計算の理論 II
帰納的関数3

月曜5校時

大月美佳

今日の講義内容

1. 復習
   1. 原始帰納的関数の定義
   2. 原始帰納的な集合・述語
2. 原始帰納的な集合・述語つづき
3. 部分帰納的関数と帰納的関数

原始帰納的関数
(primitive recursive)

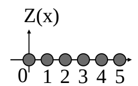
- 定義

  初期関数(1), (2), (3)に
  操作(I), (II)を
  有限回(0回以上)適用して
  得られた関数。

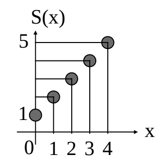
初期関数

- 原始帰納的関数の素。

(1) $Z(x)=0$
  どんなxも0にしてしまう。



(2) $S(x)=x+1$
  xに1を加える。



(3) $U_n^i(x_1, \ldots, x_i, \ldots, x_n)=x_i$
  i番目のx_iを取り出す。

これらに操作を加えて原始帰納的関数を作成する。

合成と原始帰納

- **初期関数に加える操作**
  - (I) **合成**
    - r**変数の関数**h**と**r**個の**n**変数関数**$g_i(1\leqq i\leqq r)$**から、**
    - n**変数の関数**f**を以下の操作で作ること。**
    - $f(x_1, \ldots, x_n)=h(g_1(x_1, \ldots, x_n), \ldots, g_r(x_1, \ldots, x_n))$
  - (II) **原始帰納** (primitive recursion)
    - n-1**変数の関数**g**と**n+1**変数の関数**h**から、**
    - n**変数の関数**f**を以下の操作で作ること。**
    - $f(x_1, \ldots, x_n)=g(x_1, \ldots, x_{n-1})$ ($x_n=0$**のとき**)
    - $f(x_1, \ldots, x_n)=h(x_1, \ldots, x_{n-1}, x_n-1, f(x_1, \ldots, x_n-1))$ ($x_n>0$**のとき**)

原始帰納的な集合と述語

特徴関数C_S, C_p

集合$S\in N^n$

$C_S(x_1, \ldots, x_n)=0$ 　($(x_1, \ldots, x_n)\in S$**のとき**)

$C_S(x_1, \ldots, x_n)=1$ 　($(x_1, \ldots, x_n)\notin S$**のとき**)

が原始帰納的であるとき、S**は原始帰納的集合。**

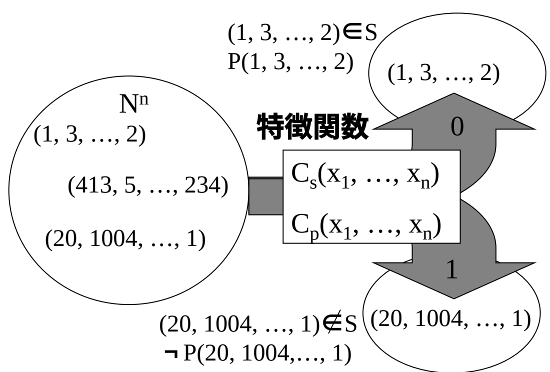
述語$P(x_1, \ldots, x_n)$

$C_p(x_1, \ldots, x_n)=0$ 　($P(x_1, \ldots, x_n)$**のとき**)

$C_p(x_1, \ldots, x_n)=1$ 　($\neg P(x_1, \ldots, x_n)$**のとき**)

が原始帰納的であるとき、P**は原始帰納的述語。**

イメージ図



原始帰納的述語と関数の例 1

(13) **述語**$x=y$

特徴関数$C_p=sg(|x-y|)$

(14) **述語**$x<y$

特徴関数$C_p=1 \dot{-} sg(y \dot{-} x)$

(15) **述語**$x\leqq y$

特徴関数$C_p=sg(x \dot{-} y)$

原始帰納的集合の性質

集合$S, R \in N^n$が原始帰納的であれば、

$\overline{S} = N^n - S$

$S \cup R$

$S \cap R$

も原始帰納的。

原始帰納的述語の性質 1

$P(x_1, \ldots, x_r)$を原始帰納的述語とし、
$h_1, \ldots, h_r$をn変数の原始帰納的関数とする。

このとき述語

$P(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n))$

は原始帰納的。

合成による特徴関数

$h_1, \ldots, h_r$はn変数の原始帰納的関数であるので、$P(x_1, \ldots, x_r)$の特徴関数$C_p(x_1, \ldots, x_r)$に$h_1, \ldots, h_r$を代入した

