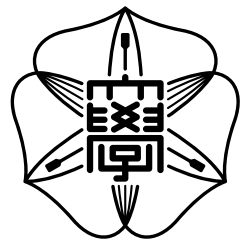
計算理論

Thomas Zeugmann

北海道大学 大学院情報科学研究科
アルゴリズム研究室

http://www-alg.ist.hokudai.ac.jp/~thomas/ToC/

第 11 回：計算モデル
(日本語訳：大久保 好章，湊 真一)



動機 I

およそ 100 年前，数学は，それを解くアルゴリズムを誰も見つけることができない問題があるという難問に直面した.

動機 I

およそ 100 年前，数学は，それを解くアルゴリズムを誰も見つけることができない問題があるという難問に直面した．

有名な例としてヒルベルトの第 10 問題があり，次の通り定式化されている．

与えられたディオファントス方程式が整数解を持つか否かを決定するアルゴリズムを設計せよ．

動機 II

ついに，計算論的に解くことができない問題が存在するかもしれないことが明らかになった．

動機 II

ついに，計算論的に解くことができない問題が存在するかもしれないことが明らかになった．

まず最初に，こうした考えがどのように明らかになったかをみよう．実際は，カントールの研究なしにはそれはずっと難しかったでろう．

動機 III

我々が知っているすべてのアルゴリムを見ると，アルゴリズムとは，次の性質を有する計算の方法論であると言える．

(1) 命令は有限のテキストである．

(2) 計算は段階的に行なわれる．ここで，各段階 (ステップ) においては基本演算が実行される．

(3) 計算の実行の各ステップにおいて，実行すべき基本演算は一意に決まっている．

(4) 次の計算ステップは，入力とそれまでに計算された途中結果のみに依存する．

動機 III

我々が知っているすべてのアルゴリムを見ると，アルゴリズムとは，次の性質を有する計算の方法論であると言える．

(1) 命令は有限のテキストである．

(2) 計算は段階的に行なわれる．ここで，各段階 (ステップ) においては基本演算が実行される．

(3) 計算の実行の各ステップにおいて，実行すべき基本演算は一意に決まっている．

(4) 次の計算ステップは，入力とそれまでに計算された途中結果のみに依存する．

動機 III

我々が知っているすべてのアルゴリムを見ると，アルゴリズムとは，次の性質を有する計算の方法論であると言える．

(1) 命令は有限のテキストである．

(2) 計算は段階的に行なわれる．ここで，各段階 (ステップ) においては基本演算が実行される．

(3) 計算の実行の各ステップにおいて，実行すべき基本演算は一意に決まっている．

(4) 次の計算ステップは，入力とそれまでに計算された途中結果のみに依存する．

動機 III

我々が知っているすべてのアルゴリムを見ると，アルゴリズムとは，次の性質を有する計算の方法論であると言える．

(1) 命令は有限のテキストである．

(2) 計算は段階的に行なわれる．ここで，各段階 (ステップ) においては基本演算が実行される．

(3) 計算の実行の各ステップにおいて，実行すべき基本演算は一意に決まっている．

(4) 次の計算ステップは，入力とそれまでに計算された途中結果のみに依存する．

動機 III

我々が知っているすべてのアルゴリムを見ると，アルゴリズムとは，次の性質を有する計算の方法論であると言える．

(1) 命令は有限のテキストである．

(2) 計算は段階的に行なわれる．ここで，各段階 (ステップ) においては基本演算が実行される．

(3) 計算の実行の各ステップにおいて，実行すべき基本演算は一意に決まっている．

(4) 次の計算ステップは，入力とそれまでに計算された途中結果のみに依存する．

動機 IV

いま，有限のアルファベット Σ が存在し，すべてのアルゴリズムは Σ^* 中の文字列として表現可能であると仮定する．Σ^* 中のすべての文字列の総数は**加算無限** *(countably infinite)* なので，アルゴリズムの総数も高々 **加算無限**個である．

動機 IV

いま，有限のアルファベット Σ が存在し，すべてのアルゴリズムは Σ^* 中の文字列として表現可能であると仮定する．Σ^* 中のすべての文字列の総数は**加算無限** *(countably infinite)* なので，アルゴリズムの総数も高々 **加算無限**個である．

カントールの次の有名な結果，すなわち，

$$\{f \mid f\colon \mathbb{N} \to \{0,1\}\}$$

は**非加算無限** *(uncountably infinite)* であることから，直ちに次の定理に辿り着く．

動機 IV

いま，有限のアルファベット Σ が存在し，すべてのアルゴリズムは Σ^* 中の文字列として表現可能であると仮定する．Σ^* 中のすべての文字列の総数は**加算無限** *(countably infinite)* なので，アルゴリズムの総数も高々 **加算無限**個である．

カントールの次の有名な結果，すなわち，

$$\{f \mid f\colon \mathbb{N} \to \{0,1\}\}$$

は**非加算無限** *(uncountably infinite)* であることから，直ちに次の定理に辿り着く．

定理 1

計算不可能な関数 $f\colon \mathbb{N} \to \{0,1\}$ が存在する．

動機 V

この結果は基本的・認識論的に重要であるが，個々の関数の性質については何も言及していない．こうした視点からの結果を得るには，さらに多くのすべきことがある．よって，現代の計算理論は，次の疑問から出発する：

