

## 5 原始帰納的関数

### [I] “図形” としての自然数

- (i) 0 は自然数である。
- (ii)  $n$  が自然数ならば、 $n'$  も自然数である。
- (iii) (i),(ii) によって定められるもののみが自然数である。

$n'$  を  $n$  の後者という。通常は  $0'$  を 1 と書き、 $0''$  を 2 と書き、 $\dots$  と約束するが、これらは略記号であり、“図形” としての自然数はあくまで、 $0, 0', 0'', 0''', \dots$  なる形をしていると考える。

### [II] 関数の帰納的定義と “図形” としての計算

- 和の定義:

$$\begin{cases} 0 + y = y \\ x' + y = (x + y)' \end{cases}$$

たとえば、

$$0''' + 0'' = (0'' + 0'')' = (0' + 0'')'' = (0 + 0'')''' = 0''''$$

は  $3 + 2 = 5$  を意味する。

- 積の定義:

$$\begin{cases} 0 \cdot y = 0 \\ x' \cdot y = x \cdot y + y \end{cases}$$

$2 \cdot 3 = 6$  は、

$$\begin{aligned} 0'' \cdot 0''' &= 0' \cdot 0''' + 0''' = (0 \cdot 0''' + 0''') + 0''' = 0''' + 0''' \\ &= (0'' + 0''')' = (0' + 0''')'' = (0 + 0''')''' = 0'''' \end{aligned}$$

と計算される。

- べきの定義:

$$\begin{cases} y^0 = 1 \\ y^{x'} = y^x \cdot y \end{cases}$$

$2^3$  の計算は

$$0''^{0'''} = 0''^{0''} \cdot 0'' = (0''^{0'} \cdot 0'') \cdot 0'' = ((0''^0 \cdot 0'') \cdot 0'') \cdot 0'' = ((0' \cdot 0'') \cdot 0'') \cdot 0''$$

によって、積の計算に帰着される。

## [III] 原始帰納的関数

## ● 基本関数

- 零関数:  $N(x) = 0$
- 後者関数:  $S(x) = x'$
- 射影関数:  $P_i^n(x_1, \dots, x_n) = x_i \quad (i = 1, \dots, n)$

● 帰納的定義 関数  $g$  と  $h$  が与えられたとき、

$$\begin{cases} f(0, x_1, \dots, x_n) = g(0, x_1, \dots, x_n) \\ f(x', x_1, \dots, x_n) = h(f(x, x_1, \dots, x_n), x, x_1, \dots, x_n) \end{cases}$$

によって定まる関数  $f$  を  $g, h$  から帰納的に定義された関数という。

## ● 定義 基本関数から出発して、合成 (代入) または帰納的定義を有限回ほどこして得られる関数を、原始帰納的関数という。

## ● 前ページの和、積、べきは原始帰納的関数である。

● 定数関数 固定された自然数  $n$  に対する定数関数:  $C_n(x) = n$  は

$$C_n(x) = \underbrace{S(S(\dots S(N(x)) \dots))}_n$$

のように  $N, S$  の合成によって得られるので、原始帰納的である。

● 恒等関数  $I(x) = x$  は、1変数の射影関数  $P_1^1(x)$  と等しいので原始帰納的である。また、

$$\begin{cases} I(0) = 0 \\ I(x') = S(I(x)) \end{cases}$$

なる帰納的定義によっても得られる。

## [IV] 原始帰納的関数の例

- 直前の数;

$$\begin{cases} \text{prev}(0) = 0 \\ \text{prev}(x') = x \end{cases}$$

- 半差、すなわち  $x > y$  のとき  $x - y$ 、 $x \leq y$  のとき 0;

$$\begin{cases} x \smile 0 = x \\ x \smile y' = \text{prev}(x \smile y) \end{cases}$$

- 差;

$$|x - y| = (x \smile y) + (y \smile x)$$

- 零判定関数;

$$\text{sg}(x) = 1 \smile (1 \smile x)$$

$$\overline{\text{sg}}(x) = 1 \smile x$$

または、帰納的定義で

$$\begin{cases} \text{sg}(0) = 0, & \overline{\text{sg}}(0) = 1 \\ \text{sg}(x') = 1, & \overline{\text{sg}}(x') = 0 \end{cases}$$

- 剰余、 $x$  を  $y$  で割ったときの余り;

$$\begin{cases} \text{rem}(0, y) = 0 \\ \text{rem}(x', y) = (\text{rem}(x, y) + 1) \cdot \text{sg}(y \smile (\text{rem}(x, y) + 1)) \\ \quad \quad \quad + (x + 1) \cdot \overline{\text{sg}}(y) \end{cases}$$

通常  $y = 0$  のときは定義されないが、ここでは  $\text{rem}(x, 0) = x$  とする。

- $x$  を  $y$  で割ったときの商;

$$\begin{cases} \text{quo}(0, y) = 0 \\ \text{quo}(x', y) = \text{quo}(x, y) + \overline{\text{sg}}(\text{rem}(x + 1, y)) \end{cases}$$

- $y$  より小さい  $x$  の約数の個数;

$$\begin{cases} \text{dnum}(x, 0) = 0 \\ \text{dnum}(x, y') = \text{dnum}(x, y) + \overline{\text{sg}}(\text{rem}(x, y)) \end{cases}$$

- 素数の判定;

$$\text{Prime}(x) = \text{sg}(|\text{dnum}(x, x) - 1|)$$

$x$  が素数のとき 0 で、そうでないとき 1 となる。

## [V] 総和・総積による定義

- 定理 5-1  $f(x, x_1, \dots, x_n)$  が原始帰納的関数ならば、

$$\text{Sum}_f(x, x_1, \dots, x_n) = \sum_{y=0}^x f(y, x_1, \dots, x_n)$$

$$\text{Prod}_f(x, x_1, \dots, x_n) = \prod_{y=0}^x f(y, x_1, \dots, x_n)$$

なる関数  $\text{Sum}_f$  および  $\text{Prod}_f$  は原始帰納的関数である。

[証明]  $\text{Sum}_f$  は帰納的に

$$\begin{cases} \text{Sum}_f(0, x_1, \dots, x_n) = f(0, x_1, \dots, x_n) \\ \text{Sum}_f(x', x_1, \dots, x_n) = \text{Sum}_f(x, x_1, \dots, x_n) + f(x', x_1, \dots, x_n) \end{cases}$$

と定義され、 $\text{Prod}_f$  も帰納的に

$$\begin{cases} \text{Prod}_f(0, x_1, \dots, x_n) = f(0, x_1, \dots, x_n) \\ \text{Prod}_f(x', x_1, \dots, x_n) = \text{Prod}_f(x, x_1, \dots, x_n) \cdot f(x', x_1, \dots, x_n) \end{cases}$$

と定義されるので原始帰納的関数である。[証明終]

- 商  $\text{quo}(x, y)$  は  $x - y, x - 2y, x - 3y, \dots$  が非負となる個数、すなわち、 $x + 1 - y, x + 1 - 2y, x + 1 - 3y, \dots$  が正となる個数だから、

$$\text{quo}(x, y) = \sum_{z=1}^x \text{sg}((x+1) \smile zy)$$

とも定義できる。

- $x$  を割る最大の  $p$  べき指数:

$$\nu(p, x) = \sum_{y=1}^x \overline{\text{sg}}(\text{rem}(x, p^y))$$

- $x$  の素因数の積:

$$\text{spf}(x) = \prod_{p=2}^x p^{\text{sg}(\nu(p, x)) \cdot \overline{\text{sg}}(\text{Prime}(p))}$$