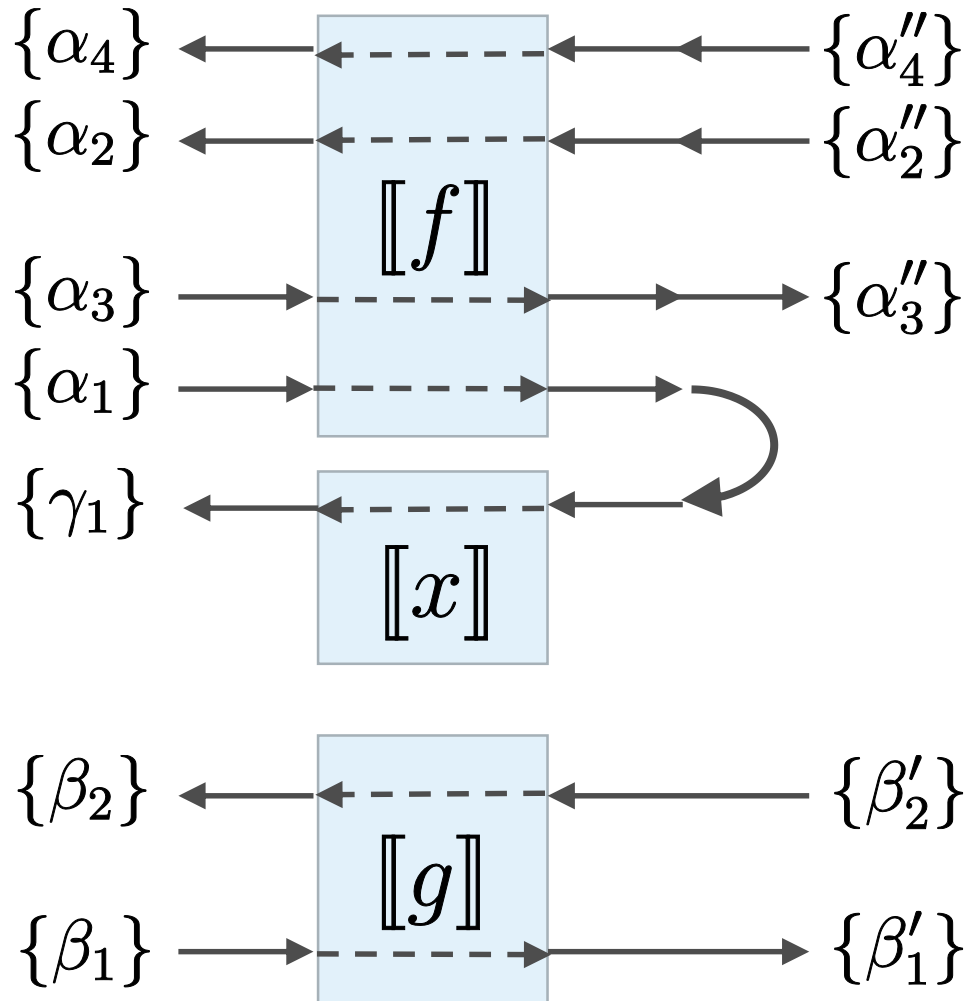


$$\llbracket g : \beta_1 \multimap \beta_2 \vdash g : \beta'_1 \multimap \beta'_2 \rrbracket$$



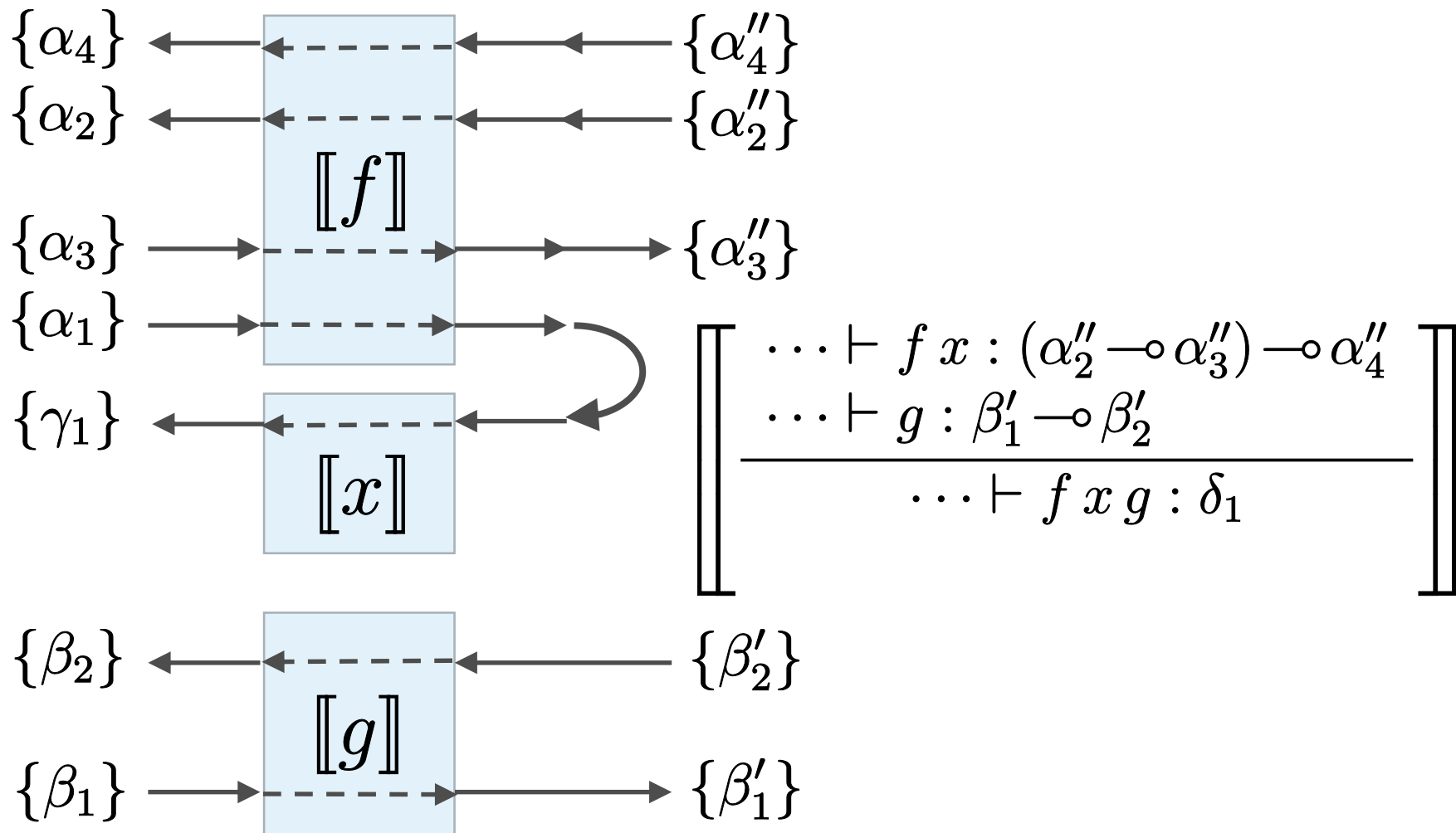
解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$



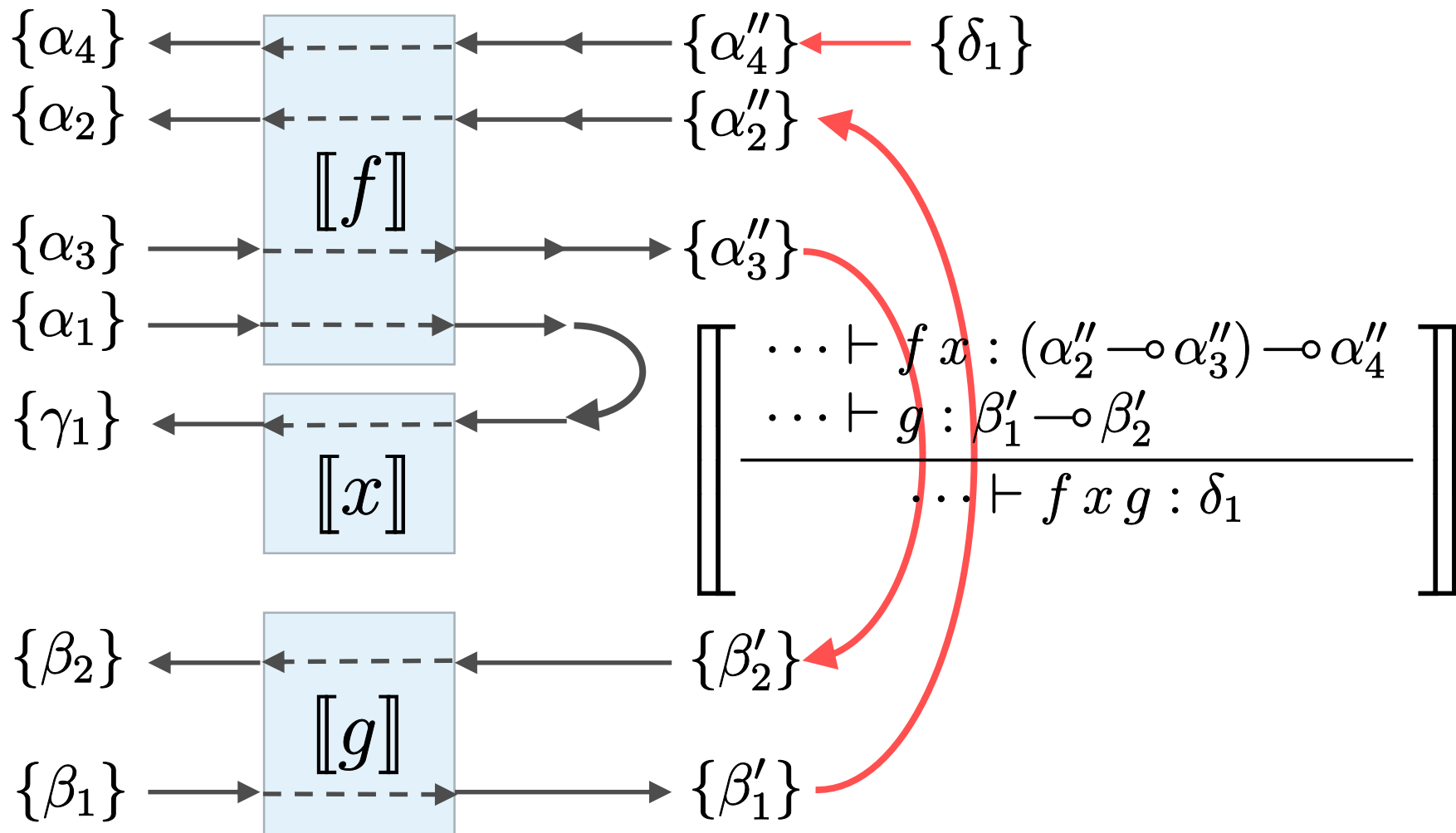
解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$



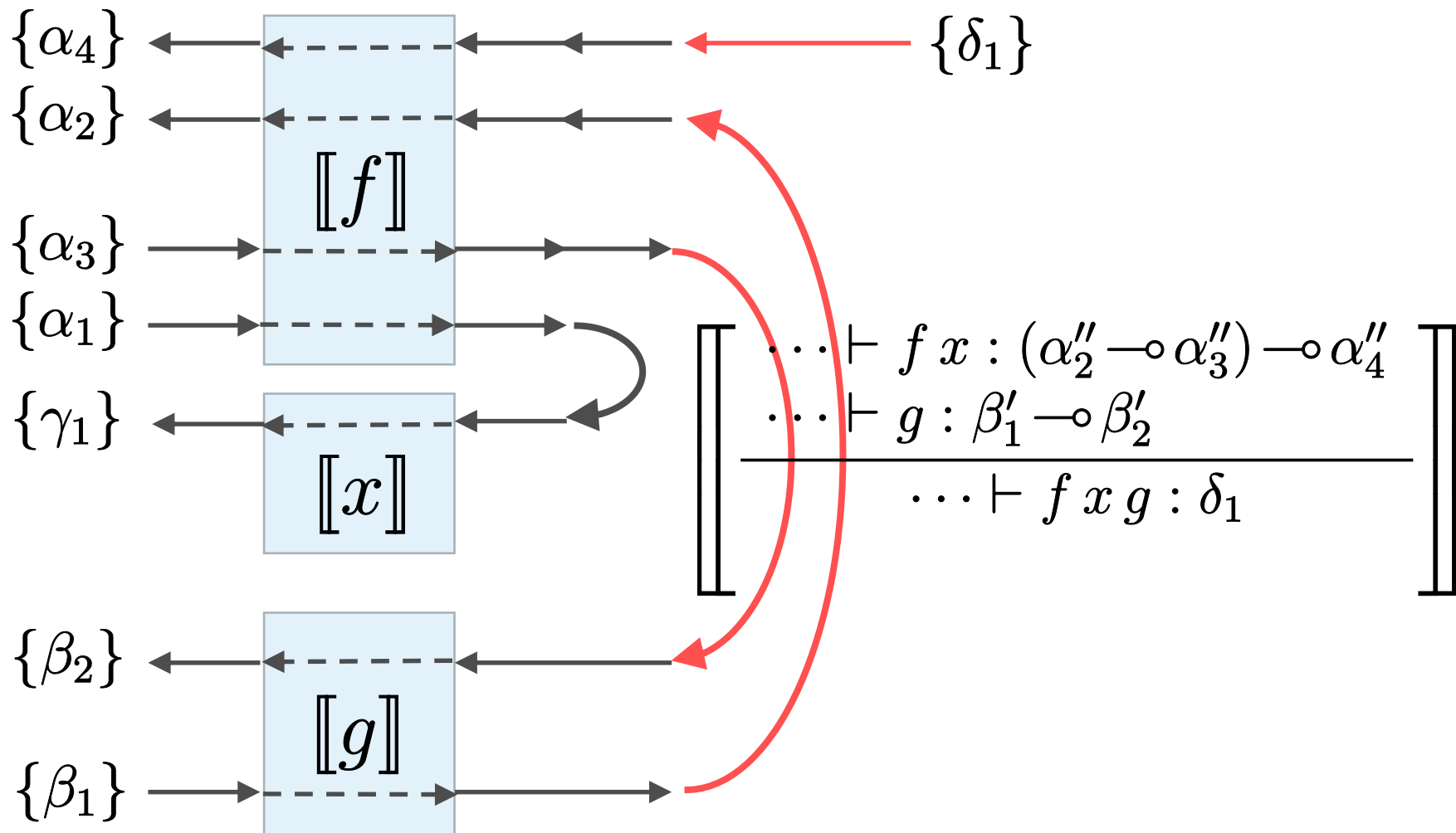
解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$



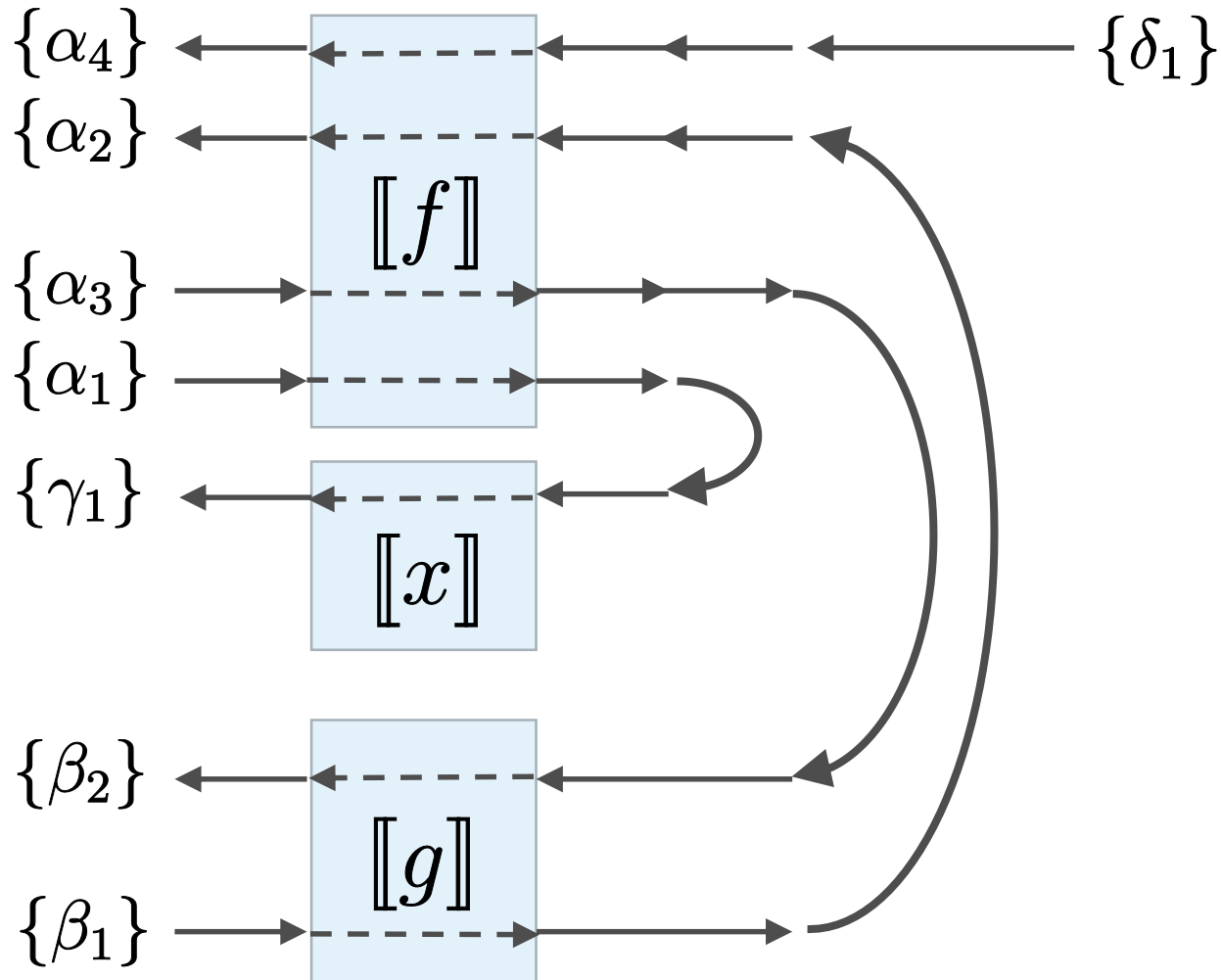
解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f x g : \delta_1$$