いかなる問題が計算論的に解けるのか？

動機 V

この結果は基本的・認識論的に重要であるが，個々の関数の性質については何も言及していない．こうした視点からの結果を得るには，さらに多くのすべきことがある．よって，現代の計算理論は，次の疑問から出発する：

いかなる問題が計算論的に解けるのか？

これに答えるためには，まずは，アルゴリズムに対する直観的な概念を数学的に定式化しなければならない．

動機 V

この結果は基本的・認識論的に重要であるが，個々の関数の性質については何も言及していない．こうした視点からの結果を得るには，さらに多くのすべきことがある．よって，現代の計算理論は，次の疑問から出発する：

いかなる問題が計算論的に解けるのか？

これに答えるためには，まずは，アルゴリズムに対する直観的な概念を数学的に定式化しなければならない．

この講義では，ゲーデルとチューリングの定式化，すなわち，部分帰納的関数とチューリング機械についてそれぞれ学ぶ．

部分帰納的関数の定義 I

任意の $n \in \mathbb{N}^+$ について，$\mathbb{N}^n$ から $\mathbb{N}$ へのすべての帰納的関数 (recursive function) の集合を $\mathcal{P}^n$ と書く．ここで，$\mathbb{N}^1 = \mathbb{N}$ および $\mathbb{N}^{n+1} = \mathbb{N}^n \times \mathbb{N}$ と定義する．つまり，$\mathbb{N}^n$ は，順序付きの自然数の n-組の集合である．
すべての部分帰納的関数 (partial recursive function) の集合 $\mathcal{P}$ を定義するためのゲーデルのアイデアは次の通りである．

手順 (1): 直観的に計算可能な基本関数をいくつか定義する．

手順 (2): 計算可能であることが既に分かっている関数をもとに，新たな計算可能な関数を構成するためのいくつかの規則を定義する．

部分帰納的関数の定義 I

任意の $n \in \mathbb{N}^+$ について，$\mathbb{N}^n$ から $\mathbb{N}$ へのすべての帰納的関数 (recursive function) の集合を $\mathcal{P}^n$ と書く．ここで，$\mathbb{N}^1 = \mathbb{N}$ および $\mathbb{N}^{n+1} = \mathbb{N}^n \times \mathbb{N}$ と定義する．つまり，$\mathbb{N}^n$ は，順序付きの自然数の n-組の集合である．
すべての部分帰納的関数 (partial recursive function) の集合 $\mathcal{P}$ を定義するためのゲーデルのアイデアは次の通りである．

手順 (1): 直観的に計算可能な基本関数をいくつか定義する．

手順 (2): 計算可能であることが既に分かっている関数をもとに，新たな計算可能な関数を構成するためのいくつかの規則を定義する．

部分帰納的関数の定義 I

任意の $n \in \mathbb{N}^+$ について，$\mathbb{N}^n$ から $\mathbb{N}$ へのすべての帰納的関数 (recursive function) の集合を $\mathcal{P}^n$ と書く．ここで，$\mathbb{N}^1 = \mathbb{N}$ および $\mathbb{N}^{n+1} = \mathbb{N}^n \times \mathbb{N}$ と定義する．つまり，$\mathbb{N}^n$ は，順序付きの自然数の n-組の集合である．
すべての部分帰納的関数 (partial recursive function) の集合 $\mathcal{P}$ を定義するためのゲーデルのアイデアは次の通りである．

手順 (1): 直観的に計算可能な基本関数をいくつか定義する．

手順 (2): 計算可能であることが既に分かっている関数をもとに，新たな計算可能な関数を構成するためのいくつかの規則を定義する．

部分帰納的関数の定義 II

手順 (1) に関して，次の関数 Z, S, $V\colon \mathbb{N} \to \mathbb{N}$ を定義する．

$$
\begin{aligned}
Z(n) &= 0 \quad \text{for all } n \in \mathbb{N}, \\
S(n) &= n+1 \quad \text{for all } n \in \mathbb{N}, \\
V(n) &= \begin{cases} 0, & \text{if } n = 0\,; \\ n-1, & \text{for all } n \geqslant 1\,. \end{cases}
\end{aligned}
$$

すなわち，Z は定数 0 関数 (constant 0 function)，S は後者関数 (successor function)，V は前者関数 (predecessor function) である．らかに，これら関数は直観的に計算可能である．よって，定義より，Z, S, $V \in \mathcal{P}^1$ となる．これで手順 (1) は完了である．

部分帰納的関数の定義 III

次に，規則を定義する (手順 (2) を参照).

部分帰納的関数の定義 III

次に，規則を定義する (手順 (2) を参照).

(2.1) (架空変数の導入)
$n \in \mathbb{N}^+$ とする．$\tau \in \mathcal{P}^n$ かつ
$\psi(x_1, \ldots, x_n, x_{n+1}) =_{df} \tau(x_1, \ldots, x_n)$ ならば，$\psi \in \mathcal{P}^{n+1}$ である.

部分帰納的関数の定義 III

次に，規則を定義する (手順 (2) を参照).

(2.1) (架空変数の導入)
$n \in \mathbb{N}^+$ とする．$\tau \in \mathcal{P}^n$ かつ
$\psi(x_1, \ldots, x_n, x_{n+1}) =_{df} \tau(x_1, \ldots, x_n)$ ならば，$\psi \in \mathcal{P}^{n+1}$ である.

