Turing 次数

宮部賢志

2017 年 8 月 4 日

目次

0.1 Turing 還元と相対的計算可能性

計算可能な集合の定義と性質はすでにみた．計算可能ではないかもしれないが，計算可能に近似できる集合として c.e. 集合が存在することを紹介した．計算不可能な c.e. 集合の例として停止問題などの重要な例があることも見た．このような計算不可能な集合を計算することがどのくらい難しいのかを調べる道具が，帰着 (還元，reducibility) の概念である．

計算論において最も重要な概念は，Turing 機械などの計算モデルによる計算可能性の定義であろう．その次に重要な概念は，この帰着の概念だと思う．この帰着の概念の定義が，初期の計算論の研究の方向性を決定づけたと言っても良い．計算論は計算不可能な集合の計算不可能性を調べる分野になったのである．

どちらにしろ計算不可能なものの計算困難さを調べることに一体何の意味があるのかと思う人もあると思う．数理論理学や純粋数学において出て来る様々な構造，例えばランダムネスの理論における実数の複雑性などを調べる上で，計算困難さを測る道具はとても有用である．これから定義する Turing 還元は最も基本的な帰着の概念であるが，様々な分野において帰着の概念は導入され，基本的な道具になっている．

まず，帰着という言葉の数学における一般的な使い方を復習しておこう．3 次方程式 $ax^3 + bx^2 + cx + d = 0$ の解を求めるという問題を考えよう．これは解くことのできる問題である．ここで，$y = a^{1/3}(x + \frac{3b}{a})$ とおくことで，$y^3 + a'y + b' = 0$ という形の 3 次方程式を解く問題に帰着できる．ここで「帰着」という言葉は，問題 A を解くことができれば，問題 B を解くことができる，という意味で使う．問題 B の難しさは，問題 A の難しさ以上ではない．普通は「簡単な問題」に帰着する（そうでなければ，帰着を考える意味がない）．しかし，帰着という言葉そのものからすると，問題 A の方が簡単ではないことを意味する．

2 進無限列 $A, B \in 2^\omega$ はそれぞれ問題の答えだと思って，計算できるかという問題の困難さをこの帰着を使って表現する．$n \in A$ かどうかを知りたい．A が計算可能であれば計算すれば良いが，A は計算可能でないかもしれない．もし計算可能な関数 $f : \omega \to \omega$ が存在して，$n \in A \iff f(n) \in B$ であれば，B の答えが分かれば，A の答えは計算できる．このような場合には，問題 A が問題 B に帰着できるという．

ここで「B の答えが分かれば」と表現した部分で，計算機が B の情報へアクセスできるようにしたい．ちょうど計算機がインターネットにアクセスして情報を入手するように，計算機に B への情報をアクセスする機構を付け加える．そのようにして，この帰着の概念を定式化しよう．

Turing 機械を拡張して神託付き Turing 機械を定義する．神託 (oracle) とは，神のお

告げということで，人間には知り得ないことを神様に教えてもらうという意味で使う．神託付き Turing 機械では，神託 (oracle) が書き込まれているテープがあり，そのテープの値を計算中に聞くことができる．神託は通常 $X \in 2^\omega$ として与えられているとし，計算途中で $X(k)$ の値は何かを使って計算を行うことができる．計算途中で何回聞いても良い．

指数 e の計算を Φ_e で表すのであった．$Y \in 2^\omega$ が計算可能であれば，ある e が存在して，すべての n で $Y(n) = \Phi_e(n)$ である．神託 X を使った指数 e のプログラムによる入力 n の計算結果を $\Phi_e^X(n)$ と書く．Y が X を使って計算できるのであれば，すべての n で $Y(n) = \Phi_e^X(n)$ である．これを単に $Y = \Phi_e^X$ と書く．Φ_e は神託 2^ω から 2^ω への関数と見ることができて，そう見たときには Φ_e はTuring 還元 (Turing reduction) と呼ばれる．