$C_p(h_1(x_1, \ldots, x_n), \ldots, h_r(x_1, \ldots, x_n))$も

特徴関数とすることができる

例：$x<y$と$a+b$, $c-d$

新しい述語は、$(a+b) < (c-d)$

特徴関数は、$1 - sg((a+b) \dot{-} (c-d))$

原始帰納的述語の性質 2

$g_1, \ldots, g_{m+1}$をn変数の原始帰納的関数とし、
$P_1, \ldots, P_m$をn変数の原始帰納的述語で、
各$(x_1, \ldots, x_n)$に対して高々1個の$P_i(x_1, \ldots, x_n)$が真になるものとする。

このとき関数

$f(x_1, \ldots, x_n) = g_1(x_1, \ldots, x_n)$ （$P_1(x_1, \ldots, x_n)$のとき）

…

$f(x_1, \ldots, x_n) = g_m(x_1, \ldots, x_n)$ （$P_m(x_1, \ldots, x_n)$のとき）

$f(x_1, \ldots, x_n) = g_{m+1}(x_1, \ldots, x_n)$ （それ以外のとき）

は原始帰納的。→証明

原始帰納的述語と関数の例 2

場合わけと合成

(13)**関数**max(x, y)

$$\max(x,y)=\begin{cases} x & (x \ge y\text{のとき}) \\ y & (\text{それ以外のとき}) \end{cases}$$

(14)**関数**min(x, y)

$$\min(x,y)=\begin{cases} x & (x \le y\text{のとき}) \\ y & (\text{それ以外のとき}) \end{cases}$$

(13) **関数**max(x_1, …, x_n)

= max(…max(x_1, x_2), x_3)…), x_n)

原始帰納的述語の性質 3

P(x_1, …, x_n), Q(x_1, …, x_n)**を**
原始帰納的述語とすれば、述語

¬ P(x_1, …, x_n)
P(x_1, …, x_n)∨Q(x_1, …, x_n)
P(x_1, …, x_n)∧Q(x_1, …, x_n)
P(x_1, …, x_n)→Q(x_1, …, x_n)
P(x_1, …, x_n)←Q(x_1, …, x_n)
は原始帰納的。→証明

原始帰納的述語の性質 4

P(x_1, …, x_n, y)**を原始帰納的述語とすれば、述語**

($\exists$y)$_{<z}$P(x_1, …, x_n , y)
⇔P(x_1, …, x_n , 0)∨…∨ P(x_1, …, x_n , z-1)
($\forall$y)$_{<z}$P(x_1, …, x_n , y)
⇔P(x_1, …, x_n , 0)∧…∧ P(x_1, …, x_n , z-1)
は原始帰納的。

特徴関数4

P(x_1, …, x_n , 0)∨…∨ P(x_1, …, x_n , z-1)

$$g(x_1,...,x_n,z)=\prod_{y<z} Cp(x_1,...,x_n,y)$$

P(x_1, …, x_n , 0)∧…∧ P(x_1, …, x_n , z-1)

$$h(x_1,...,x_n,z)=sg(\sum_{y<z} Cp(x_1,...,x_n,y))$$

原始帰納的述語と関数の例 3

(13) **述語**$x|y$ (**xはyを割り切る**)

$x|y \Leftrightarrow (\exists z)_{\leqq y}(x \cdot z=y)$

(14) **述語**$P_r(x)$ (**xは素数である**)