(2.2) (変数の同一視)
$n \in \mathbb{N}^+$ とする．$\tau \in \mathcal{P}^{n+1}$ かつ
$\psi(x_1, \ldots, x_n) =_{df} \tau(x_1, \ldots, x_n, x_n)$ ならば，$\psi \in \mathcal{P}^n$ である.

部分帰納的関数の定義 III

次に，規則を定義する (手順 (2) を参照).

(2.1) (架空変数の導入)
$n \in \mathbb{N}^+$ とする．$\tau \in \mathcal{P}^n$ かつ
$\psi(x_1, \ldots, x_n, x_{n+1}) =_{df} \tau(x_1, \ldots, x_n)$ ならば，$\psi \in \mathcal{P}^{n+1}$ である．

(2.2) (変数の同一視)
$n \in \mathbb{N}^+$ とする．$\tau \in \mathcal{P}^{n+1}$ かつ
$\psi(x_1, \ldots, x_n) =_{df} \tau(x_1, \ldots, x_n, x_n)$ ならば，$\psi \in \mathcal{P}^n$ である．

(2.3) (変数の置換)
$n \in \mathbb{N}^+$, $n \geqslant 2$ および $i \in \{1, \ldots, n\}$ とする．$\tau \in \mathcal{P}^n$ かつ
$\psi(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n) =_{df} \tau(x_1, \ldots, x_{i+1}, x_i, \ldots, x_n)$ ならば，$\psi \in \mathcal{P}^n$ である．

部分帰納的関数の定義 IV

(2.4) (合成)

$n \in \mathbb{N}$ および $m \in \mathbb{N}^+$ とする．さらに，$\tau \in \mathcal{P}^{n+1}$，$\psi \in \mathcal{P}^m$ とし，

$\phi(x_1,\ldots,x_n,y_1,\ldots,y_m) =_{df} \tau(x_1,\ldots,x_n,\psi(y_1,\ldots,y_m))$

と定義する．

このとき，$\phi \in \mathcal{P}^{n+m}$ である．

部分帰納的関数の定義 IV

(2.4) (合成)
$n \in \mathbb{N}$ および $m \in \mathbb{N}^+$ とする．さらに，$\tau \in \mathcal{P}^{n+1}$，
$\psi \in \mathcal{P}^m$ とし，
$\phi(x_1,\ldots,x_n,y_1,\ldots,y_m) =_{df} \tau(x_1,\ldots,x_n,\psi(y_1,\ldots,y_m))$
と定義する．

このとき，$\phi \in \mathcal{P}^{n+m}$ である．

(2.5) (原始再帰)
$n \in \mathbb{N}$，$\tau \in \mathcal{P}^n$ および $\psi \in \mathcal{P}^{n+2}$ とする．もし，

$$\begin{aligned} \phi(x_1,\ldots,x_n,0) &=_{df} \tau(x_1,\ldots,x_n); \\ \phi(x_1,\ldots,x_n,y+1) &=_{df} \psi(x_1,\ldots,x_n,y,\phi(x_1,\ldots,x_n,y)), \end{aligned}$$

ならば，$\phi \in \mathcal{P}^{n+1}$ である．

部分帰納的関数の定義 V

(2.6) (μ-再帰)

$n \in \mathbb{N}^+$ とする．このとき，次の通りとする：

もし $\tau \in \mathcal{P}^{n+1}$ かつ

$\psi(x_1,\ldots,x_n) =_{df} \mu y[\tau(x_1,\ldots,x_n,y) = 1]$

$$=_{df} \begin{cases} \text{以下を満たす最小の } y \\ (1)\ \text{任意の } v \leqslant y \text{ について } \tau(x_1,\ldots,x_n,v) \text{ が定義される;} \\ (2)\ \text{任意の } v < y \text{ について } \tau(x_1,\ldots,x_n,v) \neq 1; \\ (3)\ \tau(x_1,\ldots,x_n,y) = 1 \text{ なる } y \text{ が存在;} \\ \text{未定義}, \qquad \text{それ以外;}. \end{cases}$$

ならば $\psi \in \mathcal{P}^n$ とする．

Dedekind の正当化定理 I

(2.5) を除くすべての操作は明示的であることに注意しよう．操作 (2.5) では，左辺および右辺に ϕ が出現するため，それ自身が暗示的 *(implicit)* な定義の一部をなしている．よって，議論を続ける前に，操作 (2.5) が常にある関数を定義するか否かを確かめる必要がある．これに対する明白な手段はない．すべての暗示的な定義は正当化を必要とすることを思い出して欲しい．

よって，次の定理を示す必要がある．

Dedekind の正当化定理 II

定理 2

τ および ψ が関数であるとき，操作 *(2.5)* を満たす唯一の関数 ϕ が存在する．

Dedekind の正当化定理 II

定理 2

τ および ψ が関数であるとき，操作 *(2.5)* を満たす唯一の関数 ϕ が存在する．

証明： ϕ が唯一であることと存在することを示す必要がある．

Dedekind の正当化定理 III

主張 1. 操作 *(2.5)* を満たす高々ひとつの関数 ϕ が存在する.