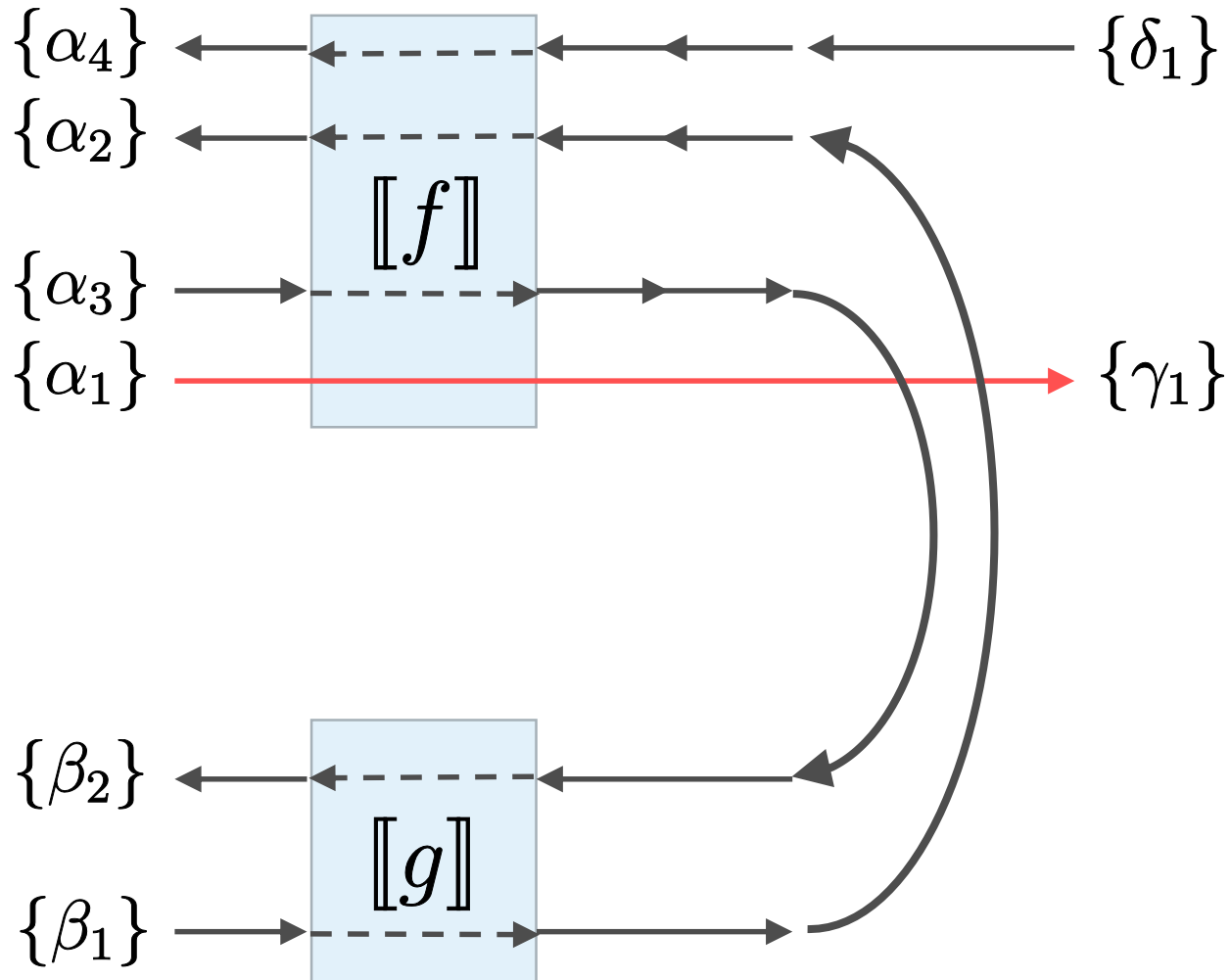
解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2, x : \gamma_1 \vdash f \ x \ g : \delta_1$$



解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2 \vdash \lambda x. f \ x \ g : \gamma_1 \multimap \delta_1$$



$\mathcal{G}_2$ による解釈の性質

定理 (モデルの妥当性)

$$M =_{\beta\eta} N \text{ ならば } \llbracket M \rrbracket = \llbracket N \rrbracket$$

定理 (faithfulness)

$$\llbracket M \rrbracket = \llbracket N \rrbracket \text{ ならば } M =_{\beta\eta} N$$

アウトライン

1. 単純なゲーム G_2 。

2. G_2 による affine λ 計算の解釈

- 型の解釈
- 項の解釈（帰納的）
- 項の解釈（直接的）

3. 問題点: 項に対応しない戦略がある

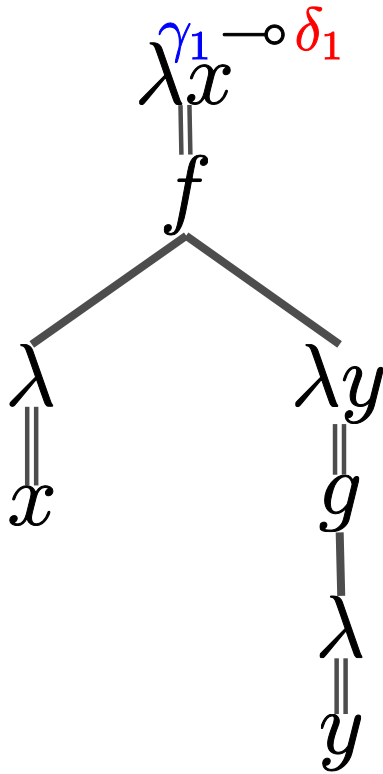
直接的解釈

1. 項を標準形に変換する
2. 各 λ 抽象 と 変数 に move を割り当てる
 - λ 抽象 には O-move
 - 変数 には P-move
3. $\lambda \vec{x}. y \vec{M}$ の形の部分に注目
 - λx に a が、 y に b が割り当てられていたら、 $a \mapsto b$

直接解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2 \vdash \lambda x. f \ x \ g : \gamma_1 \multimap \delta_1$$

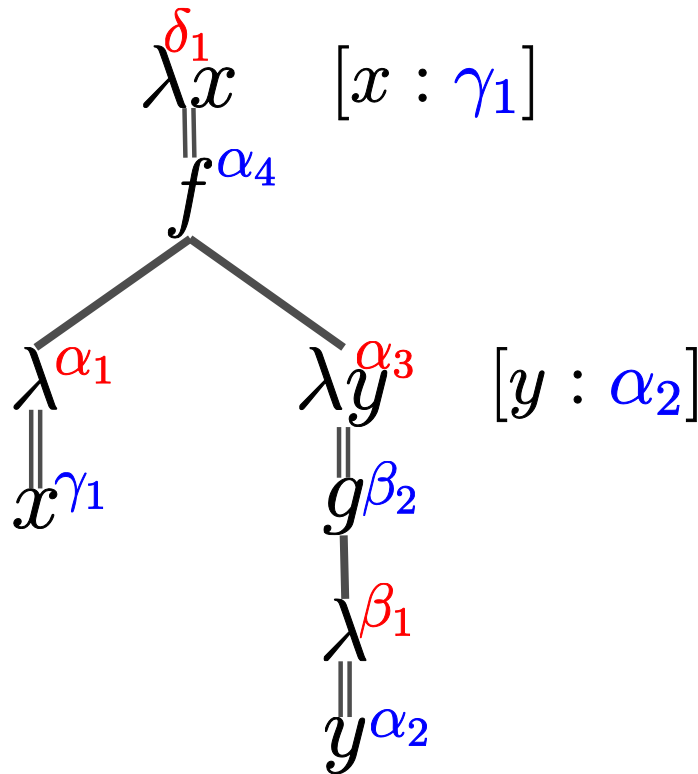
$$\lambda x. f \ x \ g =_{\beta\eta} \lambda x. f \ x \ (\lambda y. g \ y)$$



直接解釈の例

$$f : \alpha_1 \multimap (\alpha_2 \multimap \alpha_3) \multimap \alpha_4, g : \beta_1 \multimap \beta_2 \vdash \lambda x. f \ x \ g : \gamma_1 \multimap \delta_1$$

$$\lambda x. f \ x \ g =_{\beta\eta} \lambda x. f \ x \ (\lambda y. g \ y)$$



$$\delta_1 \mapsto \alpha_4$$

$$\alpha_1 \mapsto \gamma_1$$

$$\alpha_3 \mapsto \beta_2$$

$$\beta_1 \mapsto \alpha_2$$

アウトライン

1. 単純なゲーム $\mathcal{G}_2$ 。
2. $\mathcal{G}_2$ による affine λ 計算の解釈
3. 問題点: 項に対応しない戦略がある

$\mathcal{G}_2$ の戦略は definable とは限らない

命題

ある型 T と $\llbracket T \rrbracket$ の戦略 f が存在して

$$\forall (\emptyset \vdash M : T). \llbracket M \rrbracket \neq f$$

例 1 $\tau = ((\alpha \multimap \beta) \multimap \gamma) \multimap \delta$

$$\llbracket \tau \rrbracket = (\{\beta, \delta\}, \{\alpha, \gamma\})$$

$$f(\beta) = \gamma \quad f(\delta) = \alpha$$

$\mathcal{G}_2$ の戦略は definable とは限らない

命題

ある型 T と $\llbracket T \rrbracket$ の戦略 f が存在して

$$\forall (\emptyset \vdash M : T). \llbracket M \rrbracket \neq f$$

例 2 $\tau = ((\alpha_1 \multimap \beta_1) \multimap (\alpha_2 \multimap \beta_2) \multimap \gamma) \multimap \delta$

$$\llbracket \tau \rrbracket = (\{\beta_1, \beta_2, \delta\}, \{\alpha_1, \alpha_2, \gamma\})$$

$$f(\beta_1) = \alpha_2 \quad f(\beta_2) = \alpha_1 \quad f(\delta) = \gamma$$

$\mathcal{G}_2$ の戦略は definable とは限らない

命題

ある型 T と $\llbracket T \rrbracket$ の戦略 f が存在して

$$\forall (\emptyset \vdash M : T). \llbracket M \rrbracket \neq f$$

例 3 $\tau = \alpha_1 \multimap \alpha_2 \multimap (\beta \multimap \gamma) \multimap \delta$

$$\llbracket \tau \rrbracket = (\{\beta, \delta\}, \{\alpha_1, \alpha_2, \gamma\})$$

$$f(\beta) = \alpha_1 \quad f(\delta) = \alpha_2$$

理由：変数のスコープ

$$T = ((\alpha \multimap \beta) \multimap \gamma) \multimap \delta$$

$$f(\beta) = \gamma$$

$$f(\delta) = \alpha$$

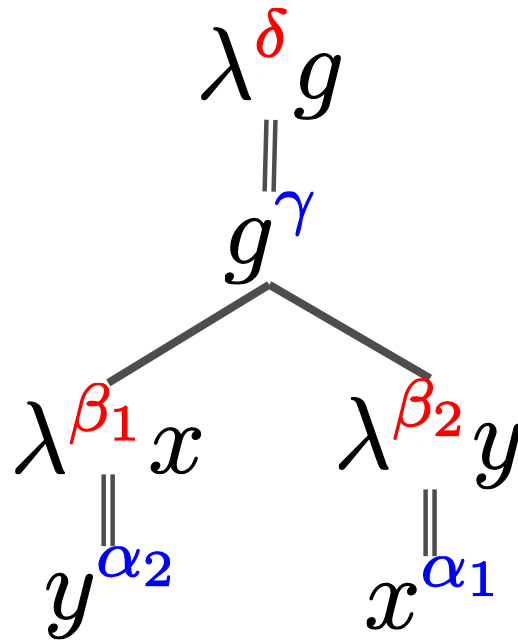
$$\lambda^{\delta} g$$
$$\parallel$$
$$?\alpha$$

α を結果に持つ変数は
まだ導入されていない

理由：変数のスコープ

$$T = ((\alpha_1 \multimap \beta_1) \multimap (\alpha_2 \multimap \beta_2) \multimap \gamma) \multimap \delta$$

$$f(\beta_1) = \alpha_2 \quad f(\beta_2) = \alpha_1 \quad f(\delta) = \gamma$$



y のスコープの外

理由：変数のスコープ

$$T = \alpha_1 \multimap \alpha_2 \multimap (\beta \multimap \gamma) \multimap \delta$$

$$f(\beta) = \alpha_1 \quad f(\delta) = \alpha_2$$

$$\lambda^{\delta} x y g$$
$$\quad \parallel$$
$$y^{\alpha_2}$$

相手がスコープを守っている限り、 β は使われない

Affine λ 計算のゲーム意味論 II

<u>対象言語</u>	<u>ゲーム</u>	<u>線形資源計算</u>
Affine λ	$\left\{ \begin{array}{l} \mathcal{G}_{\rightarrow}^- \\ \mathcal{G}_{\rightarrow}^- \end{array} \right.$	$\cong \partial_0 \lambda_{\rightarrow}$
$\lambda_{\rightarrow}$	$\mathcal{G}$	$\cong \partial_0 \lambda_{\rightarrow}$
PCF	$\mathcal{G}^{\text{QA}}$	$\cong \partial_0 \text{PCF}$

アウトライン

$\mathcal{G}_\square$ に変数スコープの概念を導入する

1. 戦略の move 列表現

- スコープの概念の導入に便利

2. 変数スコープの機構を持つゲーム $\mathcal{G}_\square$

3. $\mathcal{G}_\square$ による affine λ 計算の解釈

アウトライン

$\mathcal{G}_\omega$ に変数スコープの概念を導入する

1. 戦略の move 列表現

- 戦略の move 列表現
- Move 列表現と単射部分関数表現の等価性
- Move 列表現された戦略の合成

2. 変数スコープの機構を持つゲーム $\mathcal{G}_\omega$

3. $\mathcal{G}_\omega$ による affine λ 計算の解釈

Move 列の集合としての戦略

戦略を到達しうる局面 (= move列) の集合として表現

定義 ゲーム $A = (M_A^O, M_A^P)$ の戦略の **move列表現** $\hat{\sigma}$ は偶数長の affine 交代列の集合 $\hat{\sigma} \subseteq \hat{P}_A^{even}$ であって以下を満たすものと言う

$$\frac{}{\varepsilon \in \hat{\sigma}} \quad \frac{s \cdot a \cdot b \in \hat{\sigma}}{s \in \hat{\sigma}} \quad \frac{s \cdot a \cdot b \in \hat{\sigma} \quad s \cdot a \cdot b' \in \hat{\sigma}}{b = b'}$$

$$\frac{s \cdot a \cdot b \cdot a' \cdot b' \cdot t \in \hat{\sigma}}{s \cdot a' \cdot b' \cdot a \cdot b \cdot t \in \hat{\sigma}} \quad \frac{s \cdot t \cdot u \in \hat{P}_A \quad s \cdot t \in \hat{\sigma} \quad s \cdot u \in \hat{\sigma}}{s \cdot t \cdot u \in \hat{\sigma}}$$

(cf. [Mellies 06])