定義 0.1.1. $A, B \in 2^\omega$ に対し，A が B に対してTuring 還元可能 (Turing reducible) であるとは，ある Turing 還元 Φ が存在して，$A = \Phi^B$ となることを言う．このことを，$A \leq_T B$ で表す．また，$A \leq_T B$ かつ $B \leq_T A$ のとき，$A \equiv_T B$ で表す．

$K = \{e : \Phi_e(e)\downarrow\}$ と，$H = \{\langle e, k\rangle : \Phi_e(k)\downarrow\}$ を思い出しておこう．

例 0.1.2. すべての c.e. 集合は H に Turing 還元可能である．

証明．A を c.e. 集合とする．c.e. の定義から，ある指数 $x \in \omega$ が存在して，$A = \{n : \Phi_x(n)\downarrow\}$ となる．$n \in A$ かどうかは H を使って計算できることを示そう．

Turing 還元 Φ を次のように定義する．神託を X とする．入力 n に対して，$y = \langle x, n\rangle$ を計算する．$X(y) = 1$ ならば $f(n) = 1$ を出力し，$X(y) = 0$ ならば $f(n) = 0$ を出力する．

神託が H であったときには A を計算する．すなわち $A(n) = \Phi^H(n)$ である．なぜなら，

$$n \in A \iff \Phi_x(n)\downarrow \iff \langle x, n\rangle \in H \iff f(n) = 1$$

であるから．　□

H 自身も c.e. 集合であるから，H は c.e. 集合の中で Turing 還元の意味で最も計算困難な集合であることが分かる．

この証明では数学的な定義にできる限り忠実に書いた．直感的に言えば，Y が X に Turing 還元可能であるとは，神託 X を「知っている」または「与えられている」ような仮想の世界を考えて，その世界で計算可能であることを意味する．そのため，Y は X から相対的に計算可能である (Y is computable relative to X) ともいう．日本語で言う場合長いので，単に，Y は X から見て計算可能であるともいう．X は Y を (Turing の意味で) 計算する (X computes Y) とも表現する．様々な還元可能性があるので，議論する

ときにはどの還元を考えているかには注意する必要がある．計算論では何も言わなければ Turing 還元を考えていると思って良い．

K と H は共に計算不可能な集合であり，その計算不可能性は同じである．数式としては K を使うほうが多いが，直感的な推論を行うときには H の方が考えやすい．それらの困難さが同じであることは，特に深く言及せずによく使われる．

命題 0.1.3.

$$K \equiv_T H$$

証明．H が与えられていれば，入力 e に対し，$H(\langle e, e\rangle)$ の値を聞くことで，$K(e)$ の値が判定できる．すなわち，$K \leq_T H$ である．

e, k が与えられたときに，n に関係なく，

$$f(n)\downarrow \iff \Phi_e(k)\downarrow$$

となる計算可能関数 f の指数を $g(e, k)$ とする．すると，

$$g(e,k) \in K \iff \Phi_{g(e,k)}(g(e,k))\downarrow \iff \Phi_e(k)\downarrow \iff \langle e, k\rangle \in H$$

となり，K を使って H が計算できる． □

演習問題 0.1.4. すべての $A \in 2^\omega$ に対し，$\overline{A} \equiv_T A$ であることを示せ．

K を使うと計算ができる集合が増えるが，K を使っても計算できない集合が存在する．

定義 0.1.5. $A \in 2^\omega$ に対し，$A' = \{n : \Phi_n^A(n)\downarrow\}$($A$ のジャンプ) と定義する．n 回ジャンプは $A^{(n)}$ と表記する．

$K = \emptyset'$ である．以降は K ではなく，$\emptyset'$ で表記する．

命題 0.1.6. 任意の $A \in 2^\omega$ に対し，

$$A <_T A'$$

特に，

$$\emptyset <_T \emptyset' <_T \emptyset'' <_T \cdots <_T \emptyset^{(n)} < \cdots$$

