Title	ゲームと戦略の計算可能性について
Sub Title	On the computability of games and strategies
Author	中山, 幹夫
Publisher	慶應義塾経済学会
Publication year	1999
Jtitle	三田学会雑誌 (Keio journal of economics). Vol.91, No.4 (1999. 1) ,p.592(38)- 615(61)
JaLC DOI	10.14991/001.19990101-0038
Abstract	
Notes	論説
Genre	Journal Article
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=AN00234610-19990101-0038

「三田学会雑誌」91巻4号（1999年1月）

ゲームと戦略の計算可能性について[1]

中 山 幹 夫

1. は じ め に

経済の一般均衡や最適政策などについて論ずる場合，その存在や経済学的特性などの分析に比べて，それらがどのような手順や手続きのもとで実際に達成されうるのかという手続き的側面の分析にはかならずしも同程度の精力が注がれてきたわけではない。それは，もちろん，存在しないものについてはこれを論じるわけにはいかないという自明の理とともに，「存在するものについては少なくとも原理的にはそこに到達することができるはずだ」という楽観的な信念が背後にあるからであろう。

人間の合理的行動を形式レベルにおいてより手続き的に記述，分析する理論であるゲーム理論においてさえ，同様の傾向がみられる。ただ，とくに1980年代の無限回繰り返しゲームの研究の中で，記憶，認識などの戦略の実行に不可欠な要素とともに，**戦略の複雑さ**（strategic complexity）とは何かについての考察がなされているのは，ゲーム理論本来の役割のひとつに立ち返る研究として特筆に値する[2]。そこでは，戦略を実行する**有限オートマトン**の最小サイズ，つまり最小の状態数として繰り返しゲームの戦略の複雑さが特徴づけられている。有限オートマトン（とくに，**ムーア・マシン**）とは，前もって指定しておいたいくつかの行動だけを状態に応じて実行するように設計された自動機械であり，一定の明確な手順を踏んで実行される意思決定を記述するのに適切なモデルのひとつである。また，完全な合理性を備えた経済主体やゲーム・プレイヤーは，このような手順を省略して臨機応変に行動し，目的を達成することができると事実上仮定されていることから，有限オートマトンはいわゆる**限定合理性**（bounded rationality）のモデルのひとつでもある[3]。有限オートマトン

（1） 本研究に与えられた1998年度慶應義塾学事振興資金からの補助に対し，深く感謝します。また，レフェリーからいただいた貴重なアドバイスにも感謝いたします。

（2） Kalai, E., and Stanford, W.(1988). Finite rationality and interpersonal complexity in repeated games. *Econometrica* 56, 2, 397-410.

マトンとしての限定合理的戦略によるゲームの均衡分析は，80年代のゲーム理論のひとつの重要なテーマであった。これについては，Kalai (1990) やMarks (1992) の詳細なサーベイ論文がある。

本稿で対象とするのは，しかし，このような有限オートマトンではない。有限オートマトンでは，一連の手続きを踏んで実行しうる政策や戦略の限界に到達することはできないことが知られている。たとえば「存在するものはそこに到達できる」という信念の妥当性は，オートマトンによる手順のもとでは確かめることはできない。オートマトンでは不可能だったとしても，ほかの手順で可能となることがありうるからである。この理由から，本稿ではオートマトンより一般的な**チューリング・マシン** (Turing machine) による**計算可能性** (computability) をとりあげ，一連の手順や手続きの計算可能性をもって**手続きの合理性** (procedural rationality) を意味するものと考える。