Move 列の集合としての戦略 (補足)

$$\frac{s \cdot a \cdot b \cdot a' \cdot b' \cdot t \in \hat{\sigma}}{s \cdot a' \cdot b' \cdot a \cdot b \cdot t \in \hat{\sigma}} \quad \frac{s \cdot t \cdot u \in \hat{P}_A \quad s \cdot t \in \hat{\sigma} \quad s \cdot u \in \hat{\sigma}}{s \cdot t \cdot u \in \hat{\sigma}}$$

上の規則は、以下の規則と同値：

$$\frac{s \cdot a \cdot b \in \hat{\sigma} \quad t \in \hat{\sigma} \quad t \cdot a \in \hat{P}_A}{t \cdot a \cdot b \in \hat{\sigma}} \quad (\text{history-free})$$

- 拡張を考える際に、下の規則では問題がある
一方、上の規則は (今日の範囲では) 最後まで使える

アウトライン

$\mathcal{G}_\omega$ に変数スコープの概念を導入する

1. 戦略の move 列表現

- 戦略の move 列表現
- Move 列表現と単射部分関数表現の等価性
- Move 列表現された戦略の合成

2. 変数スコープの機構を持つゲーム $\mathcal{G}_\omega$

3. $\mathcal{G}_\omega$ による affine λ 計算の解釈

ふたつの表現の等価性

単射部分関数から戦略の move 列表現へ

$M_A^O \rightarrow M_A^P$ の単射部分関数

戦略の move 列表現

$$(f : M_A^O \xrightarrow{\text{inj}} M_A^P) \longmapsto (\hat{\sigma}_f \subseteq \hat{P}_A^{\text{even}})$$

f が injection なので affine 交代列
(つまり $f(a_i)$ は相異なる)

$$\hat{\sigma}_f = \left\{ a_1 \cdot f(a_1) \cdot a_2 \cdot f(a_2) \cdots a_n \cdot f(a_n) \right. \\ \left. \begin{array}{l} \forall i. a_i \in M_A^O \\ \forall i, j. i \neq j \Rightarrow a_i \neq a_j \end{array} \right\}$$

ふたつの表現の等価性

戦略の move 列表現から単射部分関数へ

戦略の move 列表現

$M_A^O \rightarrow M_A^P$ の単射部分関数

$$(\hat{\sigma} \subseteq \hat{P}_A^{even}) \quad \longmapsto \quad (f_{\hat{\sigma}} : M_A^O \xrightarrow{inj} M_A^P)$$

$\hat{\sigma}$ が history-free なので
well-defined

$$\begin{aligned} f_{\hat{\sigma}}(a) = b & \stackrel{\text{def}}{\iff} \exists s. s \cdot a \cdot b \in \hat{\sigma} \\ & \iff a \cdot b \in \sigma \end{aligned}$$

系 $\hat{\sigma}$ は、その制限 $\hat{\sigma} \cap \{ a \cdot b \mid a \in M_A^O, b \in M_A^P \}$ で完全に決定される

アウトライン

$\mathcal{G}_\omega$ に変数スコープの概念を導入する

1. 戦略の move 列表現

- 戦略の move 列表現
- Move 列表現と単射部分関数表現の等価性
- Move 列表現された戦略の合成

2. 変数スコープの機構を持つゲーム $\mathcal{G}_\omega$

3. $\mathcal{G}_\omega$ による affine λ 計算の解釈

Move 列表現された戦略の合成

ゲーム A, B, C の 相互作用列 $u \in \text{Int}(A, B, C)$ とは列 $u \in (M_A + M_B + M_C)^*$ であって

- $u \upharpoonright_{A,B} \in \hat{P}_{A \multimap B}$
- $u \upharpoonright_{B,C} \in \hat{P}_{B \multimap C}$
- $u \upharpoonright_{A,C} \in \hat{P}_{A \multimap C}$

を満たすものを言う。ここで $u \upharpoonright_{A,B}$ は A, B への制限:

$$\varepsilon \upharpoonright_{A,B} = \varepsilon$$

$$(s \cdot a) \upharpoonright_{A,B} = \begin{cases} (s \upharpoonright_{A,B}) \cdot a & (\text{if } a \in M_A + M_B) \\ (s \upharpoonright_{A,B}) & (\text{if } a \in M_C) \end{cases}$$

Move 列表現された戦略の合成

$\hat{\sigma}$ を $A \multimap B$ の $\hat{\tau}$ を $B \multimap C$ の戦略の move列表現とする
 $\hat{\sigma}$ と $\hat{\tau}$ の合成 $\hat{\tau} \circ \hat{\sigma} \subseteq \hat{P}_{A \multimap C}^{even}$ を次で定義する

偶数長の列への制限

$$(\hat{\tau} \circ \hat{\sigma}) := \left\{ u \upharpoonright_{A,C} \mid \begin{array}{l} u \in Int(A, B, C) \\ u \upharpoonright_{A,B} \in \hat{\sigma} \\ u \upharpoonright_{B,C} \in \hat{\tau} \end{array} \right\}^{even}$$

例

$$\begin{array}{llllllllll} & c_1 & b_2 & b_3 & b_4 & & b_7 & b_8 & & \in \hat{\tau} \\ & & b_2 & b_3 & b_4 & a_5 & a_6 & b_7 & b_8 & a_9 & \in \hat{\sigma} \\ u & = & c_1 & b_2 & b_3 & b_4 & a_5 & a_6 & b_7 & b_8 & a_9 & \in Int(A, B, C) \\ u \upharpoonright_{A,C} & = & c_1 & & & & a_5 & a_6 & & & a_9 & \in (\hat{\tau} \circ \hat{\sigma}) \end{array}$$

2つの戦略の合成は等しい

命題 $\hat{\sigma}_{g \circ f} = (\hat{\sigma}_g \circ \hat{\sigma}_f)$

証明 (スケッチ)

$$\begin{array}{llllllllll} & c_1 & b_2 & b_3 & b_4 & & b_7 & b_8 & & \in \hat{\sigma}_g \\ & & b_2 & b_3 & b_4 & a_5 & a_6 & b_7 & b_8 & a_9 & \in \hat{\sigma}_f \\ u & = & c_1 & b_2 & b_3 & b_4 & a_5 & a_6 & b_7 & b_8 & a_9 & \in Int(A, B, C) \\ u|_{A,C} & = & c_1 & & & & a_5 & a_6 & & & a_9 & \in (\hat{\sigma}_g \circ \hat{\sigma}_f) \end{array}$$

のとき、

$$c_1 \xrightarrow{g} b_2 \xrightarrow{f} b_3 \xrightarrow{g} b_4 \xrightarrow{f} a_5 \quad a_6 \xrightarrow{f} b_7 \xrightarrow{g} b_8 \xrightarrow{f} a_9$$

ゆえに $(g \circ f)(c_1) = a_5$ かつ $(g \circ f)(a_6) = a_9$

逆も容易。

アウトライン

$\mathcal{G}_\omega$ に変数スコープの概念を導入する

1. 戦略の move 列表現

2. 変数スコープの機構を持つゲーム $\mathcal{G}_\omega$.

- アリーナ：依存関係のある move の集合
- 依存関係を守った move 列・戦略
- Innocent 戦略

3. $\mathcal{G}_\omega$ による affine λ 計算の解釈

アリーナ: 依存関係のある move の集合

定義 アリーナとは三つ組 $A = (M_A^O, M_A^P, \vdash_A)$ である。

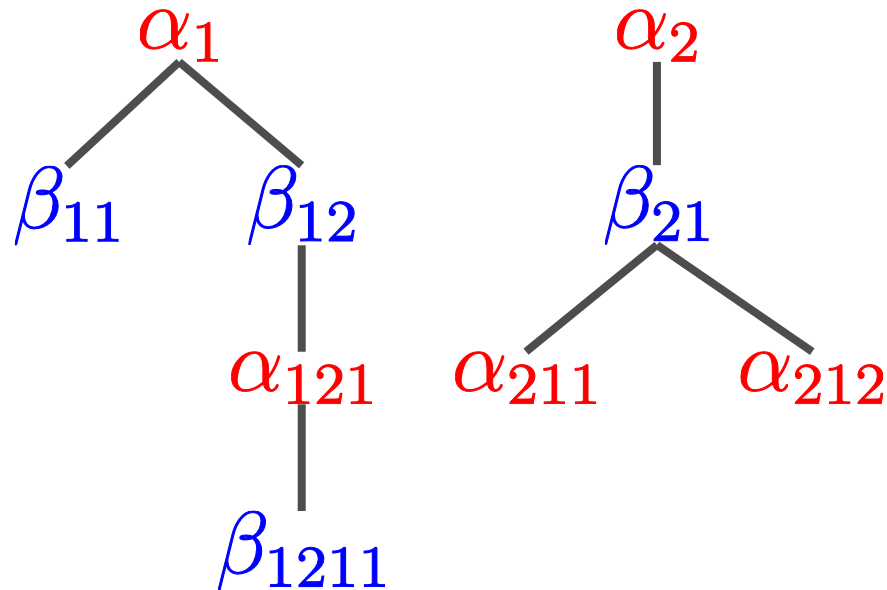
- M_A^O は O-move の集合
- M_A^P は P-move の集合
- $(\vdash_A) \subseteq (\{\star\} + M_A) \times M_A$ は **enabling relation** と呼ばれる関係で、以下を満たす

1. $\forall a \in M_A. \exists! x \in \{\star\} + M_A. x \vdash_A a$
2. $\star \vdash_A a$ ならば $a \in M_A^O$
3. $a \vdash_A a'$ かつ $a \in M_A^O$ ならば $a' \in M_A^P$
4. $a \vdash_A a'$ かつ $a \in M_A^P$ ならば $a' \in M_A^O$

$a \vdash_A a' \quad \Rightarrow \quad a$ の後でない a' は使えない

アリーナの森表現

例



$$A = (\{\alpha_1, \alpha_{121}, \alpha_2, \alpha_{211}, \alpha_{212}\}, \{\beta_{11}, \beta_{12}, \beta_{1211}, \beta_{21}\}, \vdash_A)$$

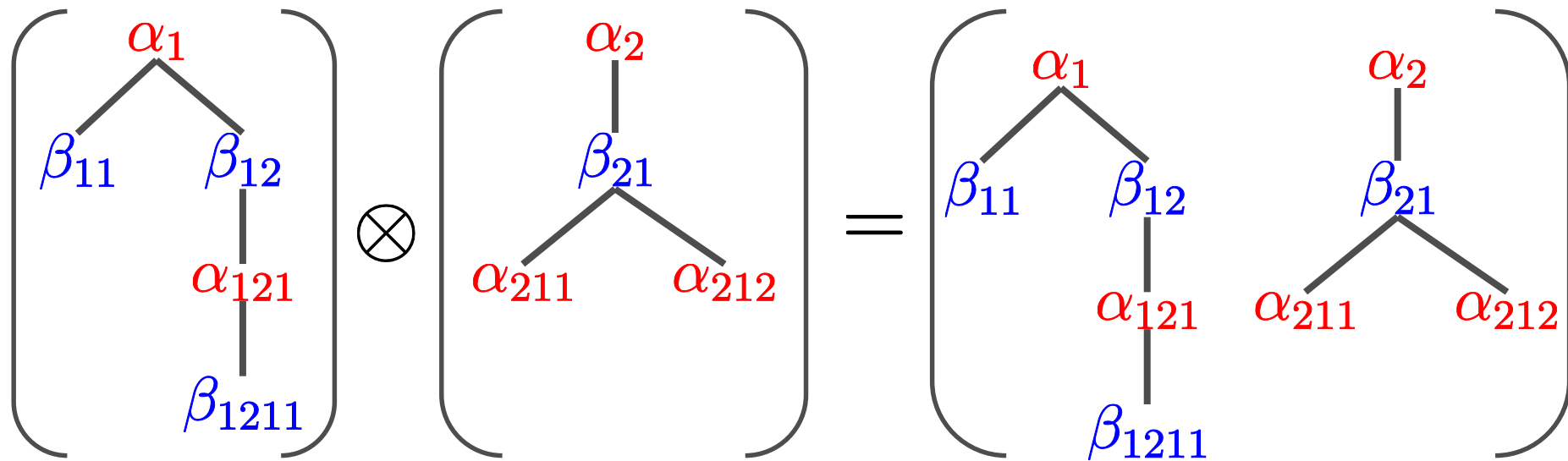
$$\begin{array}{lll} \star \vdash_A \alpha_1 & \star \vdash_A \alpha_2 & \alpha_1 \vdash_A \beta_{11} \\ \alpha_1 \vdash_A \beta_{12} & \alpha_2 \vdash_A \beta_{21} & \beta_{12} \vdash_A \alpha_{121} \\ \beta_{21} \vdash_A \alpha_{211} & \beta_{21} \vdash_A \alpha_{212} & \alpha_{121} \vdash_A \beta_{1211} \end{array}$$

組アリーナ

アリーナ A と B の組 $A \otimes B$ は次で定義される

$$A \otimes B = (M_A^O + M_B^O, M_A^P + M_B^P, (\vdash_A) \cup (\vdash_B))$$

例



関数アリーナ

アリーナ B が **pointed** $\stackrel{\text{def}}{\iff} \exists ! b \in M_B^O. \star \vdash_B b$

B は木

アリーナ A と pointed アリーナ B について、
関数アリーナ $A \multimap B$ を次で定義する

$$(A \multimap B) = (M_A^P + M_A^O, M_A^O + M_A^P, \vdash_{A \multimap B})$$

ただし $\vdash_{A \multimap B}$ は次の規則で定める

$$\frac{a \vdash_A a' \quad a \neq \star}{a \vdash_{A \multimap B} a'}$$

$$\frac{b \vdash_B b'}{b \vdash_{A \multimap B} b'}$$

$$\frac{\star \vdash_B b \quad \star \vdash_A a}{b \vdash_{A \multimap B} a}$$

関数アリーナ

アリーナ A と pointed アリーナ B について、
関数アリーナ $A \multimap B$ を次で定義する

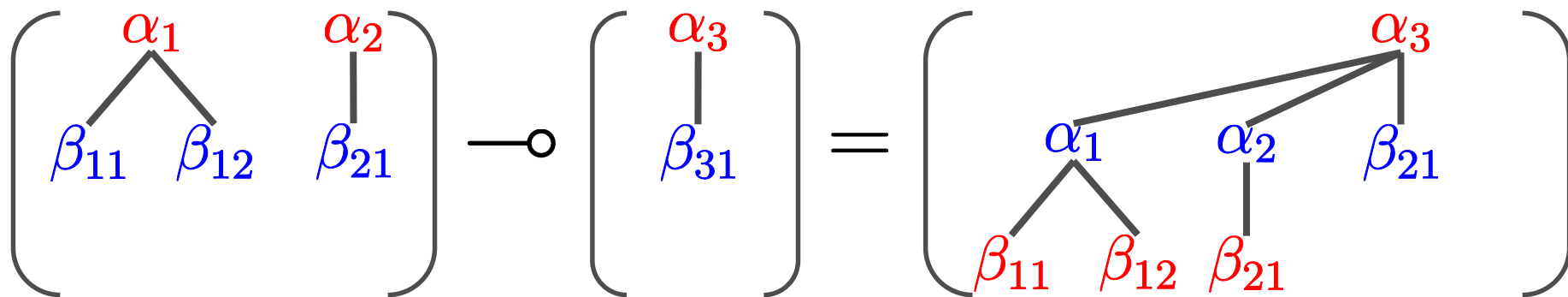
$$(A \multimap B) = (M_A^P + M_A^O, M_A^O + M_A^P, \vdash_{A \multimap B})$$

$$\frac{a \vdash_A a' \quad a \neq \star}{a \vdash_{A \multimap B} a'}$$

$$\frac{b \vdash_B b'}{b \vdash_{A \multimap B} b'}$$

$$\frac{\star \vdash_B b \quad \star \vdash_A a}{b \vdash_{A \multimap B} a}$$

例



ゲーム

定義 ($\mathcal{G}_\circ$ の) ゲームとはアリーナの組 (A, B) である

$$M_{A,B}^O := M_A^P + M_B^O \qquad M_{A,B}^P := M_A^O + M_B^P$$

$$(\vdash_{A,B}) := (\vdash_A) \cup (\vdash_B)$$

例

$$A = \left(\begin{array}{cc} \alpha_1 & \alpha_2 \\ \swarrow & \searrow & | \\ \beta_{11} & \beta_{12} & \beta_{21} \end{array} \right) \qquad B = \left(\begin{array}{c} \alpha_3 \\ | \\ \beta_{31} \end{array} \right) \quad \text{のとき}$$

$$(A, B) = \left(\begin{array}{cccc} \alpha_1 & \alpha_2 & & \alpha_3 \\ \swarrow & \searrow & | & | \\ \beta_{11} & \beta_{12} & \beta_{21} & \beta_{21} \end{array} \right)$$

型とアリーナの対応

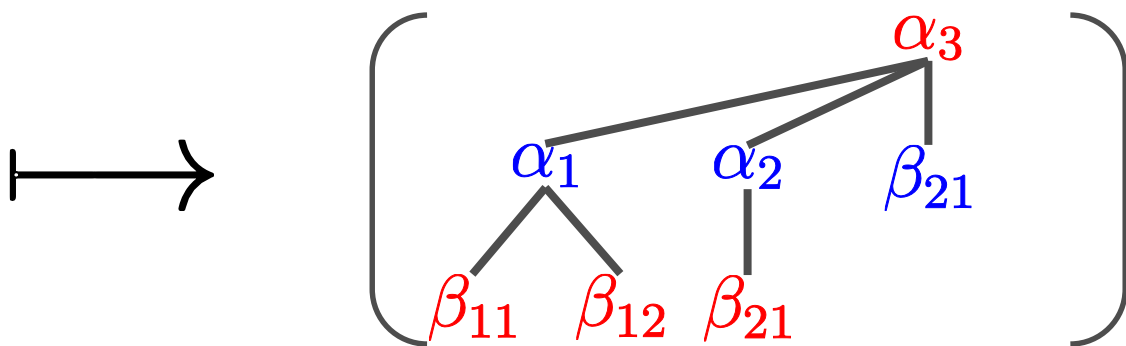
型と (有限の) pointed アリーナは、概ね一対一に対応

- ただし アリーナには引数の順序の情報がない

型環境と (有限の) アリーナの間にも、同じ対応がある

例

$$(\beta_{11} \multimap \beta_{12} \multimap \alpha_1) \multimap (\beta_{21} \multimap \alpha_2) \multimap \beta_{21} \multimap \alpha_3$$



型とアリーナの対応

型と (有限の) pointed アリーナは、概ね一対一に対応

- ただし アリーナには引数の順序の情報がない

型環境と (有限の) アリーナの間にも、同じ対応がある

例

$$(x : \beta_{11} \multimap \beta_{12} \multimap \alpha_1, y : \beta_{21} \multimap \alpha_2)$$

$$\vdash \rightarrow \left[\begin{array}{cc} \alpha_1 & \alpha_2 \\ \swarrow \quad \searrow & | \\ \beta_{11} \quad \beta_{12} & \beta_{21} \end{array} \right]$$

アウトライン

$\mathcal{G}_\omega$ に変数スコープの概念を導入する

1. 戦略の move 列表現

2. 変数スコープの機構を持つゲーム $\mathcal{G}_\omega$.