証明．$A \leq_T A'$ は，各 $i \in \omega$ に対し $f(n)\downarrow \iff A(i) = 1$ となる関数 f を考えれば，その指数を見ることで A' から A が計算できることが分かる．

$A' \not\leq_T A$ は停止問題の計算不可能性と同様の議論で示せる． □

$\leq_T$ は前順序となり，$\equiv_T$ は同値関係となる．Turing 次数には自然な関係により半順序になる．ただし，全順序にはならない．

演習問題 0.1.7. 　(1) $\leq_T$ が partial order であることを示せ．

(2) $\equiv_T \iff \leq_T \wedge \geq_T$ とすれば，$\equiv_T$ が同値関係であることを示せ．

2^ω を $\equiv_T$ で割った商集合の各元をTuring 次数という．

定義 0.1.8. A が B に多対一還元 (many-one reducible) であるとは，計算可能な関数 f が存在して，$n \in A \iff f(n) \in B$ となること．

0.2　算術的階層

集合 $A \subseteq \omega$ が c.e. であるとは，ある計算可能な関数 $\Phi :\subseteq \omega \to \omega$ が存在して，

$$n \in A \iff \Phi(n)\downarrow$$

となることであった．段階 s を使えば，

$$n \in A \iff (\exists s)\Phi(n)[s]\downarrow$$

と書くこともでき，$\Phi(n)[s]\downarrow$ は n, s に関して計算可能な関係である．これを一般化して，自然数の集合の複雑さを，記述する論理式の複雑さによって表現するのが算術的階層である．算術的階層は集合の複雑さを表現する粗い指標である．後に Turing 還元との関連も見る．

$A \subseteq \omega$ が計算可能であることと，$n \in A \iff R(n)$ となる計算可能な関係 R が存在することとは同値である．(計算可能な関係の定義そのものである．)

まずは c.e. 集合の特徴付けから始めよう．これを拡張することで算術的階層は作られる．

命題 0.2.1. $A \subseteq \omega$ に対して，以下は同値.

(1) A は *c.e.*

(2) $n \in A \iff (\exists s)R(n,s)$ となる計算可能な関係 R が存在する．

証明．(1)⇒(2). A が c.e. であれば，ある計算可能な関数 $\Phi :\subseteq \omega \to \omega$ が存在して，$n \in A \iff \Phi(n)\downarrow \iff (\exists s)\Phi(n)[s]\downarrow$. ここで，$\Phi(n)[s]\downarrow$ は計算可能な関係である．

(2)⇒(1). $n \in A \iff (\exists s)R(n,s)$ となる計算可能な関係 R が存在したとしよう．$\Phi(n) = \mu s.R(n,s)$ とする．ここで μ は最小化演算子である．すなわち，$R(n,s)$ となる最小の s を返す関数である．すると，$n \in A \iff \Phi(n)\downarrow$ となり，A は c.e. である．　□

この量化子 ∃ は複数ついていても，1 つにまとめることができる．

$$(\exists s)(\exists t)R(n,s,t) \iff (\exists u)[u = \langle s,t\rangle \wedge R(n,s,t)]$$

このことから，計算可能な関係に量化子がついた形の論理式は，$\exists$ と $\forall$ が 1 つずつ交互に現れる形に書くことができる．

定義 0.2.2. $A \in \Sigma_n^0$ であるとは，ある計算可能な関係 R が存在して，すべての $k \in \omega$ について，

$$k \in A \iff (\exists x_1)(\forall x_2)\cdots(Qx_n)R(k, x_1, x_2, \cdots, x_n)$$

となることをいう．ここで，Q は n が奇数の時は $\exists$ で，n が偶数の時は $\forall$ である．$\overline{A} \in \Sigma_n^0$ のとき，$A \in \Pi_n^0$ と書く．また，$\Delta_n^0 = \Sigma_n^0 \cap \Pi_n^0$ と書く．

Δ_1^0 集合とは計算可能な集合のことである．Σ_1^0 集合は c.e. 集合のことである．Π_1^0 集合は co-c.e. 集合のことである．

0.3 極限補題

定理 0.3.1. $A \in 2^\omega$ に関して以下は同値.