$P_r(x) \Leftrightarrow (x > 1) \wedge \neg(\exists y)_{<x}(y|x) \wedge (y > 1)$

有界μ作用素
(μ-operator)

n+1**変数の述語から**n**変数の関数を作る操作**

定義

述語$P(x_1, \ldots, x_n, y)$**に対して**

$(\mu y)_{<z}P(x_1, \ldots, x_n, y)$

$=\min\{y|y<z \wedge P(x_1, \ldots, x_n, y)\}$

$((\exists y)_{<z}P(x_1, \ldots, x_n, y)$**のとき**$)$

$(\mu y)_{<z}P(x_1, \ldots, x_n, y)$

$=z$

$(\neg(\exists y)_{<z}P(x_1, \ldots, x_n, y)$**のとき**$)$

有界μ作用素と原始帰納

述語$P(x_1, \ldots, x_n, y)$**が原始帰納であれば、**
関数$(\mu y)_{<z}P(x_1, \ldots, x_n, y)$**は**
原始帰納である

原始帰納的述語と関数の例 4

(21) **関数**x/y (**xをyで割ったときの商**)

(22) **第**n+1**番目の素数を表す関数**p_n

(23) **関数**

$l(a)=a$**の素因数分解における**0**でない数の個数**

$(a \neq 0$**のとき**$)$

$l(a)=0$ $(a=0$**のとき**$)$

(24) a**と**i**の関数**

$(a)_i=a$**の素因数分解における**p_i**のべき数** $(a \neq 0$**のとき**$)$

$(a)_i=0$ $(a=0$**のとき**$)$

原始帰納的述語と関数の例 5

(25) **関数**

$$x \circ y = x \bullet \prod_{i<l(y)} p_{l(x)+i}^{(y)_i}$$

部分帰納的関数と帰納的関数

部分帰納的(partial recursive)**関数**

初期関数(1), (2), (3)**に操作**(I), (II), <u>**(III)**</u>**を有限回適用して定義された関数。**

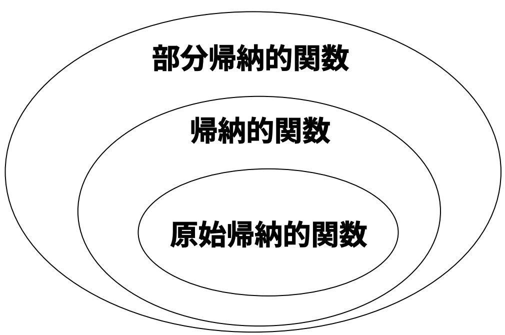
新操作

帰納的(recursive)**関数**

(**一般帰納的**(general recursive)**関数**)

初期関数(1), (2), (3)**に操作**(I), (II), <u>**(III´)**</u>**を有限回適用して定義された関数。**

各関数の関係



2つの新操作

(III) **全域的であることが保証されない。**

関数 $g(x_1, \ldots, x_n, y)$ **から関数** $f(x_1, \ldots, x_n)$ **を以下の操作で作る。**

$$f(x_1, \ldots, x_n) = \mu y(g(x_1, \ldots, x_n, y)=0)$$

(III´) **全域的であることが保証される。**

<u>**正則**</u>**関数** $g(x_1, \ldots, x_n, y)$ **から関数** $f(x_1, \ldots, x_n)$ **を以下の操作で作る。**

$$f(x_1, \ldots, x_n) = \mu y(g(x_1, \ldots, x_n, y)=0)$$

μ作用素
(μ-operator)

n+1変数の述語からn変数の関数を作る操作

定義

述語$P(x_1, \dots, x_n, y)$に対して

$\mu y P(x_1, \dots, x_n, y)$

$= \min\{y \mid P(x_1, \dots, x_n, y)\}$

($(\exists y) P(x_1, \dots, x_n, y)$のとき)

$\mu y P(x_1, \dots, x_n, y)$

=無定義

($\neg (\exists y) P(x_1, \dots, x_n, y)$のとき)

正則とは

述語$P(x_1, \dots, x_n, y)$が正則

＝ 任意の$(x_1, \dots, x_n)$に対して$P(x_1, \dots, x_n, y)$を真とするyが存在する。

関数$g(x_1, \dots, x_n, y)$が正則

＝任意の$(x_1, \dots, x_n)$に対して$g(x_1, \dots, x_n, y)=0$となるyが存在する。

最後に

- レポートを出します
- ミニテストを提出して帰ること
- 次回は、
  - Turing機械