情報数学

第 7 回 ラムダ計算

萩野 達也

慶應義塾大学環境情報学部

2015/11/10

これまで

- 計算可能
  - フローチャートで計算できる
  - `while` プログラムで計算できる
  - 帰納的関数である
  - チューリング機械で計算できる
- 計算不可能な関数
  - 停止問題
  - 全域問題
  - Post の対応問題
- 決定可能集合と枚挙可能集合

λ記法

- 関数
  - $f(x) = x + x \times x$
  - $f = \lambda x \in N.x + x \times x$
- 適用
  - $f(2) = 2 + 2 \times 2 = 6$
  - $(\lambda x \in N.x + x \times x)(2) = 2 + 2 \times 2 = 6$
- 高階関数
  - $\mathrm{twice}(f) = \lambda x \in N.f(f(x))$
  - $\mathrm{twice} = \lambda f \in (N \to N).(\lambda x \in N.f(f(x)))$
  - $\mathrm{twice}(\lambda x \in N.x + x \times x) =$

Curry 化

- 複数引数の関数
  - $f(x, y) = x \times (y + 5)$
  - $f : N \times N \to N$
- Curry 化
  - $f^\star = \lambda x \in N.(\lambda y \in N.(x \times (y + 5)))$
  - $f^\star : N \to (N \to N)$
  - $f(x, y) = f^\star(x)(y)$

λ 式

定義: λ 式を以下のように定義する.

(1) 変数 $x_0, x_1, \ldots$ は λ 式.

(2) M が λ 式で，x が変数であるとき，$(\lambda x.M)$ は λ 式．(関数抽象, functional abstraction)

(3) M と N が λ 式であるとき，(MN) は λ 式．(関数適用, functional application)

例: x, y, f を変数とする

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

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

自由変数と束縛変数

- 束縛変数 (bound variable)
  - $\lambda x.M$ の M に出てくる x はこの λx によって束縛されている.
  - $\lambda x.(yx)$
  - $\lambda x.(\lambda y.xyz)$
- 自由変数 (free variable)
  - 束縛されていない変数
  - $\mathrm{FV}(x) = \{x\}$
  - $\mathrm{FV}(\lambda x.M) = \mathrm{FV}(M) \backslash \{x\}$
  - $\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$
- 閉式 (closed term)
  - 自由変数を持たない
  - $\mathrm{FV}(M) =$

α 変換と代入

- 束縛変数を交換しても意味は変わらない
  - $\lambda x.x \xrightarrow{\alpha} \lambda y.y$
  - $\lambda x.(y(\lambda x.yx)x) \xrightarrow{\alpha} \lambda x.(y(\lambda z.yz)x)$
- α 変換により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 $M[x := N]$
  - $((\lambda x.yx)y)[y := (\lambda z.z)] = (\lambda x.(\lambda z.z)x)(\lambda z.z)$
- 代入の定義 (変数の束縛関係が変わらないこと)
  - $x[x := N] = N$
  - $y[x := N] = y$
  - $(\lambda y.M)[x := N] = \lambda y.(M[x := N])$
  - $(\lambda x.M)[x := N] = \lambda z.(M[x := z][x := N])$
  - $(MM')[x := N] = (M[x := N])(M'[x := N])$

β 変換

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

- $(\lambda x.M)N \xrightarrow{\beta} M[x := N]$
- $M \xrightarrow{\beta} N$ ならば,
  - $\lambda x.M \xrightarrow{\beta} \lambda x.N$
  - $MP \xrightarrow{\beta} NP$
  - $PM \xrightarrow{\beta} PN$

例:

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

- β 変換に必要ならば α 変換を含めるものとする.

β 変換列

- $P \overset{\beta}{\Longrightarrow} Q$
  - $P \equiv P_1 \overset{\beta}{\longrightarrow} P_2 \overset{\beta}{\longrightarrow} \cdots \overset{\beta}{\longrightarrow} P_n \equiv Q$
- $P \overset{\beta}{\longleftrightarrow} Q$
  - $P \overset{\beta}{\longrightarrow} Q$ または $Q \overset{\beta}{\longrightarrow} P$
- $P \overset{\beta}{=} Q$
  - $P \equiv P_1 \overset{\beta}{\longleftrightarrow} P_2 \overset{\beta}{\longleftrightarrow} \cdots \overset{\beta}{\longleftrightarrow} P_n \equiv Q$

最左 β 変換と最右 β 変換

- 最左 β 変換
  - 最も左の β 基を変換する
  - $(\lambda xy.y)((\lambda y.yx)(\lambda x.x)) \xrightarrow{\beta}$
- 最右 β 変換
  - 最も右の β 基を変換する
  - $(\lambda xy.y)((\lambda y.yx)(\lambda x.x)) \xrightarrow{\beta}$
- β 基を持たない λ 式を正規形 (normal form) という
  - $\lambda x.xx$

基本的なλ式

- $I \equiv \lambda x.x$
  - $IM \xrightarrow{\beta}$
- $K \equiv \lambda xy.x$
  - $KMN \xrightarrow{\beta}$
- $S \equiv \lambda xyz.xz(yz)$
  - $SPQR \xrightarrow{\beta}$
  - $S(\lambda x.M)(\lambda x.N) \xrightarrow{\beta}$
  - $SKK \xrightarrow{\beta}$

特殊なλ式

- $Z \equiv (\lambda x.xx)(\lambda x.xx)$
  - $(\lambda x.xx)(\lambda x.xx) \xrightarrow{\beta}$
- $Y \equiv \lambda y.(\lambda x.y(xx))(\lambda x.y(xx))$
  - Curry の不動点演算子
  - $YM \overset{\beta}{=} M(YM)$
  - $YM \xrightarrow{\beta}$
  - $Y(\lambda x.x) \xrightarrow{\beta}$
- $Y' \equiv (\lambda xy.y(xxy))(\lambda xy.y(xxy))$
  - Turing の不動点演算子
  - $Y'M \overset{\beta}{\Longrightarrow} M(Y'M)$

SK 式

- SK 式
  - 変数と S と K と関数適用により構成された λ 式
- 任意の λ 式 M に対して，SK 式 X が存在して $X \stackrel{\beta}{\Longrightarrow} M$
  - SK 式 X と変数 x に対して SK 式 $\Lambda x.X$ を定義
    - $\Lambda x.x \equiv SKK$
    - $\Lambda x.y \equiv Ky$
    - $\Lambda x.S \equiv KS$
    - $\Lambda x.K \equiv KK$
    - $\Lambda x.X_1X_2 \equiv S(\Lambda x.X_1)(\Lambda x.X_2)$
  - $\Lambda x.X \stackrel{\beta}{\Longrightarrow} \lambda x.X$
  - M の λ を Λ に置き換える
- $M = \lambda xy.xy$ に対応する SK 式を求めよ
  - $\Lambda x.(\Lambda y.xy) \equiv$

Church-Rosser の定理

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

- 証明は少し複雑
- $(\lambda x.xx)((\lambda y.y)z) \xrightarrow{\beta}$

- λ 式のどの β 基を変換しても，最終結果は変わらないことを意味している．
- この定理から，M が正規形 N を持つ時 $(M \xrightarrow{\beta} N)$，$N$ は α 変換を除いて一意的である．
- 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