(1) A が極限計算可能 *(limit computable)*（もしくは，計算近似可能，*computably approximable*）．
(2) $A \in \Delta_2$.
(3) $A \leq_T \emptyset'$.

証明．(1)⇒(2)．$A(x) = \lim_s A_s(x)$ で $\{A_s\}_s$ は計算可能な有限集合列とする．すると，

$$x \in A \iff (\exists s)(\forall t)[t > s \Rightarrow A_t(x) = 1],$$
$$x \in \overline{A} \iff (\exists s)(\forall t)[t > s \Rightarrow A_t(x) = 0].$$

よって，$A \in \Sigma_2$ かつ $\overline{A} \in \Sigma_2$.

(2)⇒(3).

$$x \in A \iff (\exists s)(\forall t)R(x, s, t)$$

および

$$x \in \overline{A} \iff (\exists s)(\forall t)S(x, s, t)$$

と書けたとする．$(\forall t)R(x, s, t)$ は Π_1 述語なので，$\emptyset'$ で計算可能．よって，$(\exists s)(\forall t)R(x, s, t)$ は $\Sigma_1(\emptyset')$． 同様にして，$(\exists s)(\forall t)S(x, s, t)$ も $\Sigma_1(\emptyset')$． よって，$A \leq_T \emptyset'$.

(3)⇒(1)．$A = \Phi_e^{\emptyset'}$ であったとする．$\mathbf{0}_s = \emptyset'[s]$ とおいて，

$$f(x, s) = \begin{cases} \Phi_{e,s}^{\mathbf{0}_s}(x) & \text{定義されていれば} \\ 0 & \text{それ以外} \end{cases}$$

とおく．各 x について，$\Phi_e^{\emptyset'}$ で使用する神託は有限なので，十分大きな s については上記の計算は正しい．よって，$A(x) = \lim_s f(x,s)$. □

これまでジャンプの階層と算術的階層を見てきたが，これらは非常に深く関係している．

定理 0.3.2 (Post の定理)**.** (1) $A \in \Sigma_{n+1}$ であることと，A が $\emptyset^{(n)}$ から *c.e.* であることは同値.

(2) $A \in \Delta_{n+1}$ であることと，$A \leq_T \emptyset^{(n)}$ であることは同値.

特に算術的階層が潰れていないこともこれから分かる．

0.4　指数集合

定義 0.4.1. $A \subseteq \omega$ が指数集合 (index set) であるとは，

$$x \in A, \Phi_x = \Phi_y \Rightarrow y \in A$$

を満たすことをいう．

$n \in A$ かどうかが，n を指数とする関数の性質だけに依存することを意味している．

定義 0.4.2. A がΣ_n 完全 (Σ_n-complete) であるとは，$A \in \Sigma_n$ であって，すべての $B \in \Sigma_n$ に対して，$B \leq_m A$ であることをいう．

定理 0.4.3 (Rice の定理)**.** A を指数集合で，空集合でも自然数全体でもないものとする．この時，A は計算不可能である．

多くの教科書では，停止問題 $\emptyset'$ を A または A の補集合へ帰着する方法で証明を与えている．

証明. $a \in \omega$ を $W_a = \emptyset$ となる指数とする．一般性を失わずに，$a \in A$ と仮定して良い．$b \in \omega$ を $b \notin A$ なる指数とする．