Dedekind の正当化定理 III

主張 1. 操作 (2.5) を満たす高々ひとつの関数 ϕ が存在する.

操作 (2.5) を満たす関数 ϕ_1 および ϕ_2 が存在すると仮定する. y に関する帰納法で次を示す.

$$\phi_1(x_1,\ldots,x_n,y) = \phi_2(x_1,\ldots,x_n,y) \text{ for all } x_1,\ldots,x_n,y \in \mathbb{N}\,.$$

帰納法の基底として, $y = 0$ を考える. このとき, すべての $x_1,\ldots,x_n \in \mathbb{N}$ について, 以下を得る.

$$\begin{aligned}\phi_1(x_1,\ldots,x_n,0) &= \tau(x_1,\ldots,x_n)\\ &= \phi_2(x_1,\ldots,x_n,0)\,.\end{aligned}$$

Dedekind の正当化定理 IV

いま，帰納法の仮定 (IH と略) として，すべての $x_1, \ldots, x_n \in \mathbb{N}$ とある $y \in \mathbb{N}$ について次を仮定する．

$$\phi_1(x_1, \ldots, x_n, y) = \phi_2(x_1, \ldots, x_n, y) .$$

帰納ステップは y から $y+1$ に対して行なわれる．操作 (2.5) を用いて次を得る．

$$\begin{aligned}
\phi_1(x_1, \ldots, x_n, y+1) &= \psi(x_1, \ldots, x_n, y, \phi_1(x_1, \ldots, x_n, y)) \\
&\qquad \text{定義より} \\
&= \psi(x_1, \ldots, x_n, y, \phi_2(x_1, \ldots, x_n, y)) \\
&\qquad \text{IH より} \\
&= \phi_2(x_1, \ldots, x_n, y+1) \quad \text{定義より} .
\end{aligned}$$

よって $\phi_1 = \phi_2$ となり，主張 1 が証明される．

Dedekind 正当化定理 V

主張 2. 操作 *(2.5)* を満たすある関数 ϕ が存在する.

ϕ の存在を証明するために，ϕ の帰納的かつ暗黙的な定義を，明示的な定義の無限列で置き換える．すなわち，以下の通りとする.

$$\phi_0(x_1,\ldots,x_n,y) = \begin{cases} \tau(x_1,\ldots,x_n), & \text{if } y = 0\,; \\ \text{not defined}, & \text{otherwise}\,. \end{cases}$$

$$\phi_1(x_1,\ldots,x_n,y) = \begin{cases} \phi_0(x_1,\ldots,x_n,y), & \text{if } y < 1\,; \\ \psi(x_1,\ldots,x_n,0,\phi_0(x_1,\ldots,x_n,0)), & \text{if } y = 1\,; \\ \text{not defined}, & \text{otherwise}\,. \end{cases}$$

$$\ldots$$

Dedekind's 正当化定理 VI

$$\phi_{i+1}(x_1,\ldots,x_n,y) = \begin{cases} \phi_i(x_1,\ldots,x_n,y), & \text{if } y < i+1\,; \\ \psi(x_1,\ldots,x_n,i,\phi_i(x_1,\ldots,x_n,i)), & \text{if } y = i+1\,; \\ \text{not defined}, & \text{otherwise}\,. \end{cases}$$
$$\ldots$$

関数 ϕ_i のすべての定義は**明示的**であり，よって，公理を形成する集合から，関数 ϕ_i は存在する．結果として，$y \in \mathbb{N}$ とすべての $x_1,\ldots,x_n \in \mathbb{N}$ について，

$$\phi(x_1,\ldots,x_n,y) =_{df} \phi_y(x_1,\ldots,x_n,y)$$

なる関数 ϕ が必ず存在する．

Dedekind's 正当化定理 VII

さらに，構成の仕方より，直ちに

$$\begin{aligned}
\phi(x_1,\ldots,x_n,0) &= \phi_0(x_1,\ldots,x_n,0)\\
&= \tau(x_1,\ldots,x_n) \quad \text{and}\\
\phi(x_1,\ldots,x_n,y+1) &= \phi_{y+1}(x_1,\ldots,x_n,y+1)\\
&= \psi(x_1,\ldots,x_n,y,\phi_y(x_1,\ldots,x_n,y))\\
&= \psi(x_1,\ldots,x_n,y,\phi(x_1,\ldots,x_n,y))\,,
\end{aligned}$$

が得られ，よって，ϕ は操作 (2.5) を満たす. ∎

部分帰納関数の定義 VI

定義 1

すべての部分帰納関数 *(partial recursive function)* の族 (クラス) $\mathcal{P}$ を，関数 Z，S，V および，これらに操作 (2.1) から (2.6) を有限回適用して得られる関数を含む最小の関数族と定義する．

部分帰納関数の定義 VI

定義 1

すべての部分帰納関数 *(partial recursive function)* の族 (クラス) $\mathcal{P}$ を，関数 Z，S，V および，これらに操作 (2.1) から (2.6) を有限回適用して得られる関数を含む最小の関数族と定義する．

さらに，その重要な部分族である原始帰納的関数の族を次の通り定義する．

定義 2

すべての原始帰納的関数 *(primitive recursive function)* の族 (クラス) *Prim* を，関数 Z，S，V および，これらに操作 (2.1) から (2.5) を有限回適用して得られる関数を含む最小の関数族と定義する．

例. 1 (恒等関数)

任意の $x \in \mathbb{N}$ について $I(x) = x$ と定義される恒等関数
$I\colon \mathbb{N} \to \mathbb{N}$ は原始帰納的である.

