



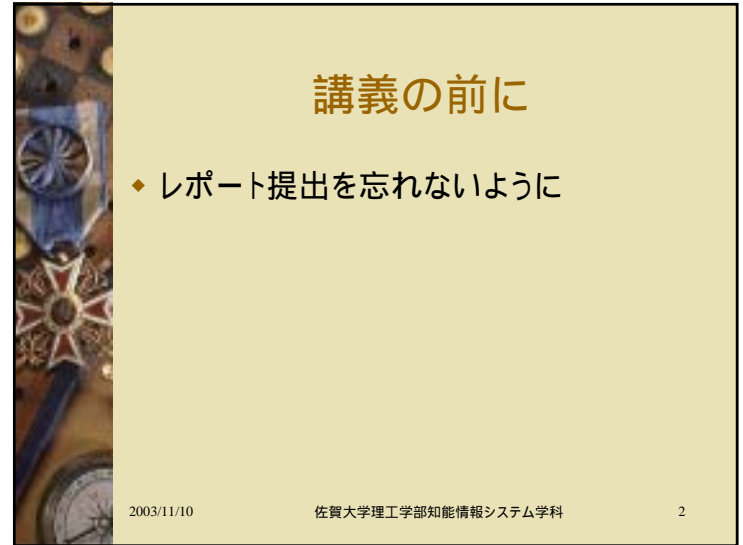
計算の理論 II 帰納的関数3

月曜4校時
大月美佳

2003/11/10

佐賀大学工学部知情報システム学科

1



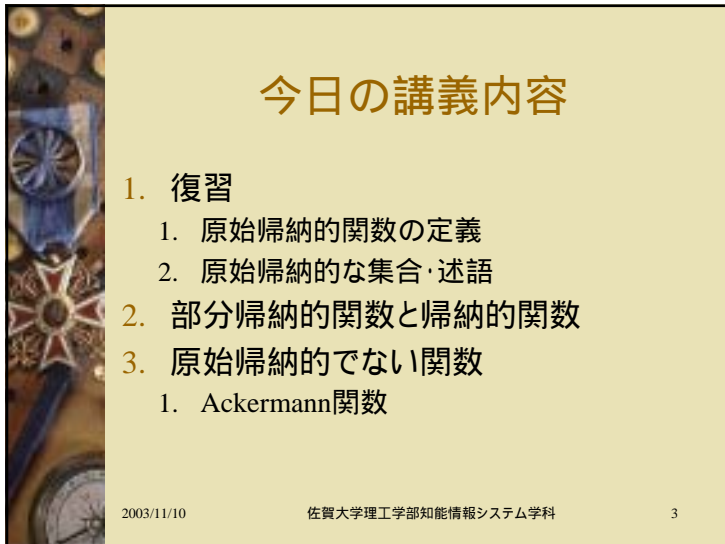
講義の前に

- ◆ レポート提出を忘れないように

2003/11/10

佐賀大学工学部知情報システム学科

2



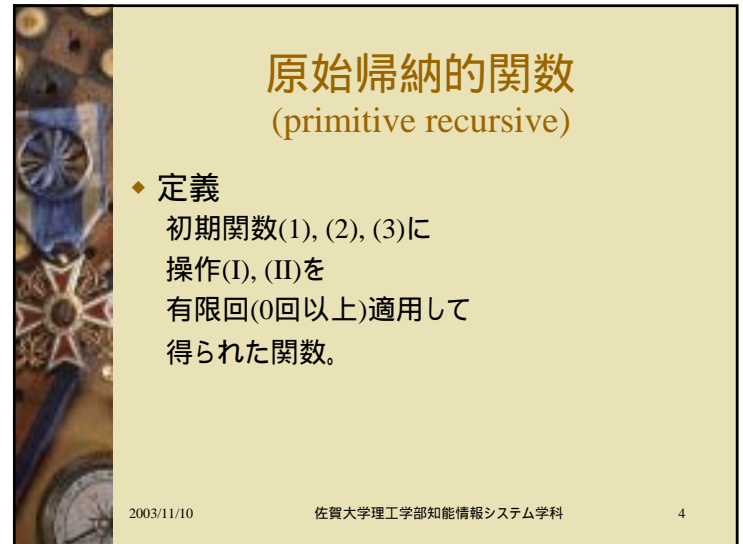
今日の講義内容

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. 部分帰納的関数と帰納的関数
3. 原始帰納的でない関数
 1. Ackermann関数

2003/11/10

佐賀大学工学部知情報システム学科

3



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。

2003/11/10

佐賀大学工学部知情報システム学科

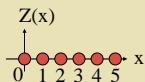
4

初期関数

◆ 原始帰納的関数の素。

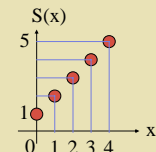
(1) $Z(x)=0$

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) $U_n^i(x_1, \dots, x_i, \dots, x_n)=x_i$
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。

2003/11/10

佐賀大学理工学部知情報システム学科

5

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 $g_i(1 \leq i \leq r)$ から、
n変数の関数fを以下の操作で作ること。

$$f(x_1, \dots, x_n) = h(g_1(x_1, \dots, x_n), \dots, g_r(x_1, \dots, x_n))$$

(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。

$$f(x_1, \dots, x_n) = g(x_1, \dots, x_{n-1}) \quad (x_n = 0 \text{ のとき})$$

$$f(x_1, \dots, x_n) = h(x_1, \dots, x_{n-1}, x_{n-1}, f(x_1, \dots, x_{n-1})) \quad (x_n > 0 \text{ のとき})$$

