情報数学
第6回　ラムダ計算

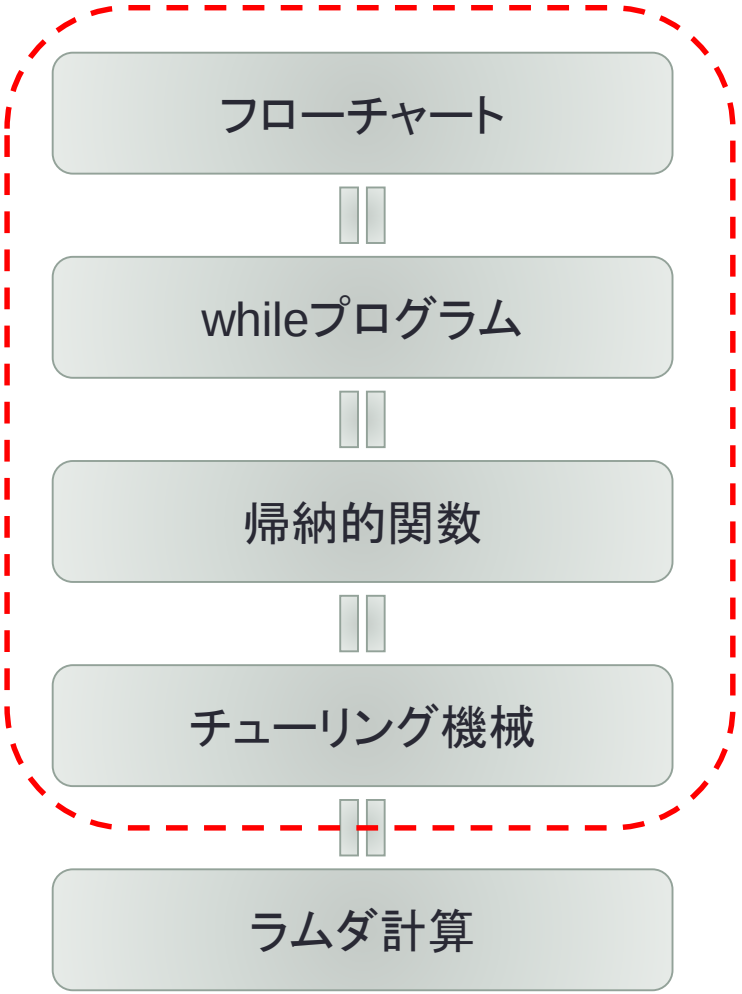
萩野　達也
hagino@sfc.keio.ac.jp

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
  - $twice(f) = \lambda x.\, f(f(x))$
  - $twice = \lambda f.\left(\lambda x.\, f\left(f(x)\right)\right)$
  - $twice(\lambda x.\, x + x \times x) =$

カリー化（Curry化）

- 2引数以上の関数
  - $f(x, y) = x \times (y + 5)$
  - $f: N \times N \to N$

- Curry化
  - $f^*(x)(y) = x \times (y + 5)$
  - $f^*(x) = \lambda y.\left(x \times (y + 5)\right)$
  - $f^* = \lambda x.\left(\lambda y.\left(x \times (y + 5)\right)\right)$
  - $f^*: N \to (N \to N)$
  - $f(x, y) = f^*(x)(y)$

ラムダ式（λ式）

- **定義:** λ式を以下のように定義する.
  (1) 変数 $x, y, z, x_1, x_2, y', \ldots$ 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.
  - $\left(\lambda x.\left(f\left(f\ x\right)\right)\right)$
  - $\left(\left(\lambda f.\left(f\ x\right)\right)\left(\lambda y.\ y\right)\right)$

- 省略記法
  - $\lambda x_1\ x_2 \cdots\ x_n.M \equiv \left(\lambda x_1.\left(\lambda x_2.\left(\cdots\left(\lambda x_n.M\right)\cdots\right)\right)\right)$
  - $M_1\ M_2\ M_3 \cdots\ M_n \equiv \left(\left(\cdots\left(\left(M_1 M_2\right)M_3\right)\cdots\right)M_n\right)$

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.
  - $\lambda x.\, x \overset{\alpha}{\to} \lambda y.\, y$
  - $\lambda x.\, (y(\lambda x.\, y\ x)x) \overset{\alpha}{\to} \lambda x.\, (y(\lambda z.\, y\ z)x)$
- 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.
  - $\big((\lambda x.\, y\ x)\ y\big)[y := (\lambda z.\, z)] \equiv (\lambda x.\, (\lambda z.\, z)x)(\lambda z.\, z)$
  - 変数の束縛関係を変えてはいけない.
  - $\big((\lambda x.\, y\ x)\ y\big)[y := x] \not\equiv (\lambda x.\, x\ x)\ x$
- α変換:
  - $\lambda x.\, M \overset{\alpha}{\to} \lambda y.\, (M[x := y])$