各 $e \in \omega$ に対して，$e \in \emptyset'$ かどうかを，A を使って次のように判定できる．「入力 x に対して，$\Phi_e(e)$ を計算して，もし止まれば $\Phi_b(x)$ を出力する」というプログラムの指数を $f(e)$ とする．もし $e \in \emptyset'$ であれば，$\Phi_{f(e)} = \Phi_b$ より $f(e) \notin A$. もし $e \notin \emptyset$ であれば，$\Phi_{f(e)} = \Phi_a$ より $f(e) \in A$. よって，$\emptyset' \leq_m \overline{A}$. □

再帰定理を使っても示すことができる．

証明. $a \in A, b \notin A$ をとる．A が計算可能であれば，

$$f(n) = \begin{cases} b & (n \in A) \\ a & (n \notin A) \end{cases}$$

は全域な計算可能関数である．再帰定理よりある指数 e が存在して，

$$\Phi_e = \Phi_{f(e)}.$$

この e に関して，

$$e \in A \iff f(e) \in A \iff e \notin A$$

となり矛盾． $\square$

指数集合は集合の複雑さを判定できる良い例であるだけでなく，様々な集合の複雑さを判定する上で使いやすい道具でもある．

定義 0.4.4.

$$\begin{aligned}
\mathrm{Fin} =& \{x \ : \ W_x \text{が有限集合}\},\\
\mathrm{Inf} =& \{x \ : \ W_x \text{が無限集合}\},\\
\mathrm{Tot} =& \{x \ : \ \Phi_x \text{が全域}\},\\
\mathrm{Cof} =& \{x \ : \ W_x \text{の補集合が有限集合}\},\\
\mathrm{Rec} =& \{x \ : \ W_x \text{が計算可能}\}.
\end{aligned}$$

命題 0.4.5.

$$\begin{aligned}
\mathrm{Fin} &\in \Pi_2,\\
\mathrm{Inf} &\in \Sigma_2,\\
\mathrm{Tot} &\in \Pi_2,\\
\mathrm{Cof} &\in \Sigma_3,\\
\mathrm{Rec} &\in \Sigma_3.
\end{aligned}$$

証明．$\mathrm{Tot} \in \Pi^0_2$ であることは，

$$e \in \mathrm{Tot} \iff \forall n \exists s[\Phi_{e,s}(n)\downarrow]$$

から従う．その他も同様． $\square$

命題 0.4.6. $\emptyset^{(n)}$ は Σ^0_2 完全．

定理 0.4.7. Tot は Π^0_2 完全．

証明．$A \in \Pi^0_2$ とすると，ある計算可能な関係 R が存在して，

$$x \in A \iff (\forall y)(\exists z)R(x, y, z)$$

各 x に対して次のような関数の指数の 1 つを $f(x)$ とする計算可能関数 f を考える：

$$\Phi_{f(x)}(u) = \begin{cases} 0 & (\forall y \le u)(\exists z)R(x, y, z) \\ \uparrow & \text{それ以外} \end{cases}$$

すると,

$$x \in A \iff f(x) \in \mathrm{Tot}.$$

□

その他の指数集合もそれぞれのクラスに関して完全であることが知られている.

0.5　高と優勢

これまで算術的階層と指数集合を通して，どんなことが計算 (判定) できるのかを見てきた．今度は関数の発散速度と算術的階層との関係を見ていこう.

$$\lim_{x \to \infty} \frac{x^2}{x} = \infty$$

であることから, x よりも x^2 の方が速く発散する．すべての n について x^n よりも $\exp(x)$ の方が速く発散する．$\exp(n)$ よりも $n!$ の方が速い．では，どこまで発散速度を大きくできるだろうか.

計算論では比よりも大きいかどうかの方が重要で，証明の使い勝手のためにも，以下の概念を用意しておく.

定義 0.5.1. $f, g : \omega \to \omega$ とする．関数 g が関数 f より優勢 (dominate) であるとは，有限個を除いて $f < g$ となることをいう．関数 g が関数 f に優る (majorize) とは，常に $f < g$ となることをいう.