2003/11/10

佐賀大学理工学部知情報システム学科

6

原始帰納的な集合と述語

特徴関数 C_S, C_P

集合 $S \subseteq \mathbb{N}^n$

$$C_S(x_1, \dots, x_n) = 0 \quad ((x_1, \dots, x_n) \in S \text{ のとき})$$

$$C_S(x_1, \dots, x_n) = 1 \quad ((x_1, \dots, x_n) \notin S \text{ のとき})$$

が原始帰納的であるとき、Sは原始帰納的集合。

述語 $P(x_1, \dots, x_n)$

$$C_P(x_1, \dots, x_n) = 0 \quad (P(x_1, \dots, x_n) \text{ のとき})$$

$$C_P(x_1, \dots, x_n) = 1 \quad (\neg P(x_1, \dots, x_n) \text{ のとき})$$

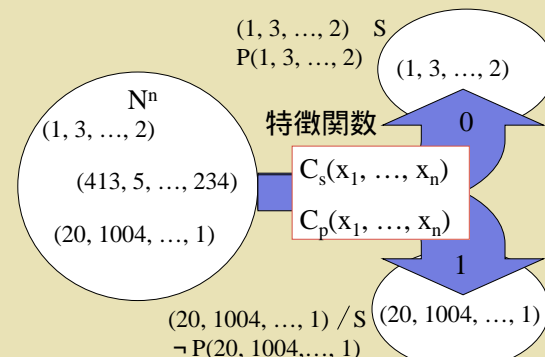
が原始帰納的であるとき、Pは原始帰納的述語。

2003/11/10

佐賀大学理工学部知情報システム学科

7

イメージ図



2003/11/10

佐賀大学理工学部知情報システム学科

8

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), (III)を有限回適用して定義された関数。

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), (III')を有限回適用して定義された関数。

新操作

2003/11/10

佐賀大学理工学部知能情報システム学科

9

2つの新操作

(III) 全域的であることが保証されない。
関数 $g(x_1, \dots, x_n, y)$ から関数 $f(x_1, \dots, x_n)$ を以下の操作で作る。

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

(III') 全域的であることが保証される。
正則関数 $g(x_1, \dots, x_n, y)$ から関数 $f(x_1, \dots, x_n)$ を以下の操作で作る。

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

2003/11/10

佐賀大学理工学部知能情報システム学科

10

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

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

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

((y) $P(x_1, \dots, x_n, y)$ のとき)

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

=無定義

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

2003/11/10

佐賀大学理工学部知能情報システム学科

11

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

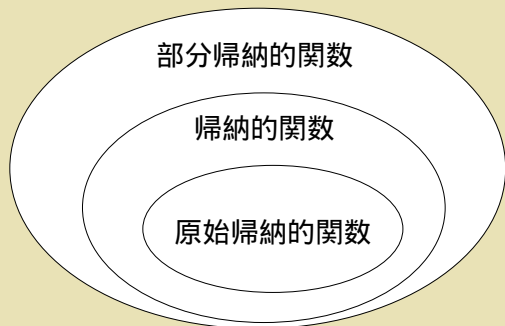
= 任意の (x_1, \dots, x_n) に対して $g(x_1, \dots, x_n, y) = 0$ となる y が存在する。

2003/11/10

佐賀大学理工学部知能情報システム学科

12

各関数の関係



2003/11/10

佐賀大学工学部知能情報システム学科

13

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。

2003/11/10

佐賀大学工学部知能情報システム学科

14

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 小レポート

類似品: たらいまわし関数

2003/11/10

佐賀大学工学部知能情報システム学科

15

計算量と数の発散

(x, y)	結果	再帰計算回数
(3, 1)	13	106
(3, 2)	29	541
(3, 3)	61	2432
(3, 4)	125	10307
(3, 5)	253	42438
(3, 6)	509	172233
(3, 7)	1021	693964

$$A(0, y) = y + 1$$

$$A(1, y) = y + 2$$

$$A(2, y) = 2y + 3$$

$$A(3, y) = 2^{y+3} - 3$$

$$A(4, y) = \underbrace{2^{2^{\cdot^{\cdot^2}}}}_{y+3} - 3$$

(2の2乗を(y+3)繰り返したものを引く)

2003/11/10

佐賀大学工学部知能情報システム学科

16

原始帰納的でないことの証明

原始帰納的と仮定 矛盾 (背理法)

$A(x, y)$ を原始帰納的と仮定すると、

$A(x, x)$ は1変数の原始帰納関数。

次ページの定理より、

$$A(x, x) \quad A(c, \max(x))=A(c, x)$$

となる c が存在する。ここで $x=c+1$ とおくと、

$$A(c+1, c+1) \quad A(c, c+1)$$

となり、次ページの補題3に反する。

ゆえに、 $A(x, y)$ は原始帰納的ではない。

証明の基礎

補題:

1. $A(x+1, y) \quad y+2$
2. $A(x, y) \quad y+1$
3. $A(x, y) < A(x, y+1) \quad A(x+1, y)$
4. 任意の c_1, c_2 に対して $A(c_1, A(c_2, x)) \quad A(c_3, x)$ となる c_3 が存在する

定理:

任意の原始帰納関数 $f(x_1, \dots, x_n)$ に対して

$$f(x_1, \dots, x_n) \quad A(c, \max(x_1, \dots, x_n))$$

となる c が存在する。

最後に

- ◆ レポートを提出してから帰ること
- ◆ 次回は、
 - Turing機械