計算の理論 II
帰納的関数

月曜4校時
大月美佳

講義の前に

- 来週はJABEE審査

今日の講義内容

1. <u>原始帰納的関数</u>
   1. <u>初期関数</u>
   2. <u>合成と原始帰納</u>
2. 原始帰納的でない関数
   1. Ackermann関数

原始帰納的関数

- 計算可能な関数の一部
- 原始帰納的関数

ー拡大→帰納的関数＝計算できる関数の族
帰納的関数＝Turing機械で受理できる言語

帰納的関数＝計算可能

原始帰納的関数

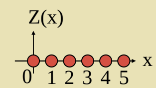
数論的関数

- **自然数 (非負整数)** N
  N={0, 1, 2, 3, …}
- **数論的関数(関数)**
  N個の自然数の組に対して、
  高々1個の自然数を対応づける関数
  $f: N^n \rightarrow N$

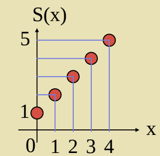
初期関数

- **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?

$f(x_1, \ldots, x_n)$
$=h(x_1, \ldots, x_{n-1}, \underline{x_n-1}, f(x_1, \ldots, \underline{x_n-1}))$
$=h(x_1, \ldots, x_{n-1}, \underline{x_n-1}, h(x_1, \ldots, x_{n-1}, \underline{x_n-2}, f(x_1, \ldots, \underline{x_n-2})))$
…
$=h(x_1, \ldots, x_{n-1}, \underline{x_n-1}, h(x_1, \ldots, x_{n-1}, \underline{x_n-2}, h(\ldots \underline{g(x_1, \ldots, x_{n-1})} ..)))$

0
x_n-3 …
x_n-2
x_n-1

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。

原始帰納的関数の例 (4)

(4) **定数関数** $C_n^k(x_1, \ldots, x_i, \ldots, x_n)=k$

なぜならば

$C_n^k(x_1, \ldots, x_i, \ldots, x_n)$

$=\boxed{S(S(\ldots S(}Z(U_n^i(x_1, \ldots, x_n)))\ldots))=k$

xを1個取り出す
(どれでも良い)

0にk回1を加算

選ばれたxを0にする

$C_5^3(x_1, x_2, x_3, x_4, x_5)=S(S(S(Z(U_5^4(x_1, x_2, x_3, x_4, x_5)))))$

$=S(S(S(Z(x_4))))=S(S(S(0)))=S(S(1))=S(2)=3$

原始帰納的関数の例 (5)

(5) x_1+x_2

$plus(x_1, x_2)= x_1+x_2$**とおくと、**

$plus(x_1, x_2)= U_1^1(x_1)$　($x_2=0$**のとき**)

$plus(x_1, x_2)=S(U_3^3(x_1, x_2-1, plus(x_1, x_2-1)))$

($x_2>0$**のとき**)

$plus(2, 4)=S(U_3^3(2, 3, plus(2, 3)))=S(plus(2, 3))$

$=S(S(U_3^3(2, 2, plus(2, 2))))=S(S(plus(2, 2)))$

帰納的 $=S(S(S(U_3^3(2, 1, plus(2, 1)))))=S(S(S(plus(2, 1))))$

$=S(S(S(S(U_3^3(2, 0, plus(2, 0))))))=S(S(S(S(plus(2, 0)))))$

$=S(S(S(S(\boxed{U_1^1(2)})))) \leftarrow g(x_1)$

$=S(S(S(S(2))))=S(S(S(3)))=S(S(4))=S(5)=6$

原始帰納的関数の例 (6)

(6) $x_1 \cdot x_2$

$times(x_1, x_2)= x_1 \cdot x_2$**とおくと、**

$times(x_1, x_2)=Z(x_1)=0$　　($x_0=0$**のとき**)

$times(x_1, x_2)=p(x_1, x_2-1, times(x_1, x_2-1))$

($x_2>0$**のとき**)

ここで、

$p(x, y, z)= plus(U_3^1(\boxed{x}, y, z), U_3^3(x, y, \boxed{z}))$

$=x+z$

(6)´の計算例

times(3, 4)=p(3, 3, times(3, 3))=3+times(3, 3)
=3+p(3, 2, times(3, 2))=3+3+times(3, 2)
=3+3+p(3, 1, times(3, 1))=3+3+3+times(3, 1)
=3+3+3+p(3, 0, times(3, 0))=3+3+3+3+times(3, 0)
=3+3+3+3+$\boxed{Z(3)}$=3+3+3+3+0=12

原始帰納的関数の例 (7)

(7) x^y

power(x_1, x_2)= $x_1^{x_2}$とおくと、 g(x_1)

power(x_1, x_2)=S(Z(x_1))=1 (x_2=0のとき)

power(x_1, x_2)=p(x_1, x_2-1, power(x_1, x_2-1))

= x_1・power(x_1, x_2-1) (x_2>0のとき)

h(x_1, x_2-1, f(x_1, x_2-1))