有限個の違いが問題になることはほとんどないので，ほぼ同じ概念であると思って良い．文献によっては，dominate の意味で majorize を使うこともある．証明を書くときには，両方の概念があると短く厳密に書けるので良い.

各計算可能関数 $f : \omega \to \omega$ に対して，f より優勢な計算可能関数 g が存在する．単に，$g(n) = f(n) + 1$ とすればよい．一方，すべての計算可能関数より優勢な計算可能関数は存在しない．すべての計算可能関数より優勢な関数を計算できることが，算術的階層を使って特徴づけられる.

定義 0.5.2. A が高い (high) とは，$\emptyset'' \leq_T A'$ であることを言う.

high は adjective として使われるときもあれば，集合族を表すときもあり，$A \in \mathrm{high}$ などと書く．A が計算可能であれば，$A' \equiv_T \emptyset' <_T \emptyset''$ であるから，$A \notin \mathrm{high}$ である．$A = \emptyset'$ であれば, $A'' \equiv_T \emptyset''$ であるから, $\emptyset' \in \mathrm{high}$ である．$\emptyset' \leq_T A$ であれば, $\emptyset'' \leq_T A'$ なので，$A \in \mathrm{high}$ である．実は $\emptyset' \not\leq_T A$ で $A \in \mathrm{high}$ となりうることが知られている．よって，「高い」とは高い計算可能性を表す概念で，$\geq_T \emptyset'$ よりも真に広い概念である.

定理 0.5.3 (High Domination Theorem, Martin, 1966b)**.** A が高いことと，すべての計算可能関数より優勢な関数 $f \leq_T A$ が存在することは同値.

証明. Tot と $\emptyset''$ は共に Σ_2 完全集合だから，$\mathrm{Tot} \equiv_T \emptyset''$.
($\Rightarrow$). $\mathrm{Tot} \leq_T A'$ より，$\lim_s g(e,s) = \mathrm{Tot}(e)$ となる A-計算可能な $\{0,1\}$-値関数 g が存在する．この g を使って，各 s について $f(s)$ を次のように定義する．すべての $e \leq s$ について，

$$t(e) = (\mu t > s)[g(e,t) = 0 \ \vee \ (\forall x \leq s)\Phi_{e,t}(x)\downarrow]$$

として，

$$f(s) = \max\{t(e) \ : \ e \leq s\}.$$

まず，$t(e)$ が存在することに注意しよう．なぜなら，Φ_e が全域関数でなければ，$\lim_t g(e,t) = 0$ である．よって，f は全域関数. g が A-計算可能であるから，$f \leq_T A$.

次に f はすべての計算可能関数より優勢であることを示そう．各 $e \in \mathrm{Tot}$ に対して，十分大きな s に対して $\Phi_e(s) \leq f(s)$ であることを示す．$e \leq s$ と仮定して良い．十分大きな s に対しては，$g(e,t) = 1$ である．よって，$\Phi_{e,t}(s)\downarrow$ となる最小の t よりも $t(e)$ の方が大きい．これより，$\Phi_e(s) < t(e) \leq f(s)$.

($\Leftarrow$). すべての計算可能関数よりも優勢な関数 $f \leq_T A$ が存在したとする．$\mathrm{Tot}(e)$ の近似関数として A-計算可能な関数 $g(e,s)$ を次のように定義する：

$$g(e,s) = \begin{cases} 1 & (\forall x \leq s)\Phi_{e,f(s)}(x)\downarrow \\ 0 & \text{それ以外} \end{cases}$$

もし Φ_e が全域関数ならば，$\theta_e(y) = (\mu s)(\forall x \leq y)\Phi_{e,s}(x)\downarrow$ も全域関数である．よって，f は θ_e よりも優勢．すなわち，有限個の s を除いて $g(e,s) = 1$.

