

情報数学

第6回 ラムダ計算

萩野 達也

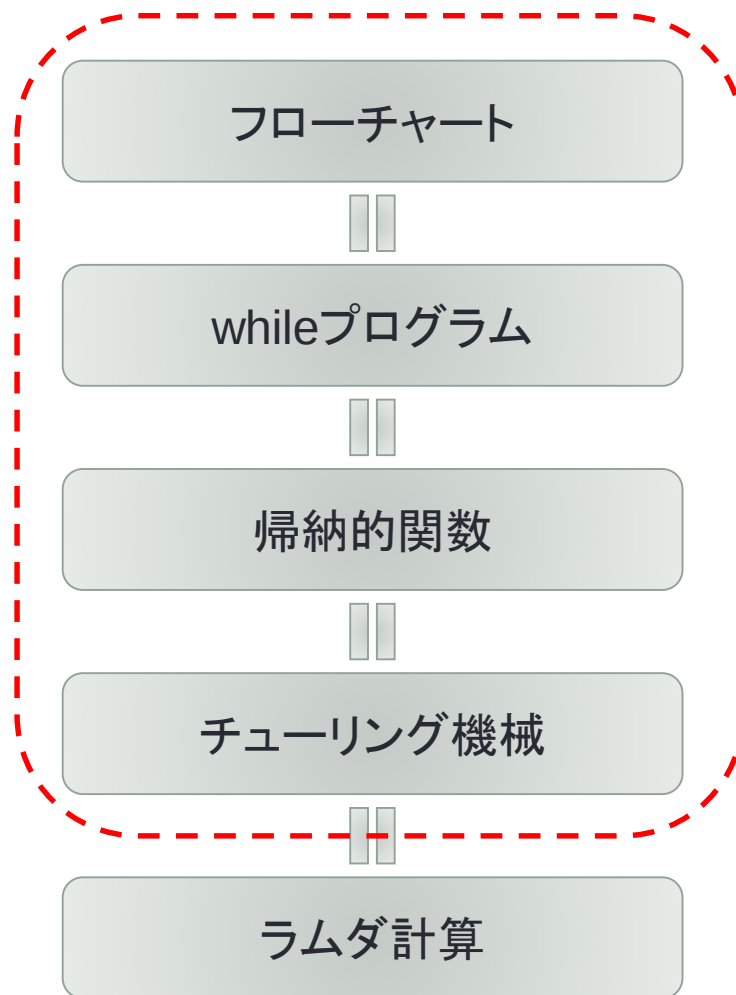
hagino@sfc.keio.ac.jp

スライドURL

<https://vu5.sfc.keio.ac.jp/slide/>

ここまで

- 計算
 - フローチャート
 - whileプログラム
 - 帰納的関数
 - 原始帰納的関数
 - 最小解演算子
 - チューリング機械
- 決定不能問題
 - 停止問題
 - 全域性問題
 - ポスト対応問題



ラムダ記法

- 関数の定義 (関数抽象)

- $f(x) = x + x \times x$

- $f = \lambda x. x + x \times x$

- 関数の利用 (関数適用)

- $f(2) = 2 + 2 \times 2 = 6$

- $(\lambda x. x + x \times x)(2) = 2 + 2 \times 2 = 6$

- 高階関数

- $twice(f) = \lambda x. f(f(x))$

- $twice = \lambda f. (\lambda x. f(f(x)))$

- $twice(\lambda x. x + x \times x) =$

カーリー化 (Curry化)

- 2引数以上の関数
 - $f(x, y) = x \times (y + 5)$
 - $f: N \times N \rightarrow N$
- Curry化
 - $f^*(x)(y) = x \times (y + 5)$
 - $f^*(x) = \lambda y. (x \times (y + 5))$
 - $f^* = \lambda x. (\lambda y. (x \times (y + 5)))$
 - $f^*: N \rightarrow (N \rightarrow N)$
 - $f(x, y) = f^*(x)(y)$

ラムダ式(λ式)

- 定義: λ式を以下のように定義する.

- (1) 変数 $x, y, z, x_1, x_2, y', \dots$ は λ式である.
- (2) λ式 M と変数 x に対して
 $(\lambda x. M)$
 は λ式である. (関数抽象, function abstraction)
- (3) λ式 M と N に対して
 $(M N)$
 は λ式である. (関数適用, function application)

- 例: x, y, f を変数とする.