チューリング・マシンはもともとヒルベルトによる数学の形式化というプログラムの影響のもとで，英国の論理学者 A. M. チューリングが，計算という人間の行為を原始的ないくつかの基本要素に分解し，これらを一連の手順＝アルゴリズムに再編成して実行する抽象的計算機として考案したものである。今日では数学基礎論やコンピュータ・サイエンスだけでなく，AI や認知科学，哲学などに大きな影響を及ぼしているが，(4) ゲーム理論においては Binmore (1987) の影響もあって，認識の問題や戦略の計算可能性についての考察に注意が向けられるようになった。重要なことは，チューリング・マシンによる計算可能性はたんに一般的であるだけでなく，他の知られているいくつかの外見上異なった計算可能性の概念と本質的に同等であることがわかっており，しかも，(無理数を除けば) 現在にいたるまでこれより広い計算可能性の概念は知られていないということである。この意味において，チューリング・マシン (およびこれと同等なアルゴリズム) によって計算可能でないものは，それを計算する手順は少なくとも現在のところ，存在しないということができる。

このように，計算可能性の理論にもとづけば，手続き的合理性に内在する限界をみきわめることが可能になる。最近の文献に現われたいくつかの考察を概観してみると，まず，Anderlini (1990) は，チューリング・マシンとして定義された相手のプレイヤー (マシン・プレイヤー) が正しく最適反応をプレイするという意味で合理的か否かを決定する計算可能な手続きは存在しないこと；Canning (1992) は，2人ゲームにおいて，合理的でつねに決定を下すことのできるマシン・プレイヤー達が Nash 均衡に到達するのはあるサブクラスに限られることを示している。また，Prasad (1991) は，戦略の数が可算無限個であるような有限人ゲームが Nash 均衡をもつかどうかを決定するアルゴリズムは存在しない (どんな方法も計算不可能である) ことを明らかにしている。さらに，Knoblauch (1994) および Nachbar-Zame (1996) は，囚人のジレンマ型の無限回繰り返しゲーム

---

(3) 限定合理性については，Simon, H. (1972). Theories of bounded rationality. In B. McGuire and R. Radner, eds: *Decision and Organization.* Amsterdam: North-Holland.

(4) たとえば，英国の数理物理学者ペンローズによる次の著作がある。Penrose, R. (1989) *The Emperor's New Mind.* Oxford: Oxford University Press.

には，いかなる最適反応も計算可能ではないような戦略が存在することを証明している。

より古典的な研究としては，Rabin (1957) による完全情報2人**勝ち負けゲーム** (win-lose game) の必勝戦略の計算不可能性，また，Jones (1982) の**算術ゲーム** (arithmetical game) における必勝戦略の計算不可能性や，どちらのプレイヤーが必勝戦略をもつかを決定する手続きの計算不可能性などの結果がある。とくに，この Jones による Rabin (1957) のゲームの再定式化は，きわめて単純なゲームのほとんど自明な必勝戦略が計算不可能であることを示しており，手続き的合理性の限界を直截に物語るものとして重要である。

こうして，「存在するものはそこに到達できる」という楽観的な信念は，計算可能性というフィルターを通すことによって再検討を余儀なくされることになる。本稿の目的は，このような手続き的合理性の基礎理論としての計算論の初等的解説をとおして，ゲームと戦略の計算可能性を上述の Jones や Rabin による古典的なゲームに限定して考察することである。

2．決定不能なゲーム

Jones (1982) によって再定式化された Rabin (1957) のゲームとは，次のように単純明快な完全情報2人ゲームである。ある集合 $S \subseteq \{0, 1, 2, \ldots\}$ が与えられており，S もその補集合 S^c もともに無限集合であるとする。プレイヤー1が最初に自然数 $x_1 \geq 0$ を選び，それを知ってプレイヤー2が自然数 $x_2 \geq 0$ を選ぶと，$x_1 + x_2 \in S$ およびそのときに限りプレイヤー1が勝つ。

S^c は無限集合だから，後手であるプレイヤー2が必勝戦略をもつことは明らかである。任意の x_1 に対して $x_1 + x_2 \in S^c$ をみたす x_2 がつねに存在するので，これを $x_2 = f(x_1)$ と書けばこの関数 f が必勝戦略であることはいうまでもない。問題は，関数 f を一定の手順＝アルゴリズムとして記述しようとするときに起こる。特別に選ばれた集合 S に対しては，与えられた x_1 から x_2 を求めるアルゴリズムは存在しない。(5) つまり，完全な合理性をもつ人間はともかく，たとえばこの必勝戦略をコンピュータ・プログラムで記述しようとしても，その試みは失敗するのである。