- アリーナ：依存関係のある move の集合
- 依存関係を守った move 列・戦略
- Innocent 戦略

3. $\mathcal{G}_\omega$ による affine λ 計算の解釈

Affine justified sequence

定義 ゲーム (A, B) の **affine justified sequence** とは affine 交代列 $a_1 \cdot a_2 \cdots a_n \in \hat{P}_{A,B}$ であって、

- $\forall i \leq n. (\star \vdash_{A,B} a_i) \vee (\exists j < i. a_j \vdash_A a_i)$
(justification)

であるものを言い、その集合を $J_{A,B}^{aff}$ と書く。

例

$$(A, B) = \left(\begin{array}{c} \gamma \\ \vdots \\ \beta \\ \vdots \\ \alpha \end{array} \quad \delta \right) \quad \text{において、}$$

$\delta \cdot \gamma \cdot \beta \cdot \alpha$ は affine justified sequence だが
 $\delta \cdot \alpha \cdot \beta \cdot \gamma$ は affine justified sequence でない

Justified な戦略

Move の依存関係を、相手が守るなら 自分も守る

定義

(A, B) を $\mathcal{G}_\perp$ のゲーム

f を $(M_{A,B}^O, M_{A,B}^P)$ の戦略

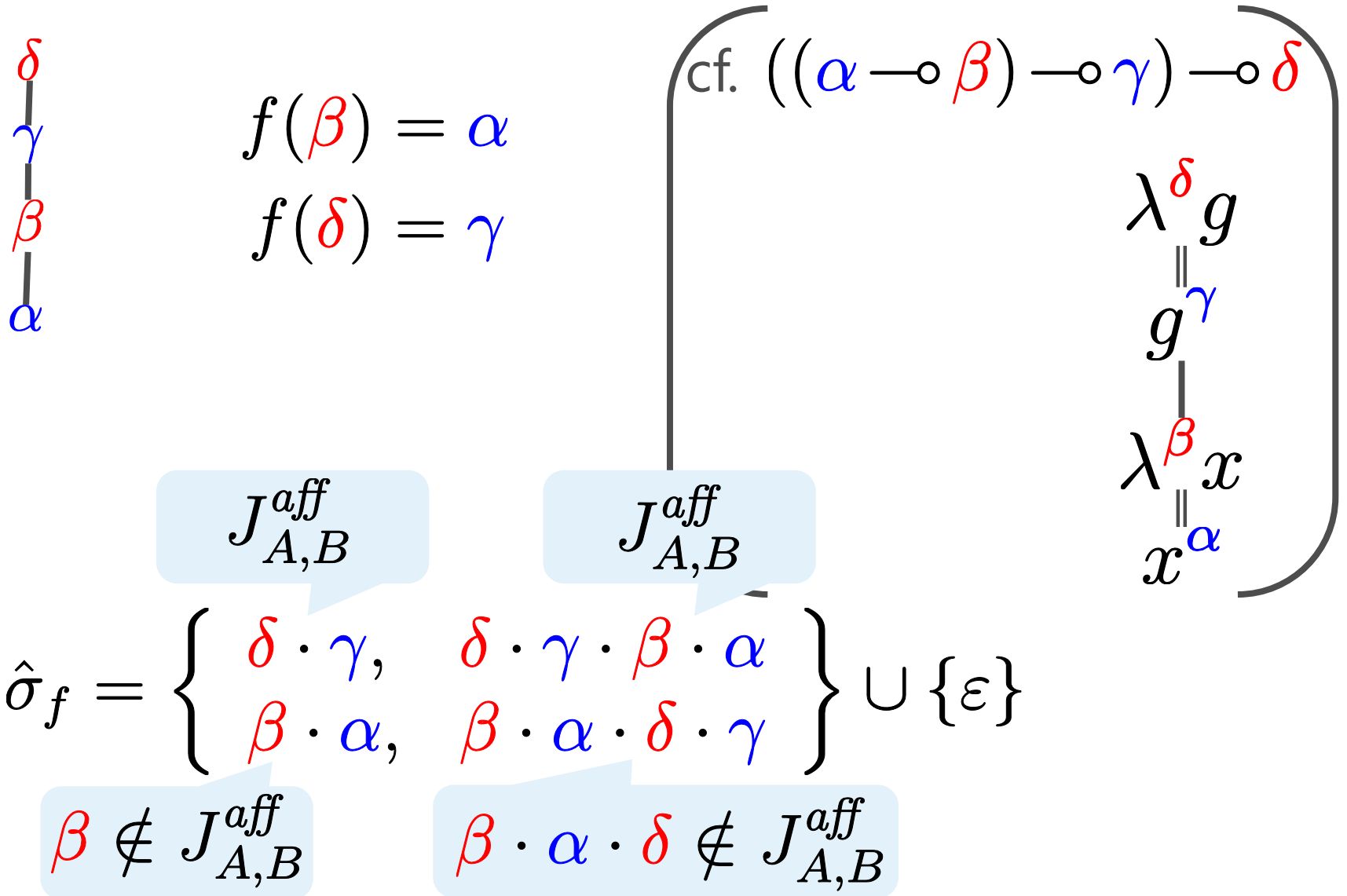
とする。 f が次を満たすとき、 **justified** であるという

$$\frac{s \in \hat{\sigma}_f \quad s \cdot \textcolor{red}{a} \in J_{A,B}^{aff}}{s \cdot \textcolor{red}{a} \cdot \textcolor{blue}{f(a)} \in J_{A,B}^{aff}}$$

このとき

$$s \cdot \textcolor{red}{a} \cdot \textcolor{blue}{f(a)} \in \hat{\sigma}_f$$

例：Justified な戦略



例：Justified でない戦略

δ
 γ
 β
 α

$$f(\beta) = \gamma$$

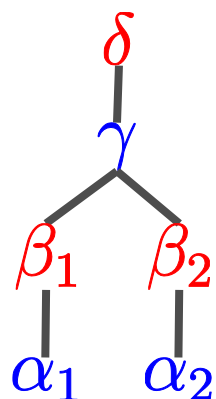
$$f(\delta) = \alpha$$

$$\left(\begin{array}{c} \text{cf. } ((\alpha \multimap \beta) \multimap \gamma) \multimap \delta \\ \lambda^{\delta} g \\ ?_{\alpha} \end{array} \right)$$

$$\delta \in J_{A,B}^{\text{aff}} \text{ and } \delta \cdot \alpha \notin J_{A,B}^{\text{aff}}$$

$$\hat{\sigma}_f = \left\{ \begin{array}{cc} \delta \cdot \alpha, & \delta \cdot \alpha \cdot \beta \cdot \gamma \\ \beta \cdot \gamma, & \beta \cdot \gamma \cdot \delta \cdot \alpha \end{array} \right\} \cup \{\varepsilon\}$$

例：Justified でない戦略



$$f(\beta_1) = \alpha_2$$

$$f(\beta_2) = \alpha_1$$

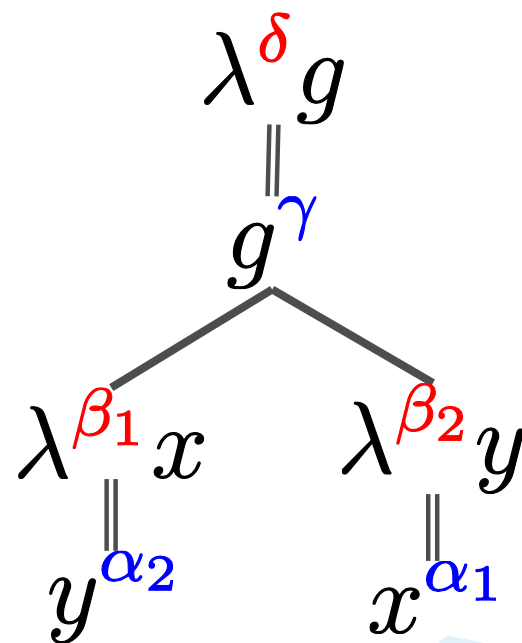
$$f(\delta) = \gamma$$

$$\delta \cdot \gamma \cdot \beta_1 \in J_{A,B}$$

$$\delta \cdot \gamma \cdot \beta_1 \cdot \alpha_2 \notin J_{A,B}$$

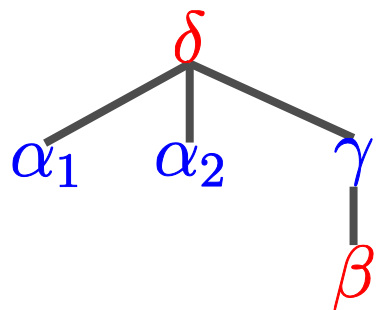
$$\hat{\sigma}_f = \left\{ \begin{array}{c} \delta \cdot \gamma, \quad \delta \cdot \gamma \cdot \beta_1 \cdot \alpha_2 \\ \vdots \end{array} \right\}$$

$$\text{cf. } ((\alpha_1 \multimap \beta_1) \multimap (\alpha_2 \multimap \beta_2) \multimap \gamma) \multimap \delta$$



スコープの外

例：Justified だが definable でない戦略



$$f(\beta) = \alpha_1$$

$$f(\delta) = \alpha_2$$

cf.

$$\alpha_1 \multimap \alpha_2 \multimap (\beta \multimap \gamma) \multimap \delta$$

$$\lambda^{\delta} x y g$$

$$y \stackrel{\alpha_2}{=}$$

依存関係を守ると
相手は β を使わない

$$\hat{\sigma}_f = \left\{ \begin{array}{ll} \delta \cdot \alpha_2, & \delta \cdot \alpha_2 \cdot \beta \cdot \alpha_1 \\ \beta \cdot \alpha_1, & \beta \cdot \alpha_1 \cdot \delta \cdot \alpha_2 \end{array} \right\} \cup \{\varepsilon\}$$

アウトライン

$\mathcal{G}_\omega$ に変数スコープの概念を導入する

1. 戦略の move 列表現

2. 変数スコープの機構を持つゲーム $\mathcal{G}_\omega$.

- アリーナ：依存関係のある move の集合
- 依存関係を守った move 列・戦略
- Innocent 戦略

3. $\mathcal{G}_\omega$ による affine λ 計算の解釈

Innocent 戦略

$\doteq$ justified 戦略の affine justified seq. による表示

定義 ゲーム (A, B) の (affine) innocent 戦略 σ とは
偶数長の affine justified seq. の集合 $\sigma \subseteq (J_{A,B}^{aff})^{even}$ で
以下を満たすものを言う

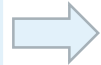
$$\begin{array}{c}
 \frac{}{\varepsilon \in \sigma} \quad \frac{s \cdot a \cdot b \in \sigma}{s \in \sigma} \quad \frac{s \cdot t \cdot u \in J_{A,B}^{aff} \quad s \cdot t \in \sigma \quad s \cdot u \in \sigma}{s \cdot t \cdot u \in \sigma} \\
 \\
 \frac{s \cdot a \cdot b \in \sigma \quad s \cdot a \cdot b' \in \sigma}{b = b'} \quad \frac{s \cdot a \cdot b \cdot a' \cdot b' \cdot t \in \sigma \quad b \not\vdash_{A,B} a'}{s \cdot a' \cdot b' \cdot a \cdot b \cdot t \in \sigma}
 \end{array}$$

命題 $\sigma: \text{innocent} \iff \exists f: \text{justified}. \sigma = \hat{\sigma}_f \cap J_{A,B}^{aff}$

Justified 戦略と innocent 戦略

Innocent 戦略の方が少ない。次の合成は恒等ではない

Justified 戦略



Innocent 戦略



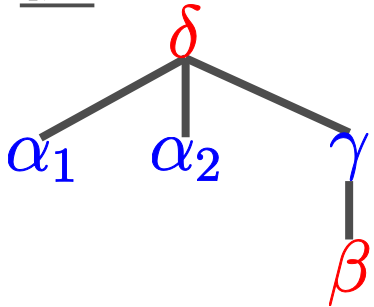
Justified 戦略

$$(f : M_{A,B}^O \xrightarrow{inj} M_{A,B}^P) \longmapsto (\hat{\sigma}_f \cap J_{A,B}^{aff})$$

$$\sigma \longmapsto (f_\sigma : M_{A,B}^O \xrightarrow{inj} M_{A,B}^P)$$

$$f_\sigma(a) = b \stackrel{\text{def}}{\iff} \exists s. s \cdot a \cdot b \in \sigma$$

例



$$\left(\begin{array}{l} f(\beta) = \alpha_1 \\ f(\delta) = \alpha_2 \end{array} \right) \longmapsto \left\{ \begin{array}{l} \varepsilon \\ \delta \cdot \alpha_2 \end{array} \right\} \longmapsto \left(\begin{array}{l} f'(\beta) \uparrow \\ f'(\delta) = \alpha_2 \end{array} \right)$$

P-view

Justified sequence の中でも、特に大切なもの

定義 **P-view** とは affine justified sequence $p \in J_{A,B}^{aff}$ であって、次の条件を満たすもの：

$$p = a_1 \cdot b_1 \cdot a_2 \cdot b_2 \cdot \dots \cdot a_n \cdot b_n \text{ とすると}$$

$$\forall i < n. b_i \vdash_{A,B} a_{i+1}$$

ゲーム (A, B) の P-view の集合を $V_{A,B}^{aff}$ と書く。

命題 Innocent 戦略 σ は $\sigma \cap V_{A,B}^{aff}$ で完全に決定される

アウトライン

$\mathcal{G}_\omega$ に変数スコープの概念を導入する

1. 戦略の move 列表現
2. 変数スコープの機構を持つゲーム $\mathcal{G}_\omega$.
3. $\mathcal{G}_\omega$ による affine λ 計算の解釈

$\mathcal{G}_\circ$ による affine λ 項の解釈

$\mathcal{G}_\circ$ による解釈と同じ

- $\mathcal{G}_\circ$ による解釈の定義中の戦略は innocent
- Innocent 戦略の合成は innocent

$$\left[\frac{}{x : T \vdash x : T} \right] = \text{id}_{\llbracket T \rrbracket}$$

$$\left[\frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x.M : T \multimap U} \right] = \Lambda(\llbracket \Gamma, x : T \vdash M : U \rrbracket)$$

$$\begin{aligned} \left[\frac{\Gamma \vdash M : T \multimap U \quad \Delta \vdash N : T}{\Gamma, \Delta \vdash M N : U} \right] \\ = \mathbf{ev} \circ (\llbracket \Gamma \vdash M : T \multimap U \rrbracket \otimes \llbracket \Delta \vdash N : U \rrbracket) \end{aligned}$$