- 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）
  - $(\lambda x.\, M)[x := N] \equiv \lambda x.\, M$
  - $(M\ M')[x := N] \equiv (M[x := N]\ M'[x := N])$

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変換列

- $P \overset{\alpha\beta}{\longrightarrow} Q$
  - $P \overset{\alpha}{\rightarrow} Q \quad \text{or} \quad P \overset{\beta}{\rightarrow} Q$
- $P \overset{\alpha\beta}{\Longrightarrow} Q$
  - $P \equiv P_1 \overset{\alpha\beta}{\rightarrow} P_2 \overset{\alpha\beta}{\rightarrow} P_3 \overset{\alpha\beta}{\rightarrow} \cdots \overset{\alpha\beta}{\rightarrow} P_n \equiv Q$
- $P \overset{\alpha\beta}{\leftrightarrow} Q$
  - $P \overset{\alpha\beta}{\rightarrow} Q \quad \text{or} \quad Q \overset{\alpha\beta}{\rightarrow} P$
- $P \overset{\alpha\beta}{\Leftrightarrow} Q$
  - $P \equiv P_1 \overset{\alpha\beta}{\leftrightarrow} P_2 \overset{\alpha\beta}{\leftrightarrow} P_3 \overset{\alpha\beta}{\leftrightarrow} \cdots \overset{\alpha\beta}{\leftrightarrow} P_n \equiv Q$

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.
  - $(\lambda x\ y.y)\big((\lambda y.y\ x)(\lambda x.x)\big) \xrightarrow{\beta}$
- 最右β変換
  - 最も右に現れる β 基を変換する.
  - $(\lambda x\ y.y)\big((\lambda y.y\ x)(\lambda x.x)\big) \xrightarrow{\beta}$
- 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 $X \overset{\alpha\beta}{\Rightarrow} M$ 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.
      - $\Lambda x.x \equiv S\ K\ K$
      - $\Lambda x.y \equiv K\ y$
      - $\Lambda x.S \equiv K\ S$
      - $\Lambda x.K \equiv K\ K$
      - $\Lambda x.X_1\ X_2 \equiv S(\Lambda x.X_1)(\Lambda x.X_2)$
    - $\Lambda x.X \overset{\alpha\beta}{\Rightarrow} \lambda x.X$
    - 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)

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

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

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

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

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.

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

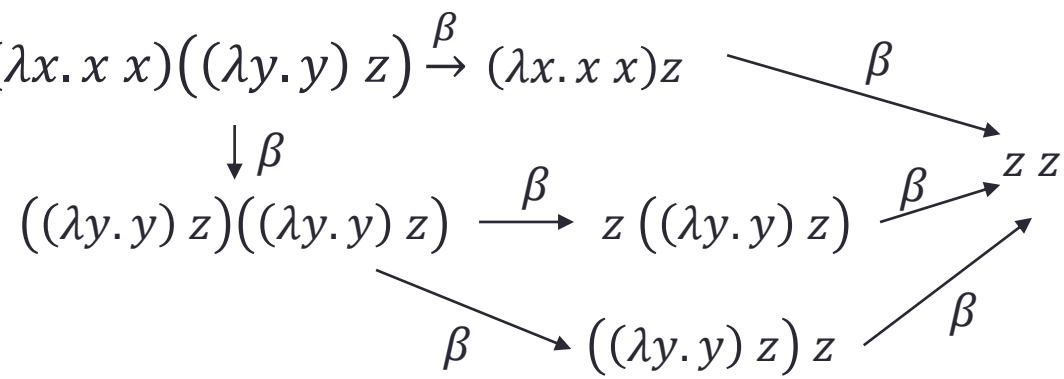
- $\Lambda x.x \equiv S\ K\ K$
- $\Lambda x.y \equiv K\ y$
- $\Lambda x.S \equiv K\ S$
- $\Lambda x.K \equiv K\ K$
- $\Lambda x.X_1\ X_2 \equiv S(\Lambda x.X_1)(\Lambda x.X_2)$

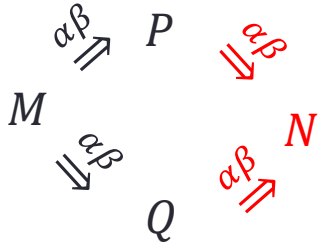
特別なλ式

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

Church-Rosserの定理

- **定理**: $M \overset{\alpha\beta}{\Rightarrow} P$ および $M \overset{\alpha\beta}{\Rightarrow} Q$ であれば，N が存在して $P \overset{\alpha\beta}{\Rightarrow} N$ および $Q \overset{\alpha\beta}{\Rightarrow} N$ となる.
  - 証明は少し難しい.
  - $(\lambda x.\, x\ x)\big((\lambda y.\, y)\ z\big) \overset{\beta}{\to} (\lambda x.\, x\ x)z$





- 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決まることが分かる（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