この特殊な集合 $S \subseteq \{0, 1, 2, \ldots\}$ とは，**単純集合** (simple set) とよばれるものである。(6) 用語の厳密な定義は後で述べるが，単純集合 S 自身は**帰納的可算** (recursively enumerable)，つまりメンバーを列挙するための手順＝アルゴリズムをもっているのに対し，補集合 S^c についてはそのどんな無限部分集合も帰納的可算ではない。このとき，S^c は**免疫性である** (immune) といわれる。それゆえ，$x_1 + x_2$ が S^c のメンバーか否かを決定する手続きは存在しない。このことを，

(5) 「少なくとも現在のところ」という限定句つきの意味である。以下，アルゴリズム（手順，手続き，プログラム）は存在しない，という表現はすべてこの意味におけるものとする。

(6) 定義と存在証明は，Post. E. L. (1944)。

（決定）問題（decision problem）'$x_1+x_2 \in S^c$' は**決定不能**であるという。こうして，プレイヤー2は，必勝戦略をもってはいるがそれを一連の手順として記述することはできない。

この事実を，チェスの必勝戦略に関するツェルメロの定理[7]と混同しないように注意しよう。チェスの場合は，有限ゲームであるから必勝戦略をもつとすれば，それを計算することは原理的には可能である。ただ，計算量が多すぎて今日まで誰も知らないだけであるのに対し，ラビンのゲームでは（今日までの知識では）計算そのものが不可能なのである。

単純集合 S は上で触れたように帰納的可算であるから，Jones（1981, Lemma 2）によれば，整数係数のある多項式 $Q(n, x_1, x_2, x_3, x_4)$ を用いて次のように特徴づけることができる：

$$\text{すべての } n \text{ に対し，} n \in S \Leftrightarrow \exists x_1 \forall x_2 \exists x_3 \forall x_4 [Q(n, x_1, x_2, x_3, x_4) \neq 0].$$

すると，$n = x_1 + x_2$ と書いて添字の番号をつけ変えれば，

$$\text{すべての } x_1 \text{ と } x_2 \text{ に対し，}$$

$$x_1 + x_2 \in S \Leftrightarrow \exists x_3 \forall x_4 \exists x_5 \forall x_6 [Q(x_1 + x_2, x_3, x_4, x_5, x_6) \neq 0]$$

のように，ラビンのゲームと多項式の値をペイオフとするゲームが互いに関連づけられる。つまり，完全情報のもとでプレイヤー1と2が交互に $x_1, x_2, \ldots$ と選んでいき，Q の値がゼロになればプレイヤー2の勝ち，そうでなければプレイヤー1の勝ちとするゲームである。これを**算術ゲーム**という。

この算術ゲームの勝ち負けはラビンのゲームと同等であるから，どのプレイヤーも計算可能な必勝戦略をもたない。Jones（1982）ではさらに，算術ゲームにおいては，どのプレイヤーが必勝戦略をもつかは決定不能であること，プレイヤー2が必勝戦略をもつけれどもそれが証明できないような算術ゲームの存在などが示されている。これらについて考察する前に，計算可能性の基本的な概念，用語の定義および結果について述べることにしよう。

3. 計算可能関数

3. 1　部分帰納的関数

$N = \{0, 1, 2, \ldots\}$ を自然数の集合とし，f を $W \subseteq N^n$ から N への関数とする。定義域 $W = Dom(f)$ が N^n の部分集合であるという意味で f を一般に n 変数の**部分関数**（partial function）という。関数 f が計算可能であるとは何を意味するかをまず明らかにすることから始めよう。直観的には，ある有限の手順やアルゴリズムがあって f の値を有限個のステップで計算できれば，f は計算可能である。しかし，これでは f が**計算可能でない**ことを確定することはできない。そのた