$\mathcal{G}_\circ$ による解釈の性質

定理 (モデルの妥当性)

$$M =_{\beta\eta} N \text{ ならば } \llbracket M \rrbracket = \llbracket N \rrbracket$$

定理 (faithfulness)

$$\llbracket M \rrbracket = \llbracket N \rrbracket \text{ ならば } M =_{\beta\eta} N$$

定理 (full definability)

$(\llbracket \Gamma \rrbracket, \llbracket T \rrbracket)$ の任意の innocent 戦略 σ に対して、
affine λ 項 $\Gamma \vdash M : T$ が存在して、

$$\sigma = \llbracket \Gamma \vdash M : T \rrbracket$$

Affine λ 計算の近似

対象言語

ゲーム

線形資源計算

Affine λ {

$\mathcal{G}_{\rightarrow}^-$

$\mathcal{G}_{\rightarrow}$

$\cong$

$\partial_0 \lambda_{\rightarrow}$

$\lambda_{\rightarrow}$

$\mathcal{G}$

$\cong$

$\partial_0 \lambda_{\rightarrow}$

PCF

$\mathcal{G}^{\text{QA}}$

$\cong$

$\partial_0 \text{PCF}$

アウトライン

プレイの構文的表現

1. 線形資源 λ 計算
2. Affine λ 項の近似解釈
3. 近似とプレイの対応

アウトライン

プレイの構文的表現

1. 線形資源 λ 計算
2. Affine λ 項の近似解釈
3. 近似とプレイの対応

線形資源 λ 計算 $\partial_0 \lambda$ 。 (cf. [Ehrhard & Regnier 08])

関数適用の引数が使用されるか否かが厳格な計算

- Affine λ 計算では $f\ x$ の x は高々一度使われる
- 線形資源 λ 計算では、関数適用は二種類
 - $f\ [x]$ の x はちょうど一度使われる
 - $f\ []$ では引数は使われない

構文

$$r ::= x \mid \lambda x. r \mid r\ \mu$$

$$\mu ::= [] \mid [r]$$

型規則

$$\frac{}{x : T \vdash x : T} \quad \frac{\Gamma, x : T \vdash r : U}{\Gamma \vdash \lambda x.r : T \multimap U} \quad \frac{\Gamma \vdash r : T \multimap U \quad \Delta \vdash \mu : T}{\Gamma, \Delta \vdash r \mu : U}$$

$$\frac{}{\vdash [] : T} \quad \frac{\Gamma \vdash r : T}{\Gamma \vdash [r] : T}$$

※ 型環境の weakening と permutation は省略

構文

$$r ::= x \mid \lambda x.r \mid r \mu$$

$$\mu ::= [] \mid [r]$$

簡約

β 簡約

- 資源 (= 引数) が期待通り使われない場合は stuck

$$(\lambda x.r)[] \longrightarrow r \quad (\text{if } x \notin \text{fv}(r))$$

$$(\lambda x.r)[] \not\longrightarrow \quad (\text{if } x \in \text{fv}(r))$$

$$(\lambda x.r)[r'] \longrightarrow r[r'/x] \quad (\text{if } x \in \text{fv}(r))$$

$$(\lambda x.r)[r'] \not\longrightarrow \quad (\text{if } x \notin \text{fv}(r))$$

η 展開

$$r \longrightarrow \lambda x.r [x]$$

$$r \longrightarrow \lambda x.r []$$

標準形

λ 項の $\beta\eta$ 標準形に対応

$$\check{r} ::= \lambda x_1 x_2 \dots x_n. y \check{\mu}_1 \dots \check{\mu}_k$$

$$\left(\begin{array}{l} n \geq 0 \\ k \geq 0 \\ y \check{\mu}_1 \dots \check{\mu}_k : o \end{array} \right)$$

$$\check{\mu} ::= [] \mid [\check{r}]$$

正規化

線形資源 λ 項の集合を、その標準形の集合にする

- NF を標準形の線形資源 λ 項の集合とする

$$\text{nf}(X) := \{ r' \in NF \mid \exists r \in X. r \longrightarrow^* r' \}$$

$$\left(\text{nf}(X) := \{ \mu' \in NF \mid \exists \mu \in X. \mu \longrightarrow^* \mu' \} \right)$$

アウトライン

プレイの構文的表現

1. 線形資源 λ 計算
2. Affine λ 項の近似解釈
3. 近似とプレイの対応

Affine λ 項の 線形資源項 による近似

Affine λ 項を 線形資源計算の項の集合とみなす

$$\langle x \rangle := \{x\}$$

$$\langle \lambda x.M \rangle := \{ \lambda x.r \mid r \in \langle M \rangle \}$$

$$\langle M N \rangle := \{ r \mu \mid r \in \langle M \rangle, \mu \in \langle\langle N \rangle\rangle \}$$

$$\langle \Omega \rangle := \emptyset$$

$$\langle\langle M \rangle\rangle := \{ [] \} \cup \{ [r] \mid r \in \langle M \rangle \}$$

ルールによる近似の定義

$$\langle M \rangle = \{ r \mid r \sqsubseteq M \}$$

$$\overline{x \sqsubseteq x}$$

$$\frac{r \sqsubseteq M}{\lambda x. r \sqsubseteq \lambda x. M}$$

$$\frac{r \sqsubseteq M \quad r' \sqsubseteq N}{r[r'] \sqsubseteq M N}$$

$$\frac{r \sqsubseteq M}{r [] \sqsubseteq M N}$$

近似の例

$$T = \alpha_1 \multimap \alpha_2 \multimap (\beta_1 \multimap \beta_2 \multimap \gamma) \multimap \delta$$

$$M = \lambda x y g . g \ x \ y$$

$$\langle\langle M \rangle\rangle = \left\{ \begin{array}{l} [] \\ [\lambda x y g . g \ [] \ []] \\ [\lambda x y g . g \ [x] \ []] \\ [\lambda x y g . g \ [] \ [y]] \\ [\lambda x y g . g \ [x] \ [y]] \end{array} \right\}$$

近似の妥当性

定理 以下は同値。

1. $M =_{\beta\eta} N$
2. $\text{nf}(\langle M \rangle) = \text{nf}(\langle N \rangle)$
3. $\text{nf}(\langle\langle M \rangle\rangle) = \text{nf}(\langle\langle N \rangle\rangle)$

アウトライン

プレイの構文的表現

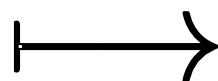
1. 線形資源 λ 計算
2. Affine λ 項の近似解釈
3. 近似とプレイの対応

標準形の資源λ項と justified seq. との対応

$$T = \alpha_1 \multimap \alpha_2 \multimap (\beta_1 \multimap \beta_2 \multimap \gamma) \multimap \delta$$

$$[\lambda x y g . g \ [\] \ [y]] \in \langle\langle \lambda x y g . g \ x \ y \rangle\rangle$$

$$\begin{array}{c} \lambda^{\delta} x y g \\ \parallel \\ g^{\gamma} \\ | \\ \lambda^{\beta_2} \\ \parallel \\ y^{\alpha_2} \end{array}$$

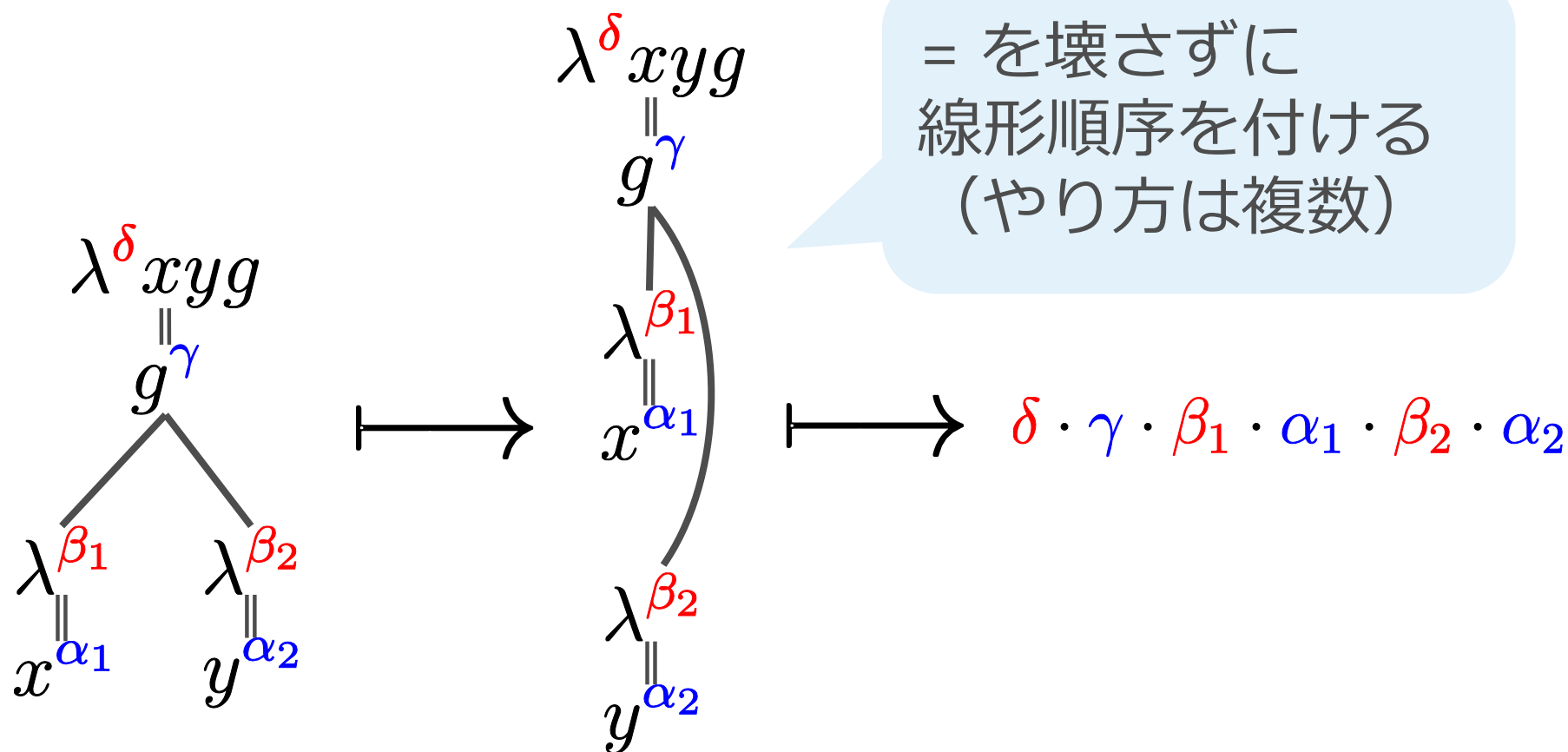


$$\delta \cdot \gamma \cdot \beta_2 \cdot \alpha_2$$

標準形の資源λ項と justified seq. との対応

$$T = \alpha_1 \multimap \alpha_2 \multimap (\beta_1 \multimap \beta_2 \multimap \gamma) \multimap \delta$$

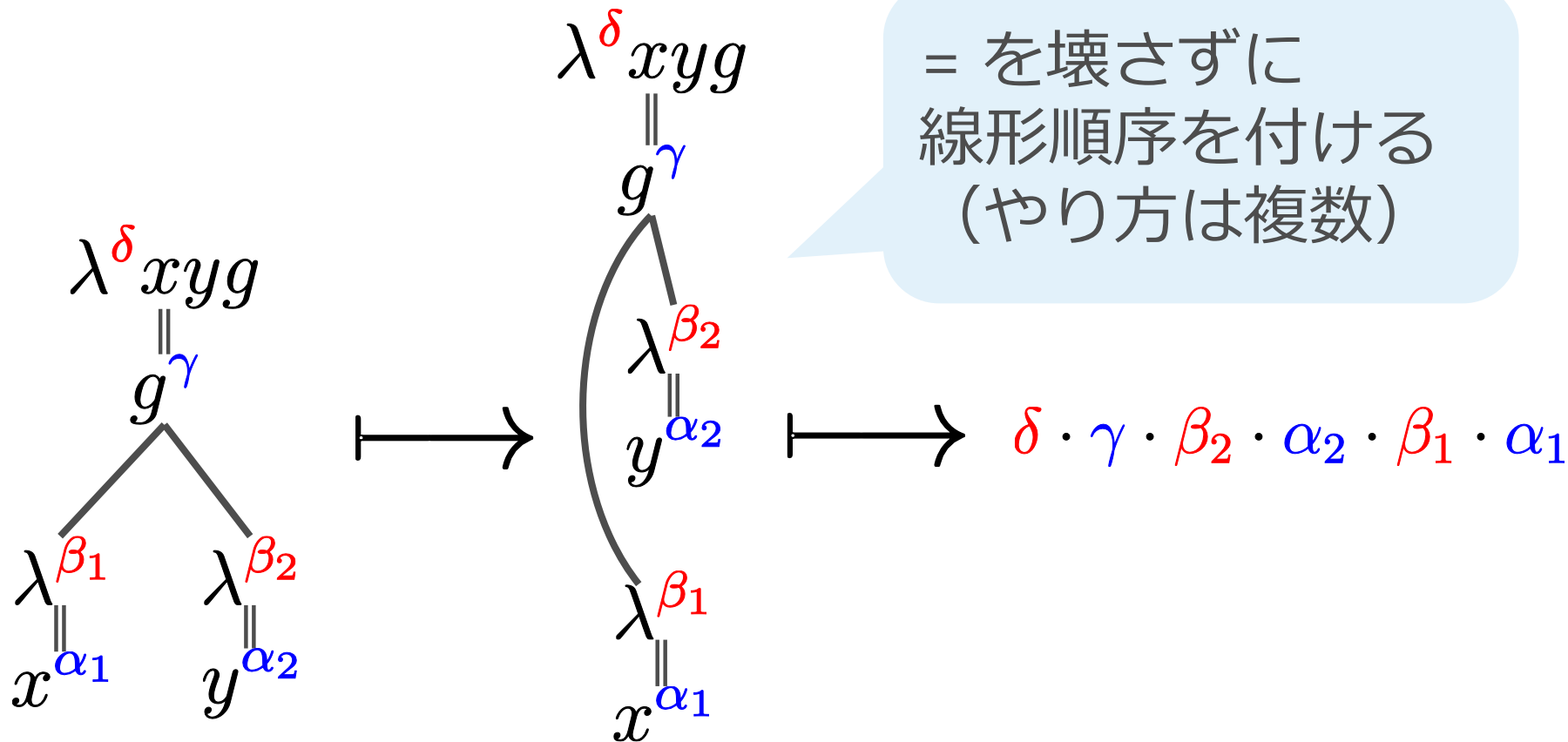
$$[\lambda x y g . g [x] [y]] \in \langle\langle \lambda x y g . g x y \rangle\rangle$$



標準形の資源λ項と justified seq. との対応

$$T = \alpha_1 \multimap \alpha_2 \multimap (\beta_1 \multimap \beta_2 \multimap \gamma) \multimap \delta$$

$$[\lambda x y g . g [x] [y]] \in \langle\langle \lambda x y g . g x y \rangle\rangle$$



線形近似とゲーム解釈の一致

$$T = \alpha_1 \multimap \alpha_2 \multimap (\beta_1 \multimap \beta_2 \multimap \gamma) \multimap \delta$$

$$M = \lambda x y g . g \ x \ y$$

$$\llbracket M \rrbracket$$

$$\langle\langle M \rangle\rangle$$

ε

$$\delta \cdot \gamma$$

$$\delta \cdot \gamma \cdot \beta_1 \cdot \alpha_1$$

$$\delta \cdot \gamma \cdot \beta_2 \cdot \alpha_2$$

$$\left. \begin{array}{l} \delta \cdot \gamma \cdot \beta_2 \cdot \alpha_2 \cdot \beta_1 \cdot \alpha_1 \\ \delta \cdot \gamma \cdot \beta_1 \cdot \alpha_1 \cdot \beta_2 \cdot \alpha_2 \end{array} \right\}$$

$[]$

$$[\lambda x y g . g \ [] \ []]$$

$$[\lambda x y g . g \ [x] \ []]$$

$$[\lambda x y g . g \ [] \ [y]]$$

$$[\lambda x y g . g \ [x] \ [y]]$$

線形近似とゲーム解釈の一致

$$\Phi_{\Gamma, T} : \{ \mu \mid \Gamma \vdash \mu : T \} \rightarrow J_{[[\Gamma]], [[T]]}^{aff}$$

を先ほどの対応関係を表す関数とする。

定理 $[[M]] = \bigcup \{ \Phi(\mu) \mid \mu \in \text{nf}(\langle\langle M \rangle\rangle) \}$