証明： 操作 (2.4) を適用する．$n = 0$ および $m = 1$ とする．定義 (手順 (1) 参照) より，$V, S \in \mathcal{P}^1$ であることがわかる．よって，V を操作 (2.4) における τ と考え ($n + 1 = 0 + 1 = 1$ であることに注意)，同様に S を ψ と考える ($m = 1$ であることに注意)．結果として，所望の関数 I は操作 (2.4) における ϕ となり ($n + m = 0 + 1 = 1$ に注意)，次の通りとなる.

$$I(x) = V(S(x)) .$$

以上より，恒等関数 I は原始帰納的である.

例. 2 (二項和)

任意の $n, m \in \mathbb{N}$ について $\alpha(n, m) = n + m$ と定義される二項和演算 (関数) $\alpha\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ は原始帰納的である.

証明: 仮定より $S \in \mathcal{P}$, また, 例 1 より $I \in \mathit{Prim}$ である. まず, 以下に示す操作を用いて, いくつかの外部関数を定義する.

$$\begin{aligned}
\psi(x_1, x_2) &=_{df} S(x_1) && \text{操作 (2.1) による;}\\
\tilde{\psi}(x_1, x_2) &=_{df} \psi(x_2, x_1) && \text{操作 (2.3) による;}\\
\tau(x_1, x_2, x_3) &=_{df} \tilde{\psi}(x_1, x_2) && \text{操作 (2.1) による;}\\
\tilde{\tau}(x_1, x_2, x_3) &=_{df} \tau(x_1, x_3, x_2) && \text{操作 (2.3) による .}
\end{aligned}$$

さて, α を定義するために操作 (2.5) を適用し, 次の通り定める.

$$\begin{aligned}
\alpha(n, 0) &=_{df} I(n),\\
\alpha(n, m+1) &=_{df} \tilde{\tau}(n, m, \alpha(n, m)) .
\end{aligned}$$

二項和 II

操作 (2.1) から (2.5) のみを用いたので，$\alpha \in Prim$ であることがわかる．

二項和 II

操作 (2.1) から (2.5) のみを用いたので，$\alpha \in \mathit{Prim}$ であることがわかる．

では，$\alpha(n,1)$ を計算してみよう．次を得る

$$\begin{aligned}
\alpha(n,1) &= \alpha(n,0+1) = \tilde{\tau}(n,0,\alpha(n,0)) \\
&= \tilde{\tau}(n,0,I(n)) \quad \alpha(n,0) = I(n) \text{より}, \\
&= \tilde{\tau}(n,0,n) \quad I(n) = n \text{より}, \\
&= \tau(n,n,0) \quad \tilde{\tau} \text{の定義より}, \\
&= \tilde{\psi}(n,n) \quad \tau \text{の定義より}, \\
&= \psi(n,n) \quad \tilde{\psi} \text{の定義より}, \\
&= S(n) = n+1 \quad \psi \text{ and } S \text{ の定義より}.
\end{aligned}$$

二項和 III

この定義は必要以上に複雑に見えるかもしれない．そうでないことを確かめるために $\alpha(n,2)$ を計算する．

$$\begin{aligned}
\alpha(n,2) &= \alpha(n,1+1) = \tilde{\tau}(n,1,\alpha(n,1)) \\
&= \tilde{\tau}(n,1,n+1) \quad \alpha(n,1) = n+1 \text{ より} \ , \\
&= \tau(n,n+1,1) \\
&= \tilde{\psi}(n,n+1) \\
&= \psi(n+1,n) \\
&= S(n+1) = n+2 \, .
\end{aligned}$$

二項積

以下では，長い技術的な段階のいくつかを省略する．例えば，二項積が原始帰納的であることを明らかにするために，次の通り定めれば十分であることを指摘するに留める．

$$\begin{aligned} m(x,0) &= Z(x)\,, \\ m(x,y+1) &= \alpha(x,m(x,y))\,. \end{aligned}$$

二項積

以下では，長い技術的な段階のいくつかを省略する．例えば，二項積が原始帰納的であることを明らかにするために，次の通り定めれば十分であることを指摘するに留める．

$$\begin{aligned} m(x,0) &= Z(x)\,, \\ m(x,y+1) &= \alpha(x,m(x,y))\,. \end{aligned}$$

さらに，定数 1 関数 c は原始帰納的であることにも注意する．すなわち，任意の $n \in \mathbb{N}$ について $c(n) = 1$ である．このことを確かめるために，次の通り定める．

$$\begin{aligned} c(0) &= S(0)\,, \\ c(n+1) &= c(n)\,. \end{aligned}$$

以下では，$c(n)$ の代わりに単に 1 と書く．

符号関数と数論的減算

符号関数 (signum function) sg が原始帰納的であることは以下から容易にわかる.

$$\begin{aligned} sg(0) &= 0\,, \\ sg(n+1) &= 1\,. \end{aligned}$$

符号関数と数論的減算

符号関数 (signum function) sg が原始帰納的であることは以下から容易にわかる.

$$\begin{aligned} sg(0) &= 0\,, \\ sg(n+1) &= 1\,. \end{aligned}$$

自然数は減算に関して閉じていないので，もし $m \geqslant n$ ならば $m \dot{-} n = m - n$，そうでなければ 0 と定義される，いわゆる数論的減算 (arithmetic difference) を慣習的に用いる．任意の $n, m \in \mathbb{N}$ について次の通りであることから，数論的減算もまた原始帰納的である.

$$\begin{aligned} m \dot{-} 0 &= I(m)\,, \\ m \dot{-} (n+1) &= V(m \dot{-} n)\,. \end{aligned}$$

これまでの例の一般化は，テキストにある.

