6 計算の可能性

6.1 オーダ

6.2 アルゴリズムの重要性

6.3 計算不可能な問題

6.4 P, NP

6.1 オーダ

オーダー： 計算量の表現法

- 入力のサイズnを引数として計算のステップ数の概略を表す関数で表す

例： オーダn^2とは、値が $C \cdot n^2$ **(**Cは定数**)**で抑えられることをいう

- $f : \mathbb{R}_+ \to \mathbb{R}$がオーダ$g : \mathbb{R}_+ \to \mathbb{R}$以下である
  $$\overset{\text{def}}{\iff} \quad \exists C \in \mathbb{R}_+ \exists N \in \mathbb{R}_+ \forall x > N \, f(x) \leq C \cdot g(x)$$
- fが多項式オーダ**:** 上の定義において多項式gが存在する

問**6.3：** $f(x)$が多項式オーダ（多項式で表せる関数）なら$\forall c \in \mathbb{R}_+ \ f(c \cdot x)$も多項式オーダー

略解： 前提より多項式x^kが存在して、

$$\exists C \in \mathbb{R}_+ \exists N \in \mathbb{R}_+ \forall x > N \ \ f(x) \le C \cdot x^k$$

$x = c \cdot x'$で変数を置き換えると、

$$\forall x' > \frac{N}{C} \ f(c \cdot x') \le C \cdot (c \cdot x')^k = (C \cdot c^k) \cdot x'^k$$

6.2 アルゴリズムの重要性

例： x^n の計算（$n \in \mathbb{N}$)

計算量$O(n)$

$$x^n = \begin{cases} 1 & \cdots\ n = 0 \\ x \cdot x^{n-1} & \cdots\ n > 0 \end{cases}$$

計算量$O(\log n)$

$$x^n = \begin{cases} 1 & \cdots\ n = 0 \\ x^{\frac{n}{2}} \cdot x^{\frac{n}{2}} & \cdots\ n > 0 \text{ かつ } n \text{ は偶数} \\ x \cdot x^{\lfloor \frac{n}{2} \rfloor} \cdot x^{\lfloor \frac{n}{2} \rfloor} & \cdots\ n > 0 \text{ かつ } n \text{ は奇数} \end{cases}$$

例： フィボナッチ数の計算

(1) 計算量$O(n)$

$$(a_n, a_{n+1}) = \begin{cases} (0,1) & \cdots\ n = 0 \\ (a_n, a_{n-1} + a_n) & \cdots\ n > 0 \end{cases}$$

(2) 計算量$O(?)$

$$a_n = \begin{cases} 0 & \cdots\ n = 0 \\ 1 & \cdots\ n = 1 \\ a_{n-1} + a_{n-2} & \cdots\ n \geq 2 \end{cases}$$

(3) 数式として与える

$$a_n = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2} \right)^n - \left(\frac{1-\sqrt{5}}{2} \right)^n \right\}$$

問**6.7 (2)** の方法でa_nを求めるのに必要な加算回数$t(n)$を漸化式で表し、その一般項を求めよ

6.3 計算不可能な問題

- 関数 f が計算可能 $\overset{\text{def}}{\Longleftrightarrow}$ f は帰納的
- 決定問題とは： 真か偽を返す全域的関数
- 問題が決定可能(決定不能) $\overset{\text{def}}{\Longleftrightarrow}$ 問題(＝全域関数)が計算可能(計算不可能)

- 例：チューリング機械の停止問題
  - 入力: チューリング機械 M のコード $\#M$ と $\vec{m} \in \mathbb{N}^n$
  - 出力： M が $\vec{m}$ を表すテープ入力 $T_{\vec{m}}$ に対して停止するか？
- 停止問題は決定不能： すなわち、上の問題を解く **(**必ず正しい答えを返す**)** プログラムは存在しない

- 定理**6.a** 次の部分関数$\mathbf{comp}_n : \mathbb{N}^{n+1} \to \mathbb{N}$は帰納的 **(**計算可能**)**

$$
\mathbf{comp}_n(z, \vec{x}) = \begin{cases} y & \cdots \quad \exists M,\ y \in \mathbb{N}. \\ & \qquad\qquad z = \#M \wedge T_{\vec{x}} \vdash_M^* T_y \\ \text{未定義} & \cdots \quad \text{それ以外のとき} \end{cases}
$$

- 証明の概略： 万能チューリング機械を構成する **(**$z = \#M$なるチューリング機械Mが存在しないときは、無限ループに入るように構成**)**

- **comp**$_n$は帰納関数に対する万能関数。なぜなら、定理**6.a**より、**comp**$_n : \mathbb{N}^{n+1} \to \mathbb{N}$の第一引数を適当に設定し固定することで、いかなる$n$引数の帰納関数を得ることも可能

- 以下を満たすときgは部分関数fの全域関数への拡張という
  - gは全域関数
  - $f(\vec{m})$が定義されるならば、その値は$g(\vec{m})$と等しい

- 定理**6.b**: $n > 0$について、部分関数**comp**$_n : \mathbb{N}^{n+1} \to \mathbb{N}$のいかなる全域関数へ拡張も帰納的関数でない**(**計算不可能**)**