ゲーム意味論 と 線形近似 は本質的には同じ

λ 計算のゲーム意味論

対象言語

ゲーム

線形資源計算

Affine λ	{	$\mathcal{G}_{\rightarrow}$		
		$\mathcal{G}_{\rightarrow}$	$\cong$	$\partial_0 \lambda_{\rightarrow}$
$\lambda_{\rightarrow}$		$\mathcal{G}$	$\cong$	$\partial_0 \lambda_{\rightarrow}$
PCF		$\mathcal{G}^{\text{QA}}$	$\cong$	$\partial_0 \text{PCF}$

アウトライン

Affineness の制約のないゲーム $\mathcal{G}$ の導入

各 move は高々一度しか使えない

1. 複製がもたらす困難 と 解決策
2. 複製のあるゲーム $\mathcal{G}$
3. $\mathcal{G}$ による λ 計算の解釈

アウトライン

Affineness の制約のないゲーム $\mathcal{G}$ の導入

各 move は高々一度しか使えない

1. 複製がもたらす困難 と 解決策
2. 複製のあるゲーム $\mathcal{G}$
3. $\mathcal{G}$ による λ 計算の解釈

複製

ひとつの変数 (または move) を複数回使うこと

- 複製なし (affine)

$$\lambda g.g (\lambda x.x)$$

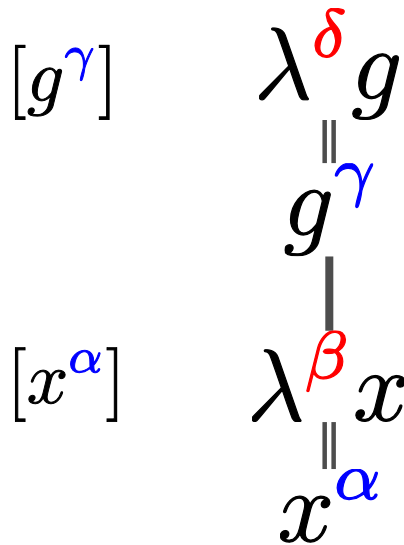
- 複製有り

$$\lambda g.g (\lambda x.g (\lambda y.x))$$

Move の意味 と 複製のもたらす困難

例（複製なし） Move に対してノードは (高々) ひとつ

$$\lambda g.g (\lambda x.x) : ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$$



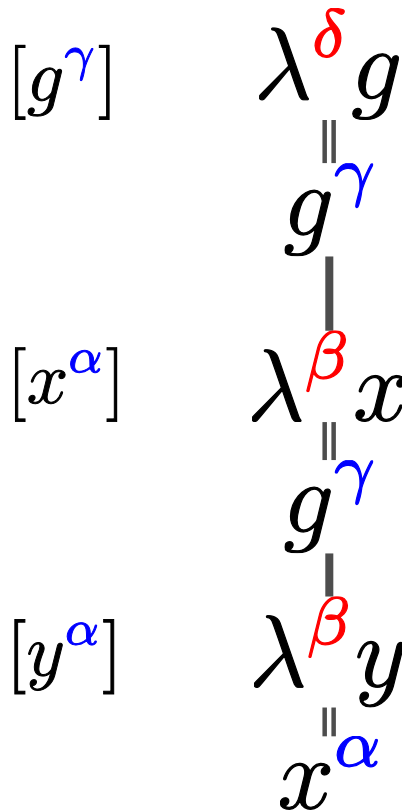
O-move β
= Opponent からの質問
「 g^γ の引数は？」

P-move α
= Proponent の回答
「 $\lambda^\beta x$ の x を使う」

Move の意味 と 複製のもたらす困難

例（複製あり） Move だけではノードを特定できない

$$\lambda g. g (\lambda x. g (\lambda y. x)) : ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$$



O-move β

= Opponent からの質問

「 g^γ の引数は？」

g^γ ってどっちの？

P-move α

= Proponent の回答

「 $\lambda^\beta x$ の x を使う」

$\lambda^\beta y$ の y ではなくて？

解決策 : Justification pointer

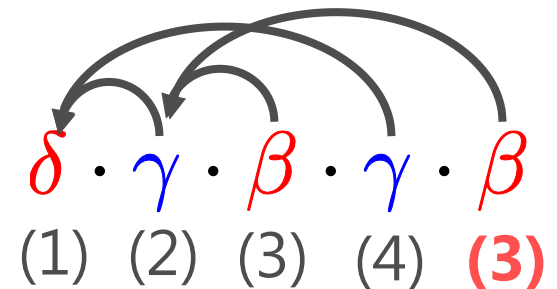
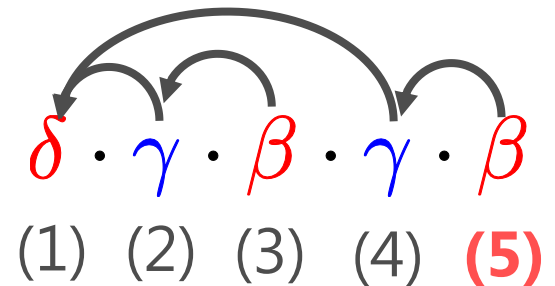
どの move が親なのかを陽に示す

例 $\lambda g. g (\lambda x. g (\lambda y. x)) : ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$

(1) $\lambda^{\delta} g$
 $\parallel$
(2) g^{γ}
 |
(3) $\lambda^{\beta} x$
 $\parallel$
(4) g^{γ}
 |
(5) $\lambda^{\beta} y$
 $\parallel$
(6) x^{α}

$\delta \cdot \gamma \cdot \beta \cdot \gamma \cdot \beta$

どっちの g^{γ} の
引数か不明

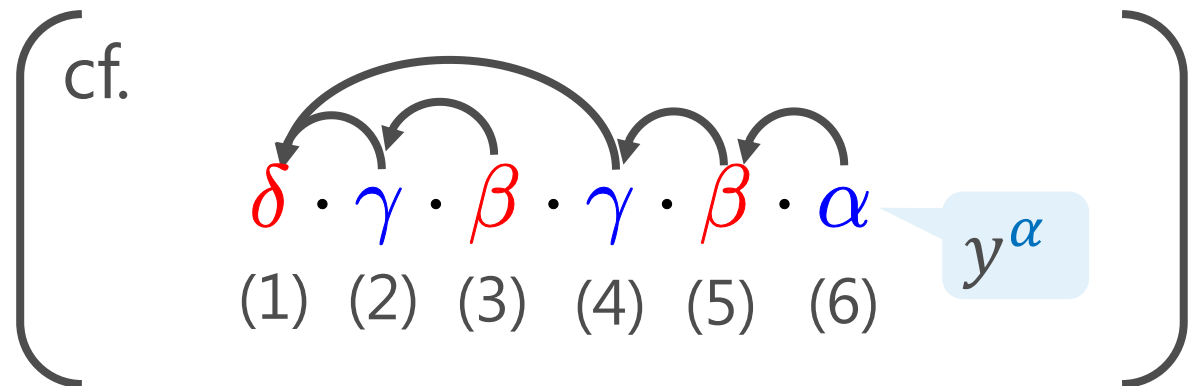
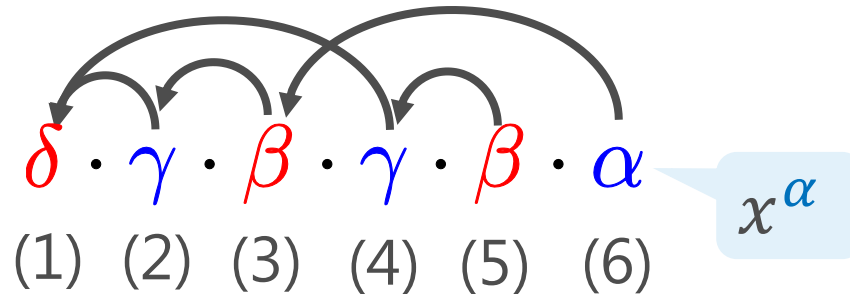


解決策 : Justification pointer

どの move が親なのかを陽に示す

例 $\lambda g.g (\lambda x.g (\lambda y.x)) : ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$

(1) $\lambda^{\delta} g$
(2) g^{γ}
(3) $\lambda^{\beta} x$
(4) g^{γ}
(5) $\lambda^{\beta} y$
(6) x^{α}



アウトライン

Affineness の制約のないゲーム $\mathcal{G}$ の導入

各 move は高々一度しか使えない

1. 複製がもたらす困難 と 解決策
2. 複製のあるゲーム $\mathcal{G}$
3. $\mathcal{G}$ による λ 計算の解釈

アリーナ と ゲーム

$\mathcal{G}_\perp$ のものと同じ

- アリーナは三つ組 $A = (M_A^O, M_A^P, \vdash_A)$
- ゲームはアリーナの組 (A, B)

Justified sequence

どの move に justify されているかを陽に記述

定義 ゲーム (A, B) の **justified sequence** とは

Move の列 $s : \{1, 2, \dots, n\} \rightarrow M_{A,B}$

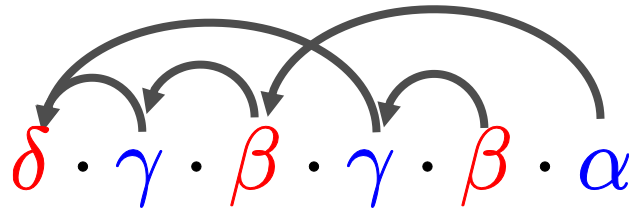
ポインタ $\rho_s : \{1, 2, \dots, n\} \rightarrow \{0, 1, 2, \dots, n\}$

の組であって、以下を満たすもの：

1. $s(2i + 1) \in M_{A,B}^O$ かつ $s(2i) \in M_{A,B}^P$
2. $\rho_s(i) < i$
3. $s(\rho_s(i)) \vdash_{A,B} s(i)$ ($s(0) = \star$ と解釈する)

Justified sequence

例



条件の再掲

1. O-move と P-move が交互に並ぶ
2. ポインタは必ず左向き
3. ポイントするなら enabling relation を満たす

$$\dots a \dots b \dots \quad \text{ならば } a \vdash b$$

戦略・相互作用列・合成

形式的には $\mathcal{G}_\circ$ のものと (ほぼ) 同じ

違い

- Affine ではない。
列に同じ move 複数回起こりえる。
- 列にはポインタが付随している。
 - 列の等しさは、ポインタも含めての等しさ
 - 列の制限は、ポインタも含めての制限

Innocent 戦略

定義 ゲーム (A, B) の innocent 戦略 σ とは justified seq. の部分集合 $\sigma \subseteq J_{A,B}$ で次を満たすもの

$$\frac{s \cdot a \cdot b \in \sigma \quad s \cdot a \cdot b' \in \sigma}{s \cdot a \cdot b = s \cdot a \cdot b'}$$

$$\frac{}{\varepsilon \in \sigma} \quad \frac{s \cdot a \cdot b \in \sigma}{s \in \sigma} \quad \frac{s \cdot a \cdot b \cdot a' \cdot b' \cdot t \in \sigma}{s \cdot a' \cdot b' \cdot a \cdot b \cdot t \in \sigma}$$


$$\frac{s \cdot t \cdot u \in J_{A,B} \quad s \cdot t \in \sigma \quad s \cdot u \in \sigma}{s \cdot t \cdot u \in \sigma}$$

P-view

定義 P-view とは justified sequence $p \in J_{A,B}$ で次の条件を満たすもの：

$$p = a_1 \cdot b_2 \cdot a_3 \cdot b_4 \cdot \cdots \cdot a_{2n-1} \cdot b_{2n} \text{ とすると}$$

$$\forall i < n. \rho_p(2i+1) = i$$

$$\left[p = \cdots b_{2i} \cdot a_{2i+1} \cdots \right]$$

ゲーム (A, B) の P-view の集合を $V_{A,B}$ と書く。

命題 Innocent 戦略 σ は $\sigma \cap V_{A,B}$ で完全に決定される

相互作用列

定義 ゲーム A, B, C の 相互作用列 $u \in \text{Int}(A, B, C)$ は

列 $u: \{1, 2, \dots, n\} \rightarrow (M_A + M_B + M_C)$

ポインタ $\rho_u: \{1, 2, \dots, n\} \rightarrow \{0, 1, 2, \dots, n\}$

の組であって、以下を満たすもの

$$u|_{A,B} \in J_{A,B} \quad u|_{B,C} \in J_{B,C} \quad u|_{A,C} \in J_{A,C}$$

ここで $u|_{A,B}$ は A, B への制限

注：暗黙にだがポインタも考えている

戦略の合成

σ を (A, B) の、 τ を (B, C) の innocent 戦略とする
 σ と τ の合成 $(\tau \circ \sigma) \subseteq J_{A,C}$ を次で定義する

偶数長の列への制限

$$(\tau \circ \sigma) := \left\{ u \upharpoonright_{A,C} \mid \begin{array}{l} u \in \text{Int}(A, B, C) \\ u \upharpoonright_{A,B} \in \sigma \\ u \upharpoonright_{B,C} \in \tau \end{array} \right\}^{\text{even}}$$

アウトライン

Affineness の制約のないゲーム $\mathcal{G}$ の導入

各 move は高々一度しか使えない

1. 複製がもたらす困難 と 解決策
2. 複製のあるゲーム $\mathcal{G}$
3. $\mathcal{G}$ による λ 計算の解釈

λ計算の解釈

$\mathcal{G}_\rightarrow$ のものと (ほぼ) 同じ

- 型の解釈は pointed アリーナ
- 型環境の解釈はアリーナ
- 項 $\Gamma \vdash M : T$ の解釈は ゲーム $(\llbracket \Gamma \rrbracket, \llbracket T \rrbracket)$ の戦略

$$\left[\frac{}{x : T \vdash x : T} \right] = \text{id}_{\llbracket T \rrbracket}$$

$$\left[\frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x. M : T \rightarrow U} \right] = \Lambda(\llbracket \Gamma, x : T \vdash M : U \rrbracket)$$

$$\left[\frac{\Gamma \vdash M : T \rightarrow U \quad \Gamma \vdash N : T}{\Gamma, \Delta \vdash M N : U} \right] = \text{ev} \circ \langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle$$

Λ

$(A \times B, C)$ の戦略から $(A, B \rightarrow C)$ の戦略への変換

定義 $\varphi: J_{A, B \rightarrow C} \rightarrow J_{A \times B, C}$ を次で定義する

- Move の列については恒等変換

$$\begin{aligned} M_{A \times B, C}^O &= M_A^P + M_B^P + M_C^O = M_{A, B \rightarrow C} \\ M_{A \times B, C}^P &= M_A^O + M_B^O + M_C^P = M_{A, B \rightarrow C} \end{aligned}$$

- ポインタも ほぼ恒等変換

- ただし $\star \vdash_B s(i)$ のとき、 $\rho_{\varphi(s)}(i) = 0$

$$(A, B \rightarrow C) = \left(\begin{array}{c} \alpha \\ \triangle A \end{array} \quad \begin{array}{c} \beta \\ \triangle B \end{array} \quad \begin{array}{c} \gamma \\ \triangle C \end{array} \right) \quad (A \times B, C) = \left(\begin{array}{c} \alpha \\ \triangle A \end{array} \quad \begin{array}{c} \beta \\ \triangle B \end{array} \quad \begin{array}{c} \gamma \\ \triangle C \end{array} \right)$$

Λ

$(A \times B, C)$ の戦略から $(A, B \rightarrow C)$ の戦略への変換

定義 $\varphi: J_{A, B \rightarrow C} \rightarrow J_{A \times B, C}$ を次で定義する

- Move の列については恒等変換

$$\begin{aligned} M_{A \times B, C}^O &= M_A^P + M_B^P + M_C^O = M_{A, B \rightarrow C} \\ M_{A \times B, C}^P &= M_A^O + M_B^O + M_C^P = M_{A, B \rightarrow C} \end{aligned}$$

$(A, B \rightarrow C)$ の戦略 σ は恒等変換

た $(A \times B, C)$ の戦略 $\varphi(s)(i) = 0$

補題 $\sigma \mapsto \{ \varphi(s) \mid s \in \sigma \}$ は innocent 戦略の全単射

定義 Λ は上の変換の逆変換

$\langle \sigma, \tau \rangle$

σ : (A, B) の innocent 戦略

τ : (A, C) の innocent 戦略

$\langle \sigma, \tau \rangle$ は次で定義される $(A, B \times C)$ の innocent 戦略

$$\langle \sigma, \tau \rangle := \{ s \mid t \in \sigma, u \in \tau, s \in t \# u \}$$

ここで $t \# u$ は ふたつの列のインターリーブ

$$\varepsilon \# \varepsilon := \{ \varepsilon \}$$

$$\begin{aligned} t \# u := & \{ a \cdot b \cdot s \mid t = a \cdot b \cdot t', s \in t' \# u \} \\ & \cup \{ a \cdot b \cdot s \mid u = a \cdot b \cdot u', s \in t \# u' \} \end{aligned}$$