- $(\lambda x. (f(f x)))$
- $((\lambda f. (f x))(\lambda y. y))$

- 省略記法

- $\lambda x_1 x_2 \cdots x_n. M \equiv (\lambda x_1. (\lambda x_2. (\cdots (\lambda x_n. M) \cdots)))$
- $M_1 M_2 M_3 \cdots M_n \equiv ((\cdots ((M_1 M_2) M_3) \cdots) M_n)$

束縛変数と自由変数

- **束縛変数** (bound variable)
 - $\lambda x. M$ の M に出てくる変数 x は λx によって**束縛**されている.
 - $\lambda x. (y x)$
 - $\lambda x. (\lambda y. x y z)$
 - $(\lambda x. y x)(\lambda y. y x)$
 - $\lambda xy. x(\lambda y. x y)$
- **自由変数** (free variable)
 - 束縛されていない変数は**自由**な変数
 - $FV(x) = \{x\}$
 - $FV(\lambda x. M) = FV(M) \setminus \{x\}$
 - $FV(M N) = FV(M) \cup FV(N)$
- **閉式** (closed term)
 - 自由変数を持たない λ 式
 - $FV(M) = \emptyset$

α 変換と代入

- 束縛変数を交換しても意味は変わらない.
 - $\lambda x. x \xrightarrow{\alpha} \lambda y. y$
 - $\lambda x. (y(\lambda x. y x)x) \xrightarrow{\alpha} \lambda x. (y(\lambda z. y z)x)$
- α 変換によって自由変数と束縛変数が異なるようにすることができる.
- 代入: $M[x := N]$
 - M の自由変数 x を式 N に置き換える.
 - $((\lambda x. y x) y)[y := (\lambda z. z)] \equiv (\lambda x. (\lambda z. z)x)(\lambda z. z)$
 - 変数の束縛関係を変えてはいけない.
 - $((\lambda x. y x) y)[y := x] \not\equiv (\lambda x. x x) x$
- 代入の形式的定義:
 - $x[x := N] \equiv N$
 - $y[x := N] \equiv y$
 - $(\lambda y. M)[x := N] \equiv \lambda y. (M[x := N])$
(ただし $y \notin FV(N)$ であること)
 - $(\lambda x. M)[x := N] \equiv \lambda x. M$
 - $(M M')[x := N] \equiv (M[x := N] M'[x := N])$
- α 変換:
 - $\lambda x. M \xrightarrow{\alpha} \lambda y. (M[x := y])$

β 変換

- 定義: β 基 $(\lambda x. M)N$ を $M[x := N]$ に置き換える.

$$(\lambda x. M)N \xrightarrow{\beta} M[x := N]$$

N の自由変数が束縛される場合は α 変換を先に行う

- 例:

- $(\lambda x. x)y \xrightarrow{\beta}$

- $(\lambda x. x y)(\lambda x. x) \xrightarrow{\beta}$

- $(\lambda x y. x y)(\lambda x. y) \xrightarrow{\beta}$

- $M \xrightarrow{\beta} N$ ならば,

- $\lambda x. M \xrightarrow{\beta} \lambda x. N$

- $M M' \xrightarrow{\beta} N M'$

- $M' M \xrightarrow{\beta} M' N$

β 变换列

- $P \xrightarrow{\alpha\beta} Q$

- $P \xrightarrow{\alpha} Q \quad \text{or} \quad P \xrightarrow{\beta} Q$

- $P \xRightarrow{\alpha\beta} Q$

- $P \equiv P_1 \xrightarrow{\alpha\beta} P_2 \xrightarrow{\alpha\beta} P_3 \xrightarrow{\alpha\beta} \dots \xrightarrow{\alpha\beta} P_n \equiv Q$

- $P \xleftrightarrow{\alpha\beta} Q$

- $P \xrightarrow{\alpha\beta} Q \quad \text{or} \quad Q \xrightarrow{\alpha\beta} P$

- $P \Leftrightarrow^{\alpha\beta} Q$

- $P \equiv P_1 \xleftrightarrow{\alpha\beta} P_2 \xleftrightarrow{\alpha\beta} P_3 \xleftrightarrow{\alpha\beta} \dots \xleftrightarrow{\alpha\beta} P_n \equiv Q$

再左 β 変換と最右 β 変換

- 最左 β 変換

- 最も左に現れる β 基を変換する.

- $(\lambda x y. y)((\lambda y. y x)(\lambda x. x)) \xrightarrow{\beta}$

- 最右 β 変換

- 最も右に現れる β 基を変換する.

- $(\lambda x y. y)((\lambda y. y x)(\lambda x. x)) \xrightarrow{\beta}$

- β 基を持たない λ 式を正規形 (normal form) という.