（7）　一方のプレイヤーが必勝戦略をもつか，またはどのプレイヤーも引き分けに導く戦略をもっている（Zermelo, E. (1912)）。

めには，いかなる手順やアルゴリズムも役に立たないことを確かめなければならないからである。

しかし，次に述べる**チャーチの提言** (Church's Thesis)(8) を受け入れることによって，そのような定義は事実上確立しているといってよい。すなわち，

直観的に明確な手順で定義された計算可能な部分関数のクラス
はチューリング・マシンで計算できる関数のクラスに一致する。

これは提言であって，定理ではないことに注意しよう。チューリング・マシンで計算できなくても計算可能といえる関数 f が将来発見される可能性は否定できないからである。しかし，すでに述べたように，現在までのところそのような関数は知られておらず，定義されているすべての計算可能関数のクラスはチューリング・マシンで計算可能であることがわかっている。

この提言を受け入れれば，次の定理によって計算可能関数とは以下に定義する部分帰納的関数であるとみなすことができる。

定理 3．1．1　チューリング・マシン計算可能関数のクラスは，部分帰納的関数のクラスと一致する。

定義 3．1．1　**部分帰納的関数** (partial recursive functions) のクラスとは，以下の 1 から 4 までの 4 条件をみたす最小のクラスである：

1．**基本関数** (basic functions) $\boldsymbol{O}, \boldsymbol{S}, \boldsymbol{T}_i^n$ が含まれる。

ここで，$\boldsymbol{O}$ はすべての自然数 x を 0 に写す関数 $\boldsymbol{O}(x)=0$；$\boldsymbol{S}$ はすべての自然数 x に 1 を加える関数 $\boldsymbol{S}(x)=x+1$；また，$\boldsymbol{T}_i^n$ は射影 $\boldsymbol{T}_i^n(x_1, \ldots, x_i, \ldots, x_n)=x_i$ である。

2．**合成** (composition) に関して閉じている；すなわち，$\gamma_1, \ldots, \gamma_m, \psi$ が含まれるならば，次のように定義される ϕ も含まれる。

$$\phi(\boldsymbol{x}) \simeq \psi(\gamma_1(\boldsymbol{x}), \ldots, \gamma_m(\boldsymbol{x}))$$

ここに $\boldsymbol{x}=(x_1, \ldots, x_n)$ である。また記号 $\simeq$ は ϕ と ψ が関数として等しいことを意味する。それゆえ，両辺が定義されていれば値は等しく，一方が定義されていなければ他方も定義されていない。

3．**原始帰納的定義** (primitive recursion) に関して閉じている；すなわち，ψ と γ が含まれるならば次のように定義される ϕ も含まれる。

(8) チャーチ=チューリングの提言，さらにはより文献に忠実に，チャーチ=チューリング=マルコフの提言という人もいる。たとえば，Bridges, D. S. (1994) など。

$$\phi(\boldsymbol{x}, 0) \simeq \psi(\boldsymbol{x})$$
$$\phi(\boldsymbol{x}, y+1) \simeq \gamma(\boldsymbol{x}, y, \phi(\boldsymbol{x}, y))$$

4．**最小化**（minimalization）に関して閉じている；すなわち，ψ が含まれるならば，次のように定義される ϕ も含まれる。

$$\phi(\boldsymbol{x}) \simeq \mu y[(\forall z \leq y)(\psi(\boldsymbol{x}, z)\downarrow) \wedge \psi(\boldsymbol{x}, y) \simeq 0].$$

ここで，記号 $\psi(\boldsymbol{x}, z)\downarrow$ は変数（$\boldsymbol{x}, z$）に対して関数 ψ が定義されていることを意味する。