場合分けI

場合分けによって関数を定義することが非常に多い．(例えば，前者関数 V の定義を参照)．よって，どのようなときに，場合分けによる定義が原始的帰納であることを保てるのかを問うことは自然である． 性質を記述する便利な方法は，**述語** *(predicate)* を用いることである．自然数上の n-引数述語 p は $\mathbb{N}^n$ の部分集合である．通常は，$(x_1,\ldots,x_n) \in p$ の代わりに $p(x_1,\ldots,x_n)$ と書く．n-引数述語の特性関数 (characteristic function) は，次の通り定義される関数 $\chi_p : \mathbb{N}^n \mapsto \{0,1\}$ である．

$$\chi_p(x_1,\ldots,x_n) = \begin{cases} 1, & \text{if } p(x_1,\ldots,x_n)\,; \\ 0, & \text{otherwise}\,. \end{cases}$$

場合分け II

定義 3

χ_p が原始帰納的であるとき，述語 p は 原始帰納的 *(primitive recursive)* であると言われる．

場合分け II

定義 3

χ_p が原始帰納的であるとき，述語 p は 原始帰納的 *(primitive recursive)* であると言われる．

定義 4

p, q を n-引数述語とする．このとき，$p \wedge q$ を集合 $p \cap q$，$p \vee q$ を $p \cup q$，および，$\neg p$ を $\mathbb{N}^n \setminus p$ と定義する．

場合分け III

補題 3

p, q を任意の原始帰納的 n-引数述語とする．このとき，$p \wedge q$，$p \vee q$，および，$\neg p$ もまた原始帰納的である．

場合分け III

補題 3

p, q を任意の原始帰納的 n-引数述語とする．このとき，$p \wedge q$，$p \vee q$，および，$\neg p$ もまた原始帰納的である．

証明： 明らかに，以下が成り立つ．

$$\begin{aligned}
\chi_{p\wedge q}(x_1,\ldots,x_n) &= \chi_p(x_1,\ldots,x_n)\cdot\chi_q(x_1,\ldots,x_n)\,,\\
\chi_{p\vee q}(x_1,\ldots,x_n) &= sg(\chi_p(x_1,\ldots,x_n)+\chi_q(x_1,\ldots,x_n))\,,\\
\chi_{\neg p}(x_1,\ldots,x_n) &= 1\dot{-}\chi_p(x_1,\ldots,x_n)\,.
\end{aligned}$$

これまでの議論より，加算・乗算・数論的減算は原始帰納的なので，補題の言明の通りとなる． ∎

場合分け IV

以上より，場合分けによる関数定義に関する定理を示すことができる．

定理 4

$p_1, \ldots, p_k$ を n-引数の原始帰納的述語とし，互いに素であるとする．また，$\psi_1, \ldots, \psi_k \in \mathcal{P}^n$ は原始帰納的関数であるとする．このとき，次の通り定義される関数 $\gamma\colon \mathbb{N}^n \to \mathbb{N}$ は，原始帰納的である．

$$\gamma(x_1, \ldots, x_n) =_{df} \begin{cases} \psi_1(x_1, \ldots, x_n), & \text{if } p_1(x_1, \ldots, x_n)\,; \\ . & \\ . & \\ . & \\ \psi_k(x_1, \ldots, x_n), & \text{if } p_k(x_1, \ldots, x_n)\,; \\ 0, & \text{otherwise}\,. \end{cases}$$

場合分け V

証明： γ は

$$\gamma(x_1,\ldots,x_n) = \sum_{i=1}^{k} \chi_{p_i}(x_1,\ldots,x_n) \cdot \psi_i(x_1,\ldots,x_n)\,,$$

と表すことができるので，一般の加算と積算の原始帰納性より定理の通りとなる．∎

対関数 I

$\mathbb{N} \times \mathbb{N}$ から $\mathbb{N}$ への全単射（対関数; pairing function）を考えることは多くの場面において非常に有用である．そこで，まずはこうした全単射が存在するか否かを問う必要がある．

対関数 I

$\mathbb{N} \times \mathbb{N}$ から $\mathbb{N}$ への全単射（対関数; pairing function）を考えることは多くの場面において非常に有用である．そこで，まずはこうした全単射が存在するか否かを問う必要がある．

これは確かに存在する．

$\mathbb{N} \times \mathbb{N}$ の要素は自然数の順序対であることから，$\mathbb{N} \times \mathbb{N}$ のすべての要素は 2 次元配列で容易に表現できる．ここで，行 x は，すべての対 (x, y)，すなわち，最初の要素が x で，$y = 0, 1, 2, \ldots$ なる対を含む
(図 1参照).