- $\lambda x y. x y$
- $\lambda x. x x$
- $x y$

基本的な λ 式

- $I \equiv \lambda x. x$
 - $I M \xrightarrow{\beta}$
- $K \equiv \lambda x y. x$
 - $K M N \xrightarrow{\beta}$
- $S \equiv \lambda x y z. x z (y z)$
 - $S P Q R \xrightarrow{\beta}$
 - $S(\lambda x. M)(\lambda x. N) \xrightarrow{\beta}$
 - $S K K \xrightarrow{\beta}$

SK式

- **SK式:**
 - 変数と S と K を関数適用して構成された λ 式
 - 任意の λ 式 M に対してSK式 X が存在して $X \stackrel{\alpha\beta}{\Rightarrow} M$ とすることができる.
 - SK式 X と変数 x に対してSK式 $\Lambda x. X$ を次のように定義する.
 - $\Lambda x. x \equiv S K K$
 - $\Lambda x. y \equiv K y$
 - $\Lambda x. S \equiv K S$
 - $\Lambda x. K \equiv K K$
 - $\Lambda x. X_1 X_2 \equiv S(\Lambda x. X_1)(\Lambda x. X_2)$
 - $\Lambda x. X \stackrel{\alpha\beta}{\Rightarrow} \lambda x. X$
 - 1つの関数抽象をなくすことができた.
 - 内側から繰り返すことで, すべての関数抽象をなくすことができる.

SK式への変換(例)

- $M \equiv \lambda x y. y$ に対応するSK式を求めよ.

$$\begin{aligned}
 \lambda x. (\lambda y. y) &\equiv \lambda x. S K K \\
 &\equiv \lambda x. (S K) K \\
 &\equiv S (\lambda x. S K) (\lambda x. K) \\
 &\equiv S (S (\lambda x. S) (\lambda x. K)) (\lambda x. K) \\
 &\equiv S (S (K S) (\lambda x. K)) (\lambda x. K) \\
 &\equiv S (S (K S) (K K)) (K K)
 \end{aligned}$$

- $S (S (K S) (K K)) (K K) \stackrel{\alpha\beta}{\Rightarrow} \lambda x y. y$

- $\lambda x. x \equiv S K K$
- $\lambda x. y \equiv K y$
- $\lambda x. S \equiv K S$
- $\lambda x. K \equiv K K$
- $\lambda x. X_1 X_2 \equiv S (\lambda x. X_1) (\lambda x. X_2)$

SK式への変換(演習)

- $M \equiv \lambda x y. x y$ に対応するSK式を求めよ.

- $\lambda x. x \equiv S K K$
- $\lambda x. y \equiv K y$
- $\lambda x. S \equiv K S$
- $\lambda x. K \equiv K K$
- $\lambda x. X_1 X_2 \equiv S(\lambda x. X_1)(\lambda x. X_2)$

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

特別なλ式

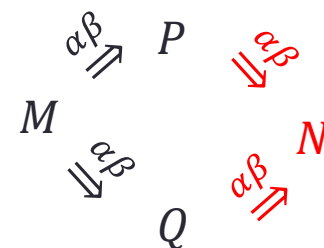
- $Z \equiv (\lambda x. x x)(\lambda x. x x)$
 - $(\lambda x. x x)(\lambda x. x x) \xrightarrow{\beta} (\lambda x. x x)(\lambda x. x x) \xrightarrow{\beta} (\lambda x. x x)(\lambda x. x x) \xrightarrow{\beta}$
- $Y \equiv \lambda y. (\lambda x. y(x x))(\lambda x. y(x x))$
 - Curryの不動点演算子
 - $Y M \stackrel{\alpha\beta}{\Leftrightarrow} M(Y M)$
 - $Y M \xrightarrow{\beta} (\lambda x. M(x x))(\lambda x. M(x x)) \xrightarrow{\beta} M(\lambda x. M(x x))(\lambda x. M(x x))$
 - $Y(\lambda x. x) \xrightarrow{\beta} (\lambda x. x x)(\lambda x. x x)$
- $Y' \equiv (\lambda x y. y(x x y))(\lambda x y. y(x x y))$
 - Turingの不動点演算子
 - $Y' M \stackrel{\alpha\beta}{\Rightarrow} M(Y' M)$

Church-Rosserの定理

- **定理:** $M \xRightarrow{\alpha\beta} P$ および $M \xRightarrow{\alpha\beta} Q$ であれば, N が存在して $P \xRightarrow{\alpha\beta} N$ および $Q \xRightarrow{\alpha\beta} N$ となる.

- 証明は少し難しい.

$$\begin{array}{c}
 (\lambda x. x x)((\lambda y. y) z) \xrightarrow{\beta} (\lambda x. x x) z \xrightarrow{\beta} z z \\
 \downarrow \beta \qquad \qquad \qquad \searrow \beta \\
 ((\lambda y. y) z)((\lambda y. y) z) \xrightarrow{\beta} z ((\lambda y. y) z) \xrightarrow{\beta} z z \\
 \searrow \beta \qquad \qquad \qquad \nearrow \beta \\
 ((\lambda y. y) z) z \xrightarrow{\beta} z z
 \end{array}$$



- この定理から, λ 式の正規形は β 基の選択に関係なく一意的决定することが分かる(α 変換を除く).
- 最左 β 変換により正規形に到達することが知られている.

まとめ

- λ 式
- 変換
 - α 変換
 - β 変換
- SK式
- 正規形
- Church-Rosserの定理