- 定理**6.b**の証明：

(a) 帰納的関数$g : \mathbb{N}^{n+1} \to \mathbb{N}$が、$\mathbf{comp}_n$の全域関数への拡張と仮定

(b) $g'(x_1, x_2, \ldots, x_n) \stackrel{\mathbf{def}}{=} g(x_1, x_1, x_2, \ldots, x_n) + 1$ とおく **(**註：g'は全域的**)**

(c) $\exists k \in \mathbb{N}.\ \mathbf{comp}_n(k, x_1, \ldots, x_n) = g'(x_1, \ldots, x_n)$

(d) x_1をkととると、gは$\mathbf{comp}_n$の拡張なので

$$
\begin{aligned}
& g'(k, m_2, \ldots, m_n) \\
&= \mathbf{comp}_n(k, k, m_2, \ldots, m_n) && \text{((c)より)} \\
&= g(k, k, m_2, \ldots, m_n) && \text{(g'は全域的より)}
\end{aligned}
$$

となり、g'の定義に矛盾

- 定理**6.c**： $n > 0$ のとき、次の述語 $\mathbf{halt}_n : \mathbb{N}^{n+1} \to \{真, 偽\}$ は帰納的でない

$$\mathbf{halt}_n(z, \vec{x}) = \begin{cases} 真 & \cdots\ \mathbf{comp}_n(z, \vec{x}) \text{ が値を持つとき} \\ 偽 & \cdots\ \mathbf{comp}_n(z, \vec{x}) \text{ が未定義のとき} \end{cases}$$

- 証明：$\mathbf{halt}_n$ か帰納的であると仮定し矛盾を導く
  - 次の関数が帰納的であることが示せる

$$g(z, \vec{x}) = \begin{cases} \mathbf{comp}_n(z, \vec{x}) & \cdots\ \mathbf{halt}_n(x, \vec{x}) \text{ が真} \\ 0 & \cdots\ \mathbf{halt}_n(x, \vec{x}) \text{ が偽} \end{cases}$$

  - 定理**6.b**より矛盾

- $\mathbf{halt}_n$ の特徴関数が帰納的でないことから、停止問題は決定不能

- 全域的な帰納関数に対する万能関数が存在しないことを示す
- 定理**6.d**： $n > 0$に対して、以下を満たす帰納的な全域関数$g : \mathbb{N}^{n+1} \to \mathbb{N}$は存在しない

$$\{g(k,\vec{x}) : \mathbb{N}^n \to \mathbb{N} \mid k \in \mathbb{N}\}$$
$$= \{f : \mathbb{N}^n \to \mathbb{N} \mid f \text{は帰納的な全域関数}\}$$

- 証明： $n = 1$の場合のみを示す。
  - 条件を満たす帰納的な全域関数gが存在すると仮定
  - $g'(x) \overset{\mathbf{def}}{=} g(x,x) + 1$も帰納的な全域関数であるから、$g(k,x) = g'(x)$ を満たす$k \in \mathbb{N}$が存在
  - xをkととると、$g(k,k) = g(k,k) + 1$が導かれ、矛盾

- 系： $\mathbf{total}_n(z) \stackrel{\mathbf{def}}{=} \forall \vec{x}.\ \mathbf{halt}_n(z, \vec{x})$ は帰納的述語ではない

- 証明：
  - $\mathbf{total}_n$ が帰納的と仮定する
  - 次の関数 $g : \mathbb{N}^{n+1} \to \mathbb{N}$ が帰納的であることが示せる

$$g(z, \vec{x}) = \begin{cases} \mathbf{comp}_n(z, \vec{x}) & \cdots\ \mathbf{total}_n(z) \text{ が真のとき} \\ 0 & \cdots\ \text{それ以外のとき} \end{cases}$$

  - 定理**6.d**に矛盾

- よって、プログラムが計算する関数が全域的かどうかを判定する問題は決定不能

- $\mathbf{comp}_n(z, \vec{x})$ の z を固定して n 変数の部分関数とみたとき、これを $\varphi_z^n(\vec{x})$ と表記する
- 帰納的でない述語の例

  - $p_1(z) = \forall \vec{x}.\ (\varphi_z^n(\vec{x}) = 0)$

  - $p_2(z) = \exists \vec{x}.\ (\varphi_z^n(\vec{x}) = 0)$

  - $p_3(z)$： φ_z^n の定義域が有限集合

  - $p_4(z)$： φ_z^n が定数関数

  - $p_5(z, z') = \varphi_z^n = \varphi_{z'}^n$

- 帰納的述語でないことを示すための強力な定理
- 定理 **(Riceの定理)**： $n \in \mathbb{N}$とするとき、**(**全域的な**)** 述語$p(z)$ が次の二つの条件を共に満たすならば$p(z)$は帰納的でない

(1) $\forall k, k' \in \mathbb{N}.\ (\varphi^n_k = \varphi^n_{k'} \Rightarrow p(k) = p(k'))$

(2) $\exists k, k' \in \mathbb{N}.\ (p(k) = 真 \wedge p(k') = 偽)$

- 証明： 略

6.4 P, NP

P問題： 多項式オーダの時間計算量で解くアルゴリズムが存在する問題

NP問題： 多項式オーダの時間計算量で解がチェックできる問題