対関数 II

$$
\begin{array}{cccccc}
(0,0) & (0,1) & (0,2) & (0,3) & (0,4) & \ldots \\
(1,0) & (1,1) & (1,2) & (1,3) & (1,4) & \ldots \\
(2,0) & (2,1) & (2,2) & (2,3) & (2,4) & \ldots \\
(3,0) & (3,1) & (3,2) & (3,3) & (3,4) & \ldots \\
(4,0) & (4,1) & (4,2) & (4,3) & (4,4) & \ldots \\
(5,0) & \ldots & & & & \\
\ldots & \ldots & & & &
\end{array}
$$

図 1: $\mathbb{N} \times \mathbb{N}$ を表す 2 次元配列.

対関数 III

ここで，すべての対を次の通り始まる列に並べ替えることを考える．

$$(0,0),(0,1),(1,0),(0,2),(1,1),(2,0),(0,3),(1,2),(2,1),(3,0),\ldots \quad (1)$$

この順序においては，$x+y<x'+y'$ のとき，かつ，そのときに限り，すべての対 (x,y) がすべての対 (x',y') の前に出現する．つまり，これらは要素の和が徐々に大きくなる順序に並べ替えられている．要素の和が同じ対は第一要素が最も小さいものから並べられる．すなわち，対 (x,y) は次の区間に置かれる．

$$(0,x+y),\ (1,x+y-1),\ \ldots,\ (x,y),\ \ldots,\ (x+y,0)\,.$$

要素の和が $x+y$ の対は，$x+y+1$ あることに注意する．よって，列 (1) 中の対 $(0,x+y)$ の前には，$x+y$ の区間があり，それは全部で

$$1+2+3+\cdots+(x+y)$$

の対を含む．

対関数 IV

ここで,

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2} = \sum_{i=1}^{n} i$$

であることを考慮し，所望の全単射 $c\colon \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ を次の通り定義することができる.

$$\begin{aligned} c(x,y) &= \frac{(x+y)(x+y+1)}{2} + x \\ &= \frac{(x+y)^2 + 3x + y}{2}. \end{aligned} \tag{2}$$

対関数 V

c の計算に含まれるすべての操作 (演算) は原始帰納的であることが既に示されている．よって，c も原始帰納的であると結論することができる．

対関数 V

c の計算に含まれるすべての操作 (演算) は原始帰納的であることが既に示されている．よって，c も原始帰納的であると結論することができる．

課題．任意の $x, y \in \mathbb{N}$ について，もし $z = c(x, y)$ ならば，$x = d_1(z)$ かつ $y = d_2(z)$ となる関数 d_1 と d_2 を決定せよ．

対関数 V

c の計算に含まれるすべての操作 (演算) は原始帰納的であることが既に示されている．よって，c も原始帰納的であると結論することができる．

課題．任意の $x, y \in \mathbb{N}$ について，もし $z = c(x, y)$ ならば，$x = d_1(z)$ かつ $y = d_2(z)$ となる関数 d_1 と d_2 を決定せよ．

課題．与えられたすべての $k \in \mathbb{N}$ および $k > 2$ にいて，原始帰納的全単射 $c_k : \mathbb{N}^k \to \mathbb{N}$ が存在することを示せ．

対関数 V

c の計算に含まれるすべての操作 (演算) は原始帰納的であることが既に示されている．よって，c も原始帰納的であると結論することができる．

課題．任意の $x, y \in \mathbb{N}$ について，もし $z = c(x, y)$ ならば，$x = d_1(z)$ かつ $y = d_2(z)$ となる関数 d_1 と d_2 を決定せよ．

課題．与えられたすべての $k \in \mathbb{N}$ および $k > 2$ にいて，原始帰納的全単射 $c_k : \mathbb{N}^k \to \mathbb{N}$ が存在することを示せ．

課題．$\mathbb{N}^*$ を自然数のすべての有限列の集合とする．原始帰納的全単射 $c_* : \mathbb{N}^* \mapsto \mathbb{N}$ が存在することを示せ．

一般帰納的関数

次に，一般帰納的関数 *(general recursive function)* の族を定義する．

定義 5

各 $n \in \mathbb{N}^+$ について，$\mathcal{R}^n$ を，任意の $x_1, \ldots, x_n \in \mathbb{N}$ に対して $f(x_1, \ldots, x_n)$ が定義されるすべての関数 $f \in \mathcal{P}^n$ の集合であると定める．さらに，$\mathcal{R} = \bigcup_{n \in \mathbb{N}^+} \mathcal{R}^n$ とする．

別の言い方をすると，$\mathcal{R}$ は，全域的かつ部分帰納的なすべての関数の集合である．このとき，次の定理を示すことができる．

一般帰納的関数

次に，一般帰納的関数 *(general recursive function)* の族を定義する．

定義 5

各 $n \in \mathbb{N}^+$ について，$\mathcal{R}^n$ を，任意の $x_1, \ldots, x_n \in \mathbb{N}$ に対して $f(x_1, \ldots, x_n)$ が定義されるすべての関数 $f \in \mathcal{P}^n$ の集合であると定める．さらに，$\mathcal{R} = \bigcup_{n \in \mathbb{N}^+} \mathcal{R}^n$ とする．

別の言い方をすると，$\mathcal{R}$ は，全域的かつ部分帰納的なすべての関数の集合である．このとき，次の定理を示すことができる．

定理 5

$Prim \subset \mathcal{R} \subset \mathcal{P}$.