λ計算の解釈

$\mathcal{G}_\circ$ のものと (ほぼ) 同じ

- 型の解釈は pointed アリーナ
- 型環境の解釈はアリーナ
- 項 $\Gamma \vdash M : T$ の解釈は、 Γ の型環境の戦略 $\llbracket \Gamma \rrbracket$ と M の戦略 $\llbracket M \rrbracket$ の積を通して定義

$$\left\llbracket \frac{}{x : T \vdash x : T} \right\rrbracket = \text{id}_{\llbracket T \rrbracket}$$

$$\left\llbracket \frac{\Gamma, x : T \vdash M : U}{\Gamma \vdash \lambda x. M : T \rightarrow U} \right\rrbracket = \Lambda(\llbracket \Gamma, x : T \vdash M : U \rrbracket)$$

$$\left\llbracket \frac{}{\Gamma, \Delta \vdash M N : U} \right\rrbracket = \text{ev} \circ \langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle$$

$\mathcal{G}$ による解釈の性質

定理 (モデルの妥当性)

$$M =_{\beta\eta} N \text{ ならば } \llbracket M \rrbracket = \llbracket N \rrbracket$$

定理 (faithfulness)

$$\llbracket M \rrbracket = \llbracket N \rrbracket \text{ ならば } M =_{\beta\eta} N$$

定理 (compact definability)

$(\llbracket \Gamma \rrbracket, \llbracket T \rrbracket)$ の任意の innocent 戦略 σ について、
 σ が compact なら、 λ 項 $\Gamma \vdash M : T$ が存在して、

P-view が有限個 $\sigma = \llbracket \Gamma \vdash M : T \rrbracket$

λ 計算の線形近似

対象言語

ゲーム

線形資源計算

Affine λ

$\left\{ \begin{array}{l} \mathcal{G}_{\rightarrow}^- \\ \mathcal{G}_{\rightarrow} \end{array} \right.$

$\mathcal{G}_{\rightarrow}^-$

$\mathcal{G}_{\rightarrow}$

$\cong$

$\partial_0 \lambda_{\rightarrow}$

$\lambda_{\rightarrow}$

$\mathcal{G}$

$\cong$

$\partial_0 \lambda_{\rightarrow}$

PCF

$\mathcal{G}^{\text{QA}}$

$\cong$

$\partial_0 \text{PCF}$

アウトライン

$\mathcal{G}$ に対応する線形資源 λ 計算の導入

- 戦略の直接的計算を可能に

1. 線形資源 λ 計算 $\partial_0 \lambda_{\rightarrow}$
2. $\partial_0 \lambda_{\rightarrow}$ による λ 項の近似
3. 近似とゲームの対応

アウトライン

$\mathcal{G}$ に対応する線形資源 λ 計算の導入

- 戦略の直接的計算を可能に

1. 線形資源 λ 計算 $\partial_0 \lambda_{\rightarrow}$

2. $\partial_0 \lambda_{\rightarrow}$ による λ 項の近似

3. 近似とゲームの対応

線形資源 λ 計算 $\partial_0 \lambda \rightarrow$ [Ehrhard+ 12]

複製にも対応した線形資源 λ 計算

- $[]$ の中身が任意の有限個
 - 使われる回数分だけ、引数を用意する
 - $[]$ の中身は必ずしも同じでなくてもよい

構文

$$r ::= x \mid \lambda x. r \mid r \mu$$

$$\mu ::= [] \mid [r_1, r_2, \dots, r_n]$$

多重集合

型規則

$$\frac{}{x : T \vdash x : T} \quad \frac{\Gamma, x : T \vdash r : U}{\Gamma \vdash \lambda x.r : T \rightarrow U} \quad \frac{\Gamma \vdash r : T \multimap U \quad \Gamma \vdash \mu : T}{\Gamma \vdash r \mu : U}$$

$$\frac{\Gamma \vdash r_1 : T \quad \dots \quad \Gamma \vdash r_n : T}{\Gamma \vdash [r_1, \dots, r_n] : T}$$

※ 型環境の weakening と permutation は省略

構文

$$r ::= x \mid \lambda x.r \mid r \mu$$

$$\mu ::= [] \mid [r_1, r_2, \dots, r_n]$$

多重集合

簡約

β 簡約

- 資源 (= 引数) の個数と 自由変数の個数が
食い違ったら stuck

$$(\lambda x. r) [r_1, \dots, r_n] \longrightarrow r'$$

r' は x の各出現に異なる r_i を代入したもの

例

$$(\lambda x. f [x] [x, x]) [r_1, r_2, r_3] \longrightarrow f [r_1] [r_2, r_3]$$

$$(\lambda x. f [x] [x, x]) [r_1, r_2, r_3] \longrightarrow f [r_2] [r_1, r_3]$$

$$(\lambda x. f [x] [x, x]) [r_1, r_2, r_3] \longrightarrow f [r_3] [r_1, r_2]$$

簡約

β 簡約

- 資源 (= 引数) の個数と 自由変数の個数が
食い違ったら stuck

$$(\lambda x. r) [r_1, \dots, r_n] \longrightarrow r'$$

r' は x の各出現に異なる r_i を代入したもの

例 以下の項は stuck する

$$(\lambda x. f [x] [x, x]) [r_1, r_2, r_3, r_4]$$

$$(\lambda x. f [x] [x, x]) [r_1, r_2]$$

簡約

β 簡約

- 資源 (= 引数) の個数と 自由変数の個数が
食い違ったら stuck

$$(\lambda x.r) [r_1, \dots, r_n] \longrightarrow r'$$

r' は x の各出現に異なる r_i を代入したもの

η 展開

$$r \longrightarrow \lambda x.r [x, x, x, \dots, x]$$

標準形

λ 項の $\beta\eta$ 標準形に対応

$$\check{r} ::= \lambda x_1 x_2 \dots x_n. y \check{\mu}_1 \dots \check{\mu}_k$$

$$\left(\begin{array}{l} n \geq 0 \\ k \geq 0 \\ y \check{\mu}_1 \dots \check{\mu}_k : o \end{array} \right)$$

$$\check{\mu} ::= [] \mid [\check{r}_1, \dots, \check{r}_n]$$

正規化

線形資源 λ 項の集合を、その標準形の集合にする

- NF を標準形の線形資源 λ 項の集合とする

$$\text{nf}(X) := \{ r' \in NF \mid \exists r \in X. r \longrightarrow^* r' \}$$

$$\left(\text{nf}(X) := \{ \mu' \in NF \mid \exists \mu \in X. \mu \longrightarrow^* \mu' \} \right)$$

アウトライン

$\mathcal{G}$ に対応する線形資源 λ 計算の導入

- 戦略の直接的計算を可能に

1. 線形資源 λ 計算 $\partial_0 \lambda_{\rightarrow}$

2. $\partial_0 \lambda_{\rightarrow}$ による λ 項の近似

3. 近似とゲームの対応

λ 項の 線形資源項 による近似

λ 項を 線形資源計算の項の集合とみなす

$$\langle x \rangle := \{x\}$$

$$\langle \lambda x.M \rangle := \{ \lambda x.r \mid r \in \langle M \rangle \}$$

$$\langle M N \rangle := \{ r \mu \mid r \in \langle M \rangle, \mu \in \langle\langle N \rangle\rangle \}$$

$$\langle \Omega \rangle := \emptyset$$

$$\langle\langle M \rangle\rangle := \{ [r_1, \dots, r_n] \mid n \geq 0, \forall i \leq n. r_i \in \langle M \rangle \}$$

ルールによる近似の定義

$$\langle M \rangle = \{ r \mid r \sqsubseteq M \}$$

$$\overline{x \sqsubseteq x}$$

$$\frac{r \sqsubseteq M}{\lambda x.r \sqsubseteq \lambda x.M}$$

$$\frac{r \sqsubseteq M \quad r_1 \sqsubseteq N \quad \dots \quad r_n \sqsubseteq N}{r [r_1, \dots, r_n] \sqsubseteq M N}$$

近似の例

$$T = ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$$

$$M = \lambda g.g (\lambda x.g (\lambda y.x))$$

$$\langle M \rangle = \left\{ \begin{array}{l} \lambda g.g [] \\ \lambda g.g [\lambda x.g []] \\ \lambda g.g [\lambda x.g [\lambda y.x]] \\ \lambda g.g [\lambda x.g [\lambda y.x, \lambda y'.x]] \\ \lambda g.g [\lambda x.g [\lambda y.x], \lambda x'.g []] \\ \vdots \end{array} \right\}$$

[...] が異なる項の集まり

近似の妥当性 [Ehrhard & Regnier 06]

定理 以下は同値。

1. $M =_{\beta\eta} N$
2. $\text{nf}(\langle M \rangle) = \text{nf}(\langle N \rangle)$
3. $\text{nf}(\langle\langle M \rangle\rangle) = \text{nf}(\langle\langle N \rangle\rangle)$

アウトライン

$\mathcal{G}$ に対応する線形資源 λ 計算の導入

- 戦略の直接的計算を可能に

1. 線形資源 λ 計算 $\partial_0 \lambda_{\rightarrow}$

2. $\partial_0 \lambda_{\rightarrow}$ による λ 項の近似

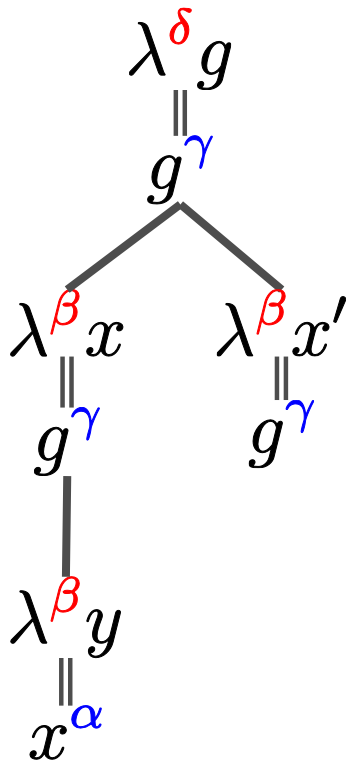
3. 近似とゲームの対応

近似 から justified seq. $\wedge$

$$T = ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$$

$$M = \lambda g. g (\lambda x. g (\lambda y. x))$$

$$[\lambda g. g [\lambda x. g [\lambda y. x], [\lambda x'. g []]]] \in \langle\langle M \rangle\rangle$$



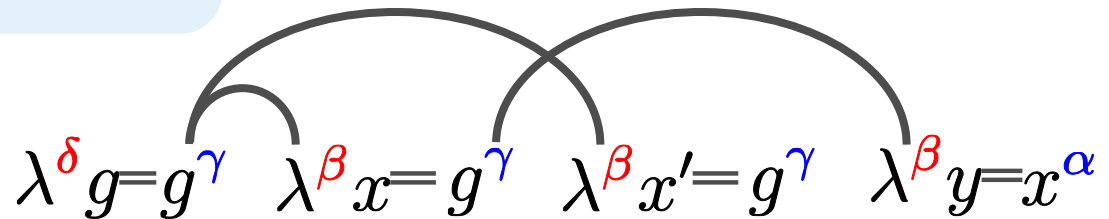
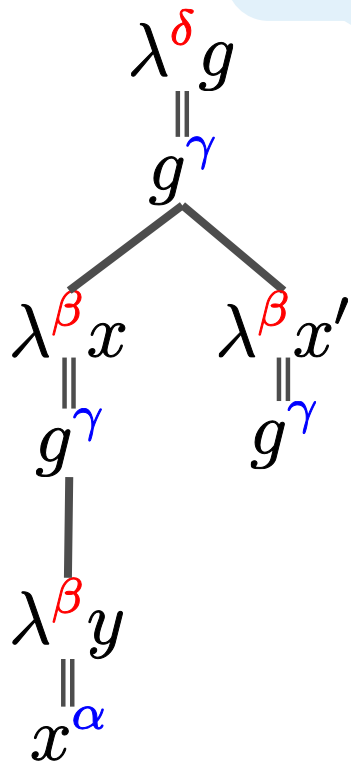
近似 から justified seq. $\wedge$