もし Φ_e が全域でなければ，ある $y \in \omega$ について $\Phi_e(y)$ は定義されないので，すべての $s \geq y$ について $g(e,s) = 0$. □

0.6 低と一般低

定義 0.6.1. A が低い (low) とは，$A' \equiv_T \emptyset'$ であることを言う．A が一般低 (generalized low, GL_1) とは，$A' \equiv_T A \oplus \emptyset'$ であることを言う.

定義 0.6.2. 部分関数 θ が部分関数 ψ に対して優勢であるとは，有限個の x を除いて，

$$\psi(x)\downarrow \Rightarrow \psi(x) < \theta(x)$$

となることを言う.

定理 0.6.3. A が一般低 (GL$_1$) であることと，ある $f \leq_T A \oplus \emptyset'$ がすべての A-部分計算可能関数に対して優勢であることが同値.

証明. $(\Rightarrow)$.

$$f(x) = \max\{\Phi_e^A(x) \ : \ e \leq x \wedge \Phi_e^A(x)\downarrow\}$$

とおけば，$f \leq_T A' \equiv_T A \oplus \emptyset'$ であり，各 $\Phi_e^A(x)$ よりも f は優勢である.

$(\Leftarrow)$. $\psi(x) \simeq \mu s.\Phi_{x,s}^A(x)\downarrow$ は A-部分計算可能関数である．この ψ に優る関数 $f \leq_T A \oplus \emptyset'$ を用意する．すると，

$$\Phi_x(x)^A\downarrow \iff \Phi_{x,f(x)}^A(x)\downarrow$$

よって，$x \in A'$ は $A \oplus \emptyset'$ で判定できる. □

0.7　計算可能な上界

定義 0.7.1. A が計算可能な上界を持つとは，各 $g \leq_T A$ がある計算可能関数な上界を持つことを言う．A が計算可能な上界を持たないとき，A の Turing 次数は超免疫 (hyperimmune) であると言う．A が計算可能な上界を持つとき，A の Turing 次数は超免疫離 (hyperimmune-free) であると言う.

定理 0.7.2. $A \in \Delta_2^0$ が計算不可能であれば，A は計算可能な上界を持たない.

証明. (A_s) を A の計算可能な近似とすると，

$$g(s) = \mu t \geq s.A_t \upharpoonright s = A \upharpoonright s$$

は全域である．また，$g \leq_T A$ である.

g が計算可能な上界 f を持ったと仮定して，A が計算可能であることを示そう．各 n と $s > n$ に対して，ある $t = g(s) \in [s, f(s))$ が存在して $A_t(n) = A(n)$ となる．十分大きな s に対しては，すべての $u \geq s$ に対して $A_u(n) = A_s(n)$ である．よって，$A(n)$ を計算するには，すべての $u \in [s, f(s))$ で $A_u(n) = A_s(n)$ となる $s > n$ を探せば，$A_s(n) = A(n)$ となる. □

0.8　応用例

高の概念が必要になる例を解析学から見てみよう.

以下の事実は解析学においてしばしば使われる.

命題 0.8.1. すべての関数 $f : \omega \to \mathbb{Q}$ で $\sum_n f(n) < \infty$ となるものに対して，単調増加な非有界関数 $g : \omega \to \omega$ が存在して，$\sum_n f(n)g(n) < \infty$.

和が有限であれば，それよりも速く発散する関数で，和が有限であるものが存在することを意味している．この f, g に計算可能の条件を課すと，この事実は成り立たなくなる．その違いを見るために，この証明を見ておこう．

証明. すべての $k \in \omega$ に対して，

$$\sum_{n \geq a_k} f(n) \leq 2^{-k}$$

となるように，狭義単調増加な数列 $\{a_k\}$ をとる．$a_k \leq n < a_{k+1}$ となる n に対して $g(n) = k$ とおく．すると，

$$\sum_{n \geq a_0} f(n)g(n) = \sum_k \sum_{a_k \leq n < a_{k+1}} f(n)g(n) \leq \sum_k k2^{-k} < \infty.$$