1から3までの条件は，明確な手順で定義できる関数を記述していることが直ちにわかる。条件4の最小化は，$\boldsymbol{x}$ に対して $\psi(\boldsymbol{x}, y)=0$ をみたす最小の y を対応させるための手順を定義しており，**unrestricted $\boldsymbol{\mu}$-recursion** といわれることもある。μ は［　　］で示された関係をみたす最小の変数 y を対応づける作用素で，**$\boldsymbol{\mu}$-作用素**（μ-operator）とよばれている。

関数 ψ が全域で定義されていても，μ-作用素は全域で定義された関数 ϕ を生成するとは限らないことに注意しよう。たとえば，$\psi(x, y)=|x-y^2|$ とすると

$$\phi(x)=\begin{cases}\sqrt{x} & x\text{ が完全平方数のとき} \\ \text{未定義} & \text{そうでないとき}\end{cases}$$

となる。部分帰納的関数がとくに N^n の全域で定義されているとき，これをたんに**帰納的関数**（recursive function）という。

以上の条件のうち4を要請しないクラスは，**原始帰納的関数**（primitive recursive function）とよばれる関数のクラスである。原始帰納的関数は，N^n の全域で定義されていることがわかる。

こうして，定理3. 1. 1とチャーチの提言は，計算可能関数であるか否かを確かめるために実際にチューリング・マシンのプログラムを設計する必要を省いてくれるばかりでなく，手順が形式的に記述されてなくても直観的に明確に定義されていれば，その手順で計算される関数はチューリング・マシン計算可能であるとみなすことも可能にする。

関数の計算可能性にもとづけば，関係や述語の計算可能性を定義することができる。ある $\boldsymbol{x}=(x_1, \ldots, x) \in N^n$ に対して性質 $M(\boldsymbol{x})$ が成立し，残りの $\boldsymbol{x}$ については成立しないとき，$M(\boldsymbol{x})$ を**関係**（relation）ないし**述語**（predicate）という。もし，$M(\boldsymbol{x})$ の**特性関数**（characteristic function）が計算可能ならば，述語 $M(\boldsymbol{x})$ は計算可能または**決定可能**（decidable）であるという。すなわち，関数

$$c_M(\boldsymbol{x})=\begin{cases}1 & M(\boldsymbol{x})\text{であるとき} \\ 0 & M(\boldsymbol{x})\text{でないとき}\end{cases}$$

の計算可能性で，述語 $M(\boldsymbol{x})$ の計算可能性を定義する。決定可能な述語は，また，帰納的である

ともいわれる。

3. 2 URM プログラム

実際に計算を実行するプログラムをみておくことにより，形式的な手順を具体的に理解しておこう。ただし，チューリング・マシンそのものより，それと同等な簡略化された URM（unlimited register machine）のプログラムをとりあげることにしよう。[9] チューリング・マシンのプログラムをコンピュータの機械語にたとえると，URM はフォートランやベーシックなどのより人間に理解しやすいプログラムにたとえることができるからである。

URM にはまずその名のとおり，無限個のレジスター $R_1, R_2, R_3, \ldots$ があると仮定される。各レジスター R_i には，自然数 r_i が格納される。URM プログラムとは次の 4 種類の**命令**（instruction）の有限個の列 $I_1, I_2, \ldots, I_s$ である：

Zero	$Z(n)$	：$n \in N$ に対し，$r_n := 0$
Successor	$S(n)$	：$n \in N$ に対し，$r_n := r_n + 1$
Transfer	$T(m, n)$	：$m, n \in N$ に対し，$r_n := r_m$
Jump	$J(m, n, q)$	：$m, n, q \in N$ に対し，$r_m := r_n$ ならば命令 q を実行する； $r_m :\neq r_n$ ならば次の命令を実行する。

データの**初期配列**（initial configuration）とは，各レジスター R_i に自然数 a_i が格納されていることを示す列 $a_1, a_2, a_3, \ldots$ である。URM はプログラムと初期配列が与えられると，各命令を順に実行していく。ジャンプ命令に出会うと，条件をチェックして指定された命令にジャンプするか，またはジャンプせずに次の命令を実行する。命令の実行の過程で，次の命令やジャンプした後の命令 q が存在しないとき，およびこのときに限って URM は停止する。このときの各レジスターの値の列，$r_1, r_2, r_3 \ldots$ を**最終配列**（final configuration）という。URM は停止するとは限らない。例えば，すべての自然数をプリントするプログラムや，ループに入り込んだプログラムなどはもちろん停止しない。