ここで、

p(x, y, z)= times(U_3^1($\boxed{x}$, y, z), U_3^3(x, y, $\boxed{z}$))

=x・z

原始帰納的関数の例 (8)

(8) x_1!

factorial(x_1)= x!とおくと、 g(k)

factorial(x_1)=S(Z(k))=1 (x_1=0のとき)

factorial(x_1)=p(x_1-1, factorial(x_1-1))

= x_1・factorial(x_1-1) (x_1>0のとき)

ここで、 h(x_1-1, f(x_1-1))

p(x, y)= times(S(U_2^1($\boxed{x}$, y)), U_2^2(x, $\boxed{y}$))

=(x+1)・y

原始帰納的関数の例 (9)

(9) pd(x_1)を

pd(x_1)=0 (x_1=0のとき)

pd(x_1)= x_1-1 (x_1>0のとき)

とおくと、 g(k)

pd(x_1)=Z(k)=0 (x_1=0のとき)

pd(x_1)=p(x_1-1, pd(x_1-1))=x_1-1 (x_1>0のとき)

ここで、 h(x_1-1, f(x_1-1))

p(x, y)= U_2^1($\boxed{x}$, y)=x

原始帰納的関数の例 (10)

(10)自然数上での減算$x_1 \dot{-} x_2$を

$x_1 \dot{-} x_2 = x_1 - x_2$ ($x_1 \geqq x_2$のとき)

$x_1 \dot{-} x_2 = 0$ ($x_1 < x_2$のとき)

とする。$x_1 \dot{-} x_2$ =n-minus(x_1, x_2)とおくと、

n-minus(x_1, x_2)= $U_1^1(x_1)$ (x_2=0のとき)

n-minus(x_1, x_2)=p(x_1, x_2-1, n-minus(x_1, x_2-1))

g(x_1) =pd(n-minus(x_1, x_2-1)) ($x_2 > 0$のとき)

ここで、 h(x_1, x_2-1, f(x_1, x_2-1))

p(x, y, z)= pd(U_3^3(x, y, $\boxed{z}$))=pd(z)

原始帰納的関数の例 (11)

(11) 差の絶対値$|x_1 - x_2|$を

$|x_1 - x_2| = x_1 - x_2$ ($x_1 \geqq x_2$のとき)

$|x_1 - x_2| = x_2 - x_1$ ($x_1 < x_2$のとき)

とする。

$|x_1 - x_2|$ =abs-minus(x_1, x_2)

=plus(n-minus(x_1, x_2), n-minus (x_2, x_1))

$= x_1 \dot{-} x_2 + x_2 \dot{-} x_1$

原始帰納的関数の例 (12)

(12) xの符号を表す関数

sg(x_1)=0 (x_1=0のとき)

sg(x_1)=1 ($x_1 > 0$のとき)

とすると、

sg(x_1)=Z(k)=0 (x_1=0のとき)

sg(x_1)=S(Z(U_2^1 (x_1-1, sd(x_1-1))) ($x_1 > 0$のとき)

原始帰納的関数その他 1

関数f(x_1, …, x_n, y)が原始帰納的であれば、有限和も有限積も原始帰納的である。

$$\sum_{y<0} f(x_1,\ldots,x_n,y) = 0$$

$$\sum_{y<z} f(x_1,\ldots,x_n,y) = f(x_1,\ldots,x_n,0) + \ldots + f(x_1,\ldots,x_n,z-1)$$

$$\prod_{y<0} f(x_1,\ldots,x_n,y) = 0$$

$$\prod_{y<z} f(x_1,\ldots,x_n,y) = f(x_1,\ldots,x_n,0) + \ldots + f(x_1,\ldots,x_n,z-1)$$

原始帰納的関数その他 2

関数f(x1, … , xn , y)が原始帰納的であれば、以下も原始帰納的である。

$$\sum_{y\le z} f(x_1,\dots,x_n,y),\ \prod_{y\le z} f(x_1,\dots,x_n,y),$$

$$\sum_{u<y<z} f(x_1,\dots,x_n,y),\ \prod_{u<y<z} f(x_1,\dots,x_n,y),$$

$$\sum_{u\le y\le z} f(x_1,\dots,x_n,y),\ \prod_{u\le y\le z} f(x_1,\dots,x_n,y),$$

原始帰納的でない関数

計算可能だが原始帰納的ではない関数

F(x, n)

任意の1変数の原始帰納的関数f(x)に対して

f(x)=F(n, x)

となる自然数が存在するような関数。

→証明

Ackerman関数

原始帰納的でない関数

A(x, y)

1. A(0, y) = y+1
2. A(x+1, 0) = A(x, 1)
3. A(x+1, y+1) = A(x, A(x+1, y))

計算に手間のかかる関数

→例 A(2, 2)

最後に

- レポートを提出してから帰ること
- 次回は、
  - 帰納関数つづき