$$T = ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$$

$$M = \lambda g. g (\lambda x. g (\lambda y. x))$$

- [一列にする
- = は離さない
 - 祖先は左

$$[]] \in \langle\langle M \rangle\rangle$$



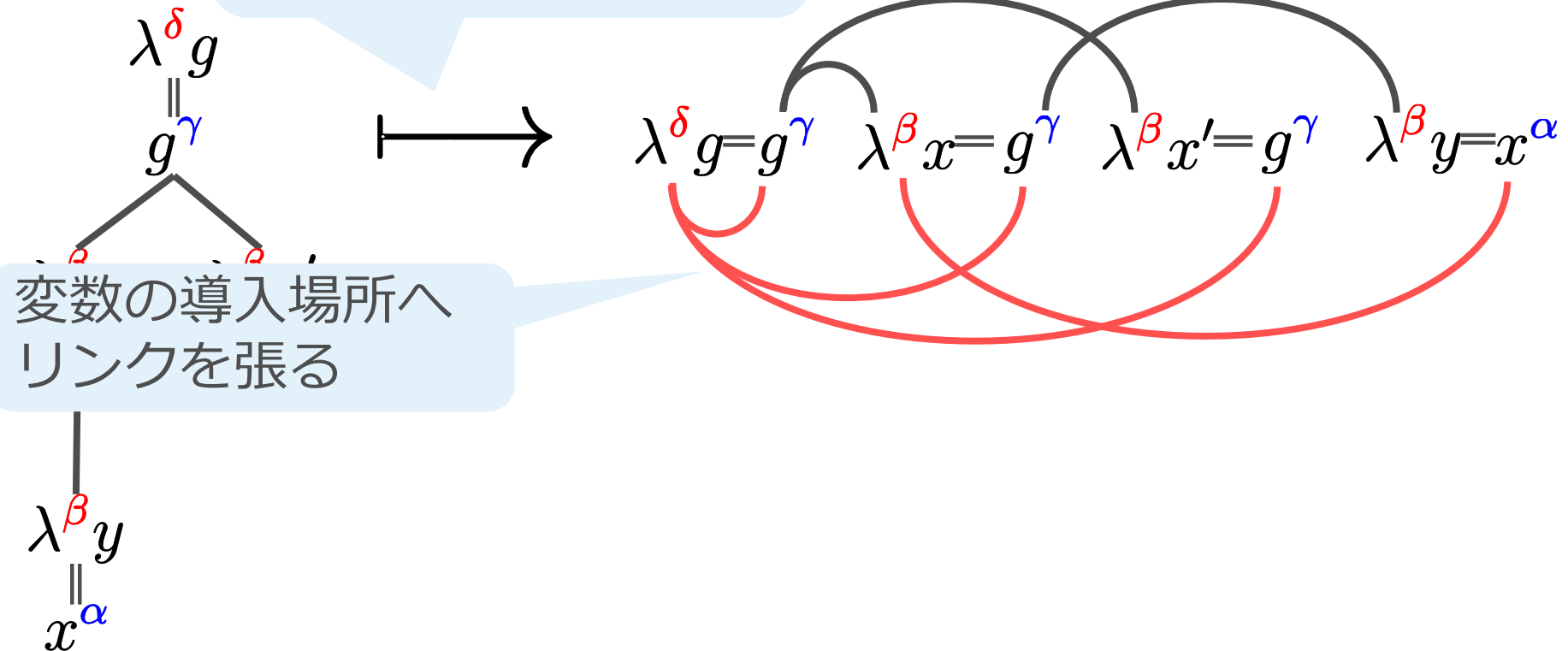
近似 から justified seq. $\wedge$

$$T = ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$$

$$M = \lambda g. g (\lambda x. g (\lambda y. x))$$

- [一列にする
- = は離さない
 - 祖先は左

$$[]]] \in \langle\langle M \rangle\rangle$$



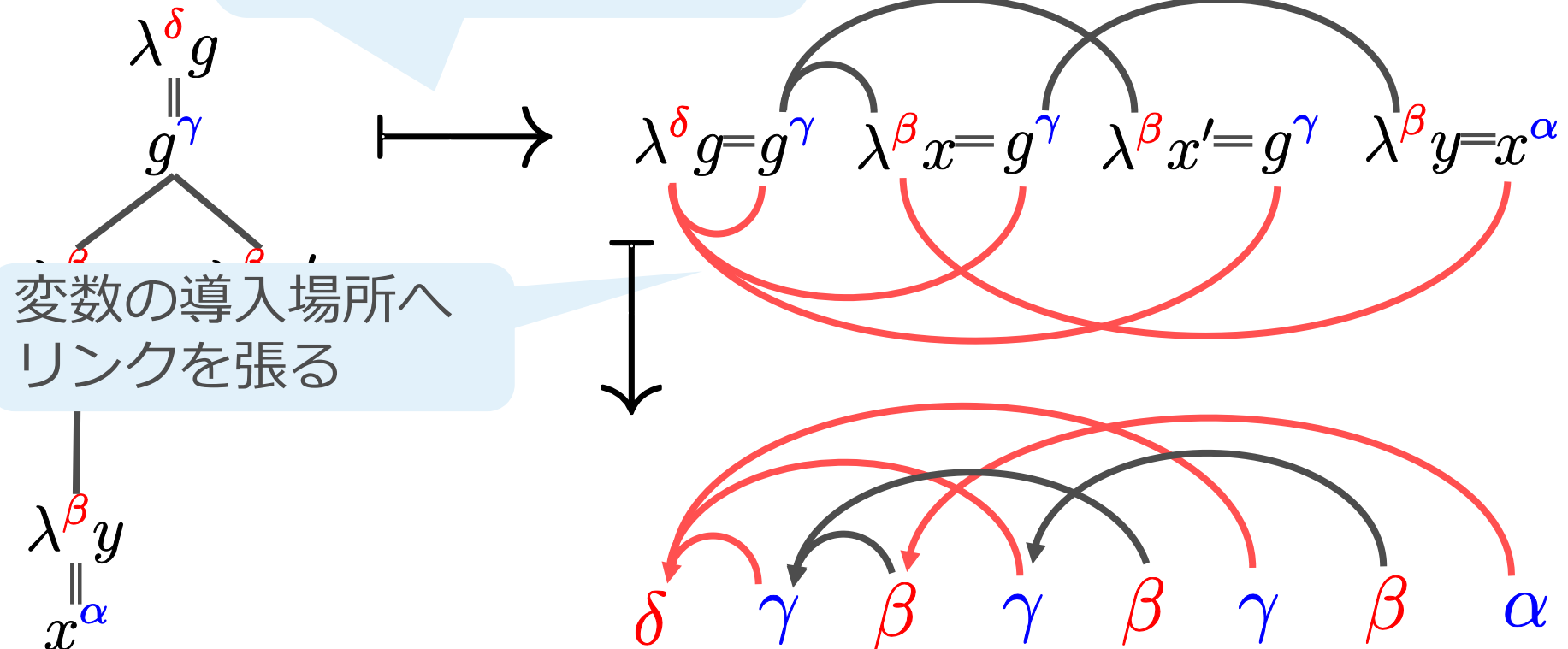
近似 から justified seq. $\wedge$

$$T = ((\alpha \rightarrow \beta) \rightarrow \gamma) \rightarrow \delta$$

$$M = \lambda g. g (\lambda x. g (\lambda y. x))$$

$$[] \in \langle\langle M \rangle\rangle$$

- [一列にする
- = は離さない
 - 祖先は左



線形近似とゲーム解釈の一致

$$\Phi_{\Gamma, T} : \{ \mu \mid \Gamma \vdash \mu : T \} \rightarrow J_{\llbracket \Gamma \rrbracket, \llbracket T \rrbracket}$$

を先ほどの対応関係を表す関数とする。

定理 $\llbracket M \rrbracket = \bigcup \{ \Phi(\mu) \mid \mu \in \text{nf}(\langle\langle M \rangle\rangle) \}$

$\text{id}_{\llbracket T \rrbracket}$ と ev の定義

$$\text{id}_{\llbracket T \rrbracket} := \{ \Phi(\mu) \mid \mu \in \text{nf}(\langle\langle x \rangle\rangle) \}$$

$$\text{ev}_{\llbracket T \rrbracket, \llbracket U \rrbracket} := \{ \Phi(\mu) \mid \mu \in \text{nf}(\langle\langle g \ x \rangle\rangle) \}$$

$$\left(\begin{array}{l} x : T \\ g : T \rightarrow U \end{array} \right)$$

PCFの意味論

<u>対象言語</u>		<u>ゲーム</u>		<u>線形資源計算</u>
Affine λ	{	$\mathcal{G}_{\rightarrow}$		
		$\mathcal{G}_{\rightarrow}$	$\cong$	$\partial_0 \lambda_{\rightarrow}$
$\lambda_{\rightarrow}$		$\mathcal{G}$	$\cong$	$\partial_0 \lambda_{\rightarrow}$
PCF		$\mathcal{G}^{\text{QA}}$	$\cong$	$\partial_0 \text{PCF}$

アウトライン

1. 再帰の解釈
2. データ型の解釈
3. PCF のための線形資源計算

アウトライン

1. 再帰の解釈
2. データ型の解釈
3. PCF のための線形資源計算

再帰の解釈 (cf. [Hyland & Ong 2000])

$Y = \lambda f. f \left(f \left(f \left(f \left(\dots \right) \right) \right) \right)$ であることを利用する

1. 有限展開による近似

$$F_0 := \Omega \quad F_{n+1} := f F_n$$

2. 各有限展開を線形近似

$$\langle\langle Y \rangle\rangle := \bigcup_n \langle\langle \lambda f. F_n \rangle\rangle$$

3. 線形近似から justified sequence を定義

$$\llbracket Y \rrbracket := \{ \Phi(\mu) \mid \mu \in \langle\langle Y \rangle\rangle \}$$

アウトライン

1. 再帰の解釈

2. データ型の解釈

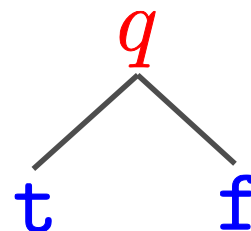
- 基本的なアイデア: Church encoding
- Question / Answer と well-bracketing

3. PCF のための線形資源計算

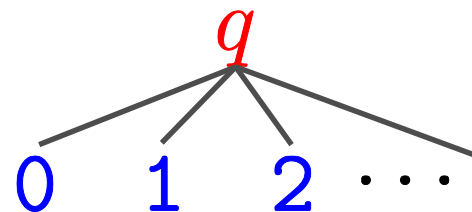
データ型の解釈

基本的には Church encoding

$\llbracket \mathbf{bool} \rrbracket \quad o \rightarrow o \rightarrow o$



$\llbracket \mathbf{nat} \rrbracket \quad \begin{array}{l} \text{無限引数関数} \\ (o \times o \times \dots) \rightarrow o \end{array}$



$$\llbracket \mathbf{t} \rrbracket = \lambda x_t x_f. x_t$$

$$\llbracket \mathbf{iszero} \rrbracket = \lambda n. \lambda x_t x_f. n(x_t, x_f, x_f, x_f, \dots)$$

単純な Church encoding の問題点

コントロールオペレータ のようなものを書ける

例 $\llbracket (\text{bool} \rightarrow \text{bool}) \rightarrow \text{bool} \rrbracket$

$$\lambda h. \lambda x_t x_f. h (\lambda y_t y_f. x_t) x_t x_f$$

$$? \llbracket \lambda h. (\text{call/cc } (\lambda k. h (k \text{ t}))) \rrbracket$$

$$\left(\text{cf. } \lambda h. \lambda x_t x_f. h (\lambda y_t y_f. y_t) x_t x_f = \llbracket \lambda h. h \text{ t} \rrbracket \right)$$

true

アウトライン

1. 再帰の解釈

2. データ型の解釈

- 基本的なアイデア: Church encoding
- Question / Answer と well-bracketing

3. PCF のための線形資源計算

Q/A 付きアリーナ

定義 O/A 付きのアリーナとは 四つ組

$$A = (M_A^O, M_A^P, \vdash_A, \lambda_A)$$

ただし

- $(M_A^O, M_A^P, \vdash_A)$ はアリーナ
- $\lambda_A : M_A \rightarrow \{Q, A\}$ は Q/A ラベル関数

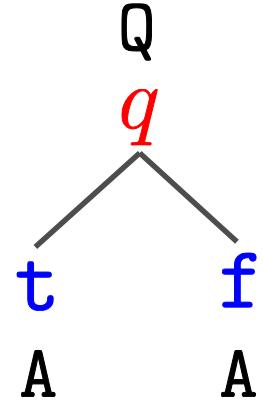
であって、次を満たす

$$\frac{\exists a' \in M_A. a \vdash_A a'}{\lambda_A(a) = Q}$$

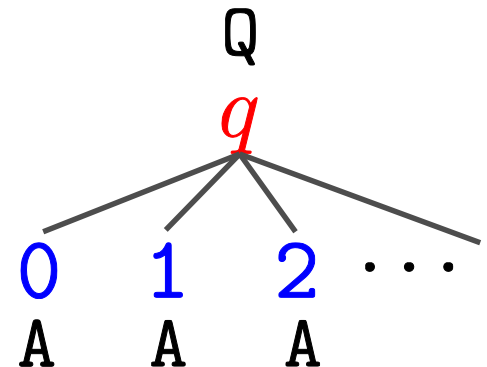
$$\frac{\lambda_A(a) = A}{\exists a' \in M_A. a' \vdash_A a}$$

データ型の解釈 (Q/A 付き)

[[bool]] $o \rightarrow o \rightarrow o$



[[nat]] 無限引数関数
 $(o \times o \times \dots) \rightarrow o$



Well-bracketing

定義 A と B を Q/A 付きアリーナとし、
 p をゲーム (A, B) の 偶数長の P-view とする。

次のいずれかが成り立つとき、 p は well-bracketing

1. p の最後の move は Question
2. p の最後の move は
O が最後にした Question をポイントしている

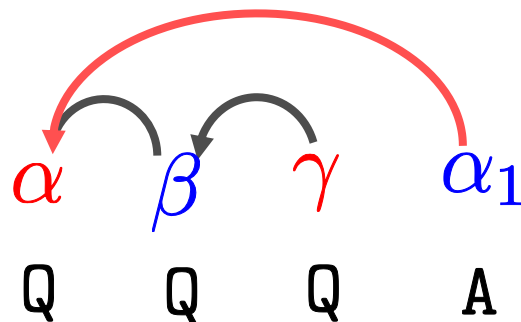
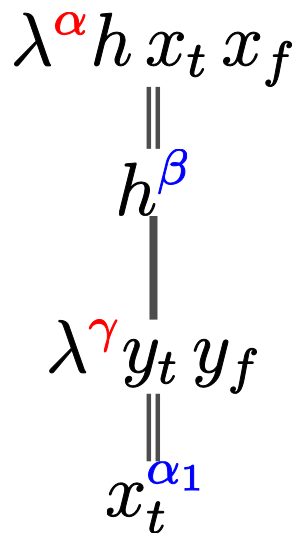
定義 Innocent 戦略 σ が well-bracketing であるとは
 $\sigma \cap V_{A,B}$ が well-bracketing P-view のみから成ること

Well-bracketing でない例

例 $\llbracket (\mathbf{bool} \rightarrow \mathbf{bool}) \rightarrow \mathbf{bool} \rrbracket$

$$= ((\underset{\mathbf{A}}{\gamma_1} \rightarrow \underset{\mathbf{A}}{\gamma_2} \rightarrow \underset{\mathbf{Q}}{\gamma}) \rightarrow (\underset{\mathbf{A}}{\beta_1} \rightarrow \underset{\mathbf{A}}{\beta_2} \rightarrow \underset{\mathbf{Q}}{\beta})) \rightarrow (\underset{\mathbf{A}}{\alpha_1} \rightarrow \underset{\mathbf{A}}{\alpha_2} \rightarrow \underset{\mathbf{Q}}{\alpha})$$

$$\lambda h x_t x_f. h [\lambda y_t y_f. x_t] [] [] \sqsubseteq \lambda h. \lambda x_t x_f. h (\lambda y_t y_f. x_t) x_t x_f$$



$\mathcal{G}^{\text{QA}}$ による解釈の性質

定理 (モデルの妥当性)

$$M =_{\beta\eta} N \text{ ならば } \llbracket M \rrbracket = \llbracket N \rrbracket$$

※ $\beta\eta$ -theory でなく Böhm-theory なら逆も成立

定理 (compact definability)

$(\llbracket \Gamma \rrbracket, \llbracket T \rrbracket)$ の innocent 戦略 σ について、
 σ が compact かつ well-bracketing ならば
PCF項 $\Gamma \vdash M : T$ が存在して、

$$\sigma = \llbracket \Gamma \vdash M : T \rrbracket$$

アウトライン

1. 再帰の解釈
2. データ型の解釈
3. PCF のための線形資源計算

PCF のための線形資源計算

PCF のゲームにも、対応する線形資源計算が作れる

- 詳細は省略

構文

$$r ::= x \mid \lambda x. r \mid r [r_1, \dots, r_n] \\ \mid \text{case } r \text{ of } [v_1 \mapsto r_1, \dots, v_n \mapsto r_n]$$

$$v ::= \mathbf{t} \mid \mathbf{f} \mid 0 \mid 1 \mid \dots$$

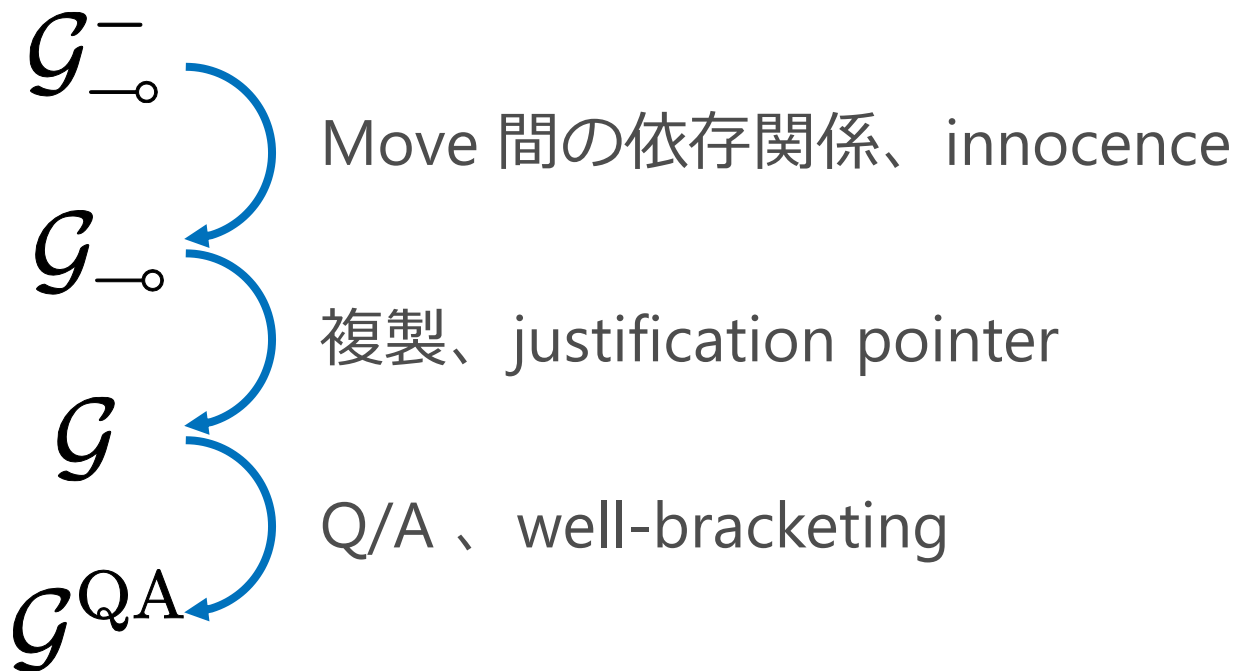
簡約

$$\text{case } v \text{ of } [v \mapsto r] \longrightarrow r$$

おわりに

まとめ

HO game model [Hyland & Ong 2000] の紹介



各モデルに対応する線形資源 λ 計算がある

発展と応用

発展

- PCF 以外のプログラミング言語の意味論
- 証明の意味論

応用例

- アルゴリズムミックな結果の証明

PCFを超えて

- 参照
 - Bad variable [Abramsky & McCusker 01] [Abramsky+ 98]
 - Nominal reference [Tzevelekos 01][Murawski & Tzevelekos 11]
- コントロールオペレータ
 - cf. Semantic cube [Abramsky & McCusker 99]
- 非決定性分岐・確率的分岐
 - [Harmer & McCusker 99] [Danos & Harmer 02]
- 並行計算 [Laird 01] [Laird 05]
- 多相性 [Abramsky & Jagadeesan 05] [Laird 13]

論理とゲーム

ゲーム意味論は線形論理 [Girard 87] と深い関係にある

- Blass game [Blass 92]
- MLL のゲーム意味論 [Abramsky & Jagadeesan 94]
- LL のゲーム意味論 [Baillot+ 99][Mellies 05]
- Polarised logic [Laurent 02]

アルゴリズムミシックな結果

戦略のオートマトン表現を利用

- Idealised Algol の観測同値の決定可能性
 - 2nd-order [Ghica & McCusker 02]
 - 3rd-order [Ong 02]
- Reduced ML の観測同値性の決定可能 / 不可能性

[Hopkins+ 11]

相互作用列による計算の分析を通して

- 高階モデル検査の決定可能性 [Ong 06]
- 高階マッチング問題の決定可能性 [Stirling 09]

ゲーム意味論の入門テキスト

"Notes on game semantics" [Curien, available online]

"Game semantics" [Abramsky & McCusker 99]