最後の収束は等比級数の和との比較から分かる． □

f が計算可能であれば，g は計算可能な関数として取れるだろうか．実は取れないことが分かっている．

注意 0.8.2. $\alpha = \sum_n f(n)$, $\beta = \sum_n f(n)g(n)$ とおき，α が 1 ランダムとなるように f を取る．$\beta \restriction m$ が分かれば，$\sum_{n \leq a} f(n)g(n) \leq 2^{-m}$ となる $a \geq m$ が計算でき，$\sum_{n \geq a} f(n)g(a) \leq \sum_{n \geq a} f(n)g(n) \leq 2^{-m}$ より，$\sum_{n \geq a} f(n) \leq 2^{-m-\log g(a)}$. すなわち，$m + K(m)$ 桁で少なくとも $m + \log g(m)$ 桁の α が表現できる．これは α が 1 ランダムであることに反する．

このことは，合計が有限という制限の中で収束がどれだけ遅くできるかが，計算可能性に関わる問題であることを意味している．この事実と前述の高の概念とは次のような関係が知られている．

定理 0.8.3 (Dobrinen-Simpson 2003, Kjos-Hanssen-Miller-Solomon 2006, Bienvenu-Miller 2012). $A \in 2^\omega$ に関して以下は同値.

(1) A は一様にほとんど至る所優勢 *(uniformly almost everywhere dominating)*，すなわち，A 計算可能な関数 f が測度 1 の X に対してすべての X 計算可能な関数より優勢.

(2) すべての $\emptyset'$ 計算可能な関数 $f : \omega \to \mathbb{Q}$ で $\sum_n f(n) < \infty$ となるものに対し，A 計算可能な関数 $g : \omega \to \mathbb{Q}$ が存在して $\sum_n g(n) < \infty$ かつ $f \leq g$.

(3) すべての 2^ω 上の Π_2 集合が，同じ測度の Σ_2^A 集合を含む.

(4) すべての A ランダム列は 2 ランダム．

数学の定理の計算可能性は，Turing 次数を通して詳細に解析できるようになる．特に確率論のような現実世界への応用が深い分野では，計算可能性は特に重要になるが，確率論の計算可能性を調べていけば，自然にランダム性に焦点が当てられることになるだろう．

0.9　参考文献

まず，計算の理論 (theory of computation) と呼ばれる分野，すなわち，Turing 機械などの計算モデルの定義を行って，停止問題の計算不可能性くらいまでを取り扱っている著書を紹介する．

(1)『コンピュータと数学 (現代基礎数学)』高橋正子，朝倉書店，2016.
(2)『C 言語による計算の理論 (Computer Science Library)』，鹿島亮，2008.
(3)『計算理論の基礎』Michael Sipser，共立出版，2008.
(4)『オートマトン言語理論』John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman，サイエンス社，2003.

(1) や (2) から取り組んでみるのが楽で良いと思う．きちんと勉強したい人はその後に (3) や (4) に挑戦しよう．

次に，計算可能性理論 (computability theory) と呼ばれる分野，Turing 還元に始まる計算不可能性の分野の著書を紹介しよう．この分野の日本語での適当な文献を勧めることは難しい．

(1) ”Computability Theory (Student Mathematical Library)”, Rebecca Weber, AMS, 2012.
(2) ”Turing Computability: Theory and Applications (Theory and Applications of Computability)”, Robert I. Soare, Springer, 2016.
(3) ”Algorithmic Randomness and Complexity”, Downey and Hirschfeldt, Springer, 2010.
(4) ”Recursively Enumerable Sets and Degrees”, Robert I. Soare, Springer, 1987.

(1) は非常に読みやすいので，まず分野の外観をするのにお勧め．(2) はこれから学ぶのであれば外せない文献になると思う．(3) のメイントピックはランダムネスであるが，計算可能性理論の部分もかなり詳しく書かれており，読みやすい．その上でも，(4) を外すことはできず，最終的には (4) を読むことになると思う．