URM プログラムの例として，和 $x+y$ の計算を考えよう。これは，初期配列 $x, y, 0, 0, 0, \ldots$ と 4 つの命令 $I_1 : J(3, 2, 5)$，$I_2 : S(1)$，$I_3 : S(3)$，および $I_4 : J(1, 1, 1)$ をこの順で実行するプログラムで達成できる。

さて，初期配列 $a_1, a_2, a_3, \ldots$ は事実上，ある有限の n に対しそれ以降の a_{n+1} はすべてゼロであるとみなしてもよいので，これを無視すれば $a_1, a_2, \ldots, a_n$ のように有限の列として書くことができる。そこで，プログラム $\boldsymbol{P}$ と初期配列 $a_1, a_2, \ldots, a_n$ に対し，

（9） URM の詳細と，それにもとづいた計算論の入門書としては，Cutland, N. J. (1980) がよい。また，チューリング・マシンの詳細については Davis, M. (1982) または Bridges (1994) など。

$\boldsymbol{P}(a_1, a_2, \ldots, a_n)\downarrow$　　URMが停止するとき

$\boldsymbol{P}(a_1, a_2, \ldots, a_n)\uparrow$　　URMが停止しないとき

のようにあらわそう。停止したときの計算結果はつねに最初のレジスター R_1 に格納されるものとする。つまり，最終配列において $r_1=b$ ならば

$$\boldsymbol{P}(a_1, a_2, \ldots, a_n)\downarrow b$$

と書き，これを，計算 $\boldsymbol{P}(a_1, a_2, \ldots, a_n)$ は b に収束するという。

以上の準備のもとに，URM計算可能関数を次のように定義することができる。

定義3.2.1 f は N^n から N への部分関数とする。

（1）プログラム $\boldsymbol{P}$ が関数 f をURM計算するとは，すべての $a_1, a_2, \ldots, a_n, b$ に対して，

$$\boldsymbol{P}(a_1, \ldots, a_n)\downarrow b \Leftrightarrow (a_1, \ldots, a_n)\in Dom(f) \text{ and } f(a_1, \ldots, a_n)=b$$

が成立することである。

（2）関数 f がURM計算可能とは，f をURM計算するプログラムが存在することである。

こうして，定理3.1.1を経由すれば，部分帰納的関数とはURM計算可能関数にほかならないことがわかる。

3.3 ゲーデル数

すべてのURMプログラムの集合を Σ としよう。Σ から自然数の集合 N への関数 γ を，全単射で γ も γ^{-1} もともに計算可能となるようにつくることができれば，URMプログラムを自然数に符号化し，逆に自然数をURMプログラムに復元することができる。このとき，$\boldsymbol{P}\in\Sigma$ に対して $\gamma(\boldsymbol{P})$ をプログラム $\boldsymbol{P}$ の**ゲーデル数**（Gödel number）という。

ゲーデル数 $\gamma(\boldsymbol{P})$ をつくる手順を以下に示そう。まず，URMプログラムの命令の全体を I とし，4種類の命令を，$m, n, q\in N_+=\{1, 2, \ldots\}$ に注意して，次の関数 $\beta: I\rightarrow N$ によって符号化する。

$$\begin{aligned}\beta(Z(n)) &= 4(n-1)\\ \beta(S(n)) &= 4(n-1)+1\\ \beta(T(m, n)) &= 4\pi(m-1, n-1)+2\\ \beta(J(m, n, q)) &= 4\zeta(m, n, q)+3\end{aligned}$$

ここに，$\pi: N\times N\rightarrow N$ と $\zeta: N_+\times N_+\times N_+\rightarrow N_+$