証明 I

証明： $Z, S, V \in \mathcal{R}$ であることは明らか．さらに，少し考えると，操作 (2.1) から (2.5) の有限回の適用は全域的関数のみを生ずることは明らかである．このことは，$Prim \subseteq \mathcal{R}$ を示している．また，$\mathcal{R} \subseteq \mathcal{P}$ は定義より明らかである．よって，あとは二つの包含関係が**真の包含関係**であることを示せばよい．

証明 I

証明： $Z, S, V \in \mathcal{R}$ であることは明らか．さらに，少し考えると，操作 (2.1) から (2.5) の有限回の適用は全域的関数のみを生ずることは明らかである．このことは，$Prim \subseteq \mathcal{R}$ を示している．また，$\mathcal{R} \subseteq \mathcal{P}$ は定義より明らかである．よって，あとは二つの包含関係が**真の包含関係**であることを示せばよい．

主張 1. $\mathcal{P} \setminus \mathcal{R} \neq \emptyset$.

定義より，$S \in \mathcal{P}$ であり，また，操作 (2.4) を用いると，$\delta(n) =_{df} S(S(n))$ もまた $\mathcal{P}$ に含まれることは容易にわかる．ここで，任意の $n \in \mathbb{N}$ について $\delta(n) = n + 2 > 1$ であることに注意する．

操作 (2.1) を用いて，$\tau(x, y) = \delta(y)$ と定義し，これより，$\tau \in \mathcal{P}$ である．結果として，

$$\psi(x) = \mu y[\tau(x, y) = 1]$$

は，どこにも定義されていない関数であり，よって $\psi \notin \mathcal{R}$ である．一方，構成の仕方より，$\psi \in \mathcal{P}$ である．故に，$\psi \in \mathcal{P} \setminus \mathcal{R}$ を得る．

証明 II

主張 2. $R \setminus \mathit{Prim} \neq \emptyset$.

この言明を示すことは，よりいっそう複雑である．最初に次の関数を定義する．

$$\begin{aligned} ap(0, m) &=_{df} m + 1\,, \\ ap(n+1, 0) &=_{df} ap(n, 1)\,, \\ ap(n+1, m+1) &=_{df} ap(n, ap(n+1, m))\,, \end{aligned}$$

これはアッカーマン-ペータ関数 (Ackermann-Péter function) と呼ばれている．ヒルベルトは，1926 年に，すべての全域的かつ計算可能な関数もまた原始帰納的であると予想した．この予想は，1928 年にアッカーマンにより反証され，1955 年にペータがアッカーマンの定義を単純化した．

証明 III

関数 ap は原始帰納的**ではない**こと，および，関数 ap は一般帰納的であることを示す必要がある．両者の証明は容易ではない．
よって，時間の都合上，いくつかの部分を省略しなければならない．しかし，証明を始める前に，関数 ap は**直観的に**計算可能であると考えることにしよう．そう考えるために，関数 ap を `peter` として実装した次の疑似コードの一部を考よう．

証明 IV

```
function peter(n, m)
            if n = 0
              return m + 1
            else if m = 0
              return peter(n - 1, 1)
            else
              return peter(n - 1, peter(n, m - 1))
```

証明 V

次に，ap は原始帰納的であり得ないことの証明の概略を述べる．まず，すべての原始帰納的関数 ϕ について，次に従ってある関数 f_ϕ を定義する．ϕ の引数の数を k とし，次を考える．

$$f_\phi(n) = \max\left\{\phi(x_1,\ldots,x_k) \mid \sum_{i=1}^{k} x_i \leqslant n\right\} .$$

関数族 *Prim* の帰納的な構成の仕方を用いて，構造的帰納法により，すべての原始帰納的関数 ϕ について，次に示すある数 $n_\phi \in \mathbb{N}$ の存在を示すことができる．

$$f_\phi(n) < \mathrm{ap}(n_\phi, n) \text{ for all } n \geqslant n_\phi .$$

直観的に述べると，後者の言明は，アッカーマン-ペータ関数の値がすべての原始帰納的関数より速く大きくなることを示している．

証明 VI

残りは容易である．$ap \in \mathit{Prim}$ と仮定する．恒等関数 I は原始帰納的であることを考慮すると，操作 (2.4) の適用により，

$$\kappa(n) = ap(I(n), I(n))$$

も原始帰納的であることが直ちにわかる．すると，κ について，次のある数 $n_\kappa \in \mathbb{N}$ が存在する．

$$f_\kappa(n) < ap(n_\kappa, n) \text{ for all } n \geqslant n_\kappa \,.$$

しかし，いま，

$$\kappa(n_\kappa) \leqslant f_\kappa(n_\kappa) \;<\; ap(n_\kappa, n_\kappa) = \kappa(n_\kappa)\,,$$

であるから，矛盾が生じる．

証明 VII

2 番目の部分については，$ap \in \mathcal{R}$ を証明しなければならない．これは主に，関数 ap を表現するための，操作 (2.1) から (2.5) および μ-操作を用いた構成方法を与えることを意味する．興味のある者は Hermes の文献を参照すること．

Thank you!