

情報数学

第7回 ラムダ計算と計算可能性

萩野 達也

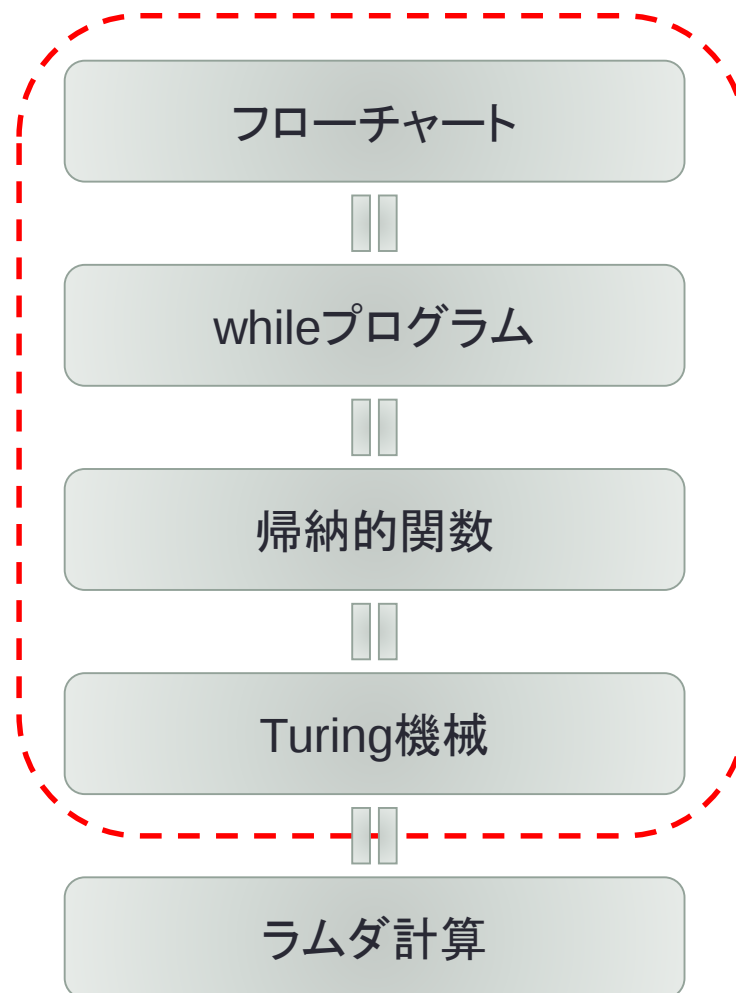
hagino@sfc.keio.ac.jp

スライドURL

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

これまで

- 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
 - Turing機械
 - 決定不可能問題
 - ラムダ計算
 - 関数抽象
 - 関数適用



λ 表現

- 部分関数 $f: N^n \rightarrow N$ の λ 表現 F とは:

- もし $f(k_1, k_2, \dots, k_n) = k$ であれば,

$$F[k_1][k_2] \dots [k_n] \stackrel{\alpha\beta}{\Rightarrow} [k]$$

ここで $[k_i]$ は, 自然数 k_i の λ 表現(次に定義)

- $[f] \equiv F$ を f の λ 表現と呼ぶ.

真偽と対の λ 表現

- 真と偽の λ 表現:

- $[\text{true}] \equiv \lambda x y. x$
- $[\text{false}] \equiv \lambda x y. y$
- $[\text{true}]M N \xRightarrow{\alpha\beta}$
- $[\text{false}]M N \xRightarrow{\alpha\beta}$

- 2つの式の対の λ 表現:

- $[M, N] \equiv \lambda x. x M N$
- $\pi_1 \equiv \lambda x. x [\text{true}]$
- $\pi_2 \equiv \lambda x. x [\text{false}]$
- $\pi_1[M, N] \xRightarrow{\alpha\beta}$
- $\pi_2[M, N] \xRightarrow{\alpha\beta}$

自然数の λ 表現

- 自然数の λ 表現:

- $[0] \equiv \lambda x y. y$
- $[1] \equiv \lambda x y. x y$
- $[2] \equiv \lambda x y. x(x y)$
- $[3] \equiv \lambda x y. x(x(x y))$
- $\vdots$
- $[n] \equiv \lambda x y. x(\overbrace{x(\cdots (x y) \cdots)})^n$

- 演算:

- $[suc] \equiv \lambda x y z. y(x y z)$
- $[add] \equiv \lambda x y z w. x z(y z w)$
- $[mul] \equiv \lambda x y z w. x(y z)w$
- $[pred] \equiv \lambda x y z. x(\lambda u v. v(u y))(\lambda a. z)(\lambda a. a)$
- $[zero?] \equiv \lambda x. x(\lambda x. [false])[true]$

λ表現での計算を試してみよう

- $[\text{suc}][2] \equiv (\lambda x y z. y(x y z))(\lambda x y. x(x y))$

$$\xRightarrow{\alpha\beta}$$

$$\xRightarrow{\alpha\beta}$$

$$\xRightarrow{\alpha\beta}$$

$$\equiv [3]$$

- $[\text{add}][3][2] \equiv (\lambda x y z w. x z(y z w))(\lambda x y. x(x(x y))) (\lambda x y. x(x y))$

$$\xRightarrow{\alpha\beta}$$

$$\xRightarrow{\alpha\beta}$$

$$\xRightarrow{\alpha\beta}$$

$$\xRightarrow{\alpha\beta}$$

$$\equiv [5]$$

帰納的関数

- 原始帰納的関数:

- 基本的関数

- $zero : N^0 \rightarrow N$ $zero() = 0$
 - $suc : N \rightarrow N$ $suc(x) = x + 1$
 - $\pi_i^n : N^n \rightarrow N$ $\pi_i^n(x_1, \dots, x_n) = x_i$

- 関数の合成

- $f(x_1, x_2, \dots, x_n) = g(h_1(x_1, x_2, \dots, x_n), \dots, h_m(x_1, x_2, \dots, x_n))$

- 原始帰納法による関数の定義

- $f(x_1, \dots, x_n, zero()) = g(x_1, \dots, x_n)$
 - $f(x_1, \dots, x_n, suc(y)) = h(x_1, \dots, x_n, y, f(x_1, \dots, x_n, y))$

- 帰納的関数:

- 最小解演算子

- $f(x_1, \dots, x_n) = \mu_y(g(x_1, \dots, x_n, y) = 0)$

原始帰納的関数の λ 表現

- 基本的関数:

- $[\text{zero}] \equiv \lambda x. [0] \equiv \lambda x y z. z$
- $[\text{suc}] \equiv \lambda x y z. y(x y z)$
- $[\pi_i^n] \equiv \lambda x_1 x_2 \cdots x_n. x_i$

- 関数合成:

- $f(x_1, x_2, \dots, x_n) = g(h_1(x_1, x_2, \dots, x_n), \dots, h_m(x_1, x_2, \dots, x_n))$ のとき,

$$[f] \equiv \lambda x_1 x_2 \cdots x_n. [g]([h_1]x_1 x_2 \cdots x_n) \cdots ([h_m]x_1 x_2 \cdots x_n)$$

- 例:

- $\text{dsuc}(x) = \text{suc}(\text{suc}(x))$
- $[\text{dsuc}] \equiv \lambda x. [\text{suc}]([\text{suc}]x)$

原始帰納法

- 単純化して, 引数は2つとする.

- $f(x, \text{zero}()) = g(x)$
- $f(x, \text{suc}(y)) = h(x, y, f(x, y))$

- $F \equiv [f]$ の構成:

- F は次の性質を持たなくてはならない:

$$F x y \stackrel{\alpha\beta}{\Rightarrow} [\text{zero?}] y ([g] x) ([h] x ([\text{pred}] y) (F x ([\text{pred}] y)))$$

- F は次の M の不動点:

$$M \equiv \lambda f x y. [\text{zero?}] y ([g] x) ([h] x ([\text{pred}] y) (f x ([\text{pred}] y)))$$

- Curryの不動点演算子 $Y \equiv \lambda y. (\lambda x. y(x x))(\lambda x. y(x x))$ を用いると:

$$F \equiv Y M$$

$$F \equiv Y \left(\lambda f x y. [\text{zero?}] y ([g] x) ([h] x ([\text{pred}] y) (f x ([\text{pred}] y))) \right)$$

最小解演算子

- 単純化して, 引数は1つとする.

- $f(x) = \mu_y(g(x, y) = 0)$

- $F \equiv [f]$ の構成:

- H を次の性質を満たす λ 式とする:

$$H x y \stackrel{\alpha\beta}{\Rightarrow} [\text{zero?}] ([g] x y) y (H x ([\text{suc}] y))$$

- H はCurryの不動点演算子 Y を用いることで定義できる:

$$H \equiv Y \left(\lambda h x y. [\text{zero?}] ([g] x y) y (h x ([\text{suc}] y)) \right)$$

- $F \equiv \lambda x. H x [0]$

$$F \equiv \lambda x. Y \left(\lambda h x y. [\text{zero?}] ([g] x y) y (h x ([\text{suc}] y)) \right) x [0]$$

計算可能性

- 計算可能な関数は λ 表現可能である.
 - 計算可能な関数は帰納的関数である.
 - 帰納的関数は λ 表現を持つ.
- λ 表現可能な関数は計算可能である.
 - λ 表現可能な関数は最左 β 変換により実行可能である.
 - λ 式を数字でコード化できる.
 - 最左 β 変換を行うプログラムを作成することができる.

η 変換

- 関数の外延性 (extensionality)
 - 2つの関数は、与えられた引数に対して同じ値を返す時に等しい.
 - $f = g \iff \forall x (f(x) = g(x))$
- $\alpha\beta$ 同等性では外延性は成り立たない.
 - $P \stackrel{\alpha\beta}{\iff} Q$ であれば, $x \notin \text{FV}(PQ)$ に対して $P x \stackrel{\alpha\beta}{\iff} Q x$ が成り立つ.
 - しかし, 逆は成り立たない: $\lambda x. y x \stackrel{\alpha\beta}{\iff} y$ ではない.
- η 変換:

$$\lambda x. P x \stackrel{\eta}{\rightarrow} P$$

ただし $x \notin \text{FV}(P)$

まとめ

- ラムダ式
- ラムダ式の変換
 - α 変換
 - β 変換
 - η 変換
- 計算可能性
 - λ 表現
 - 自然数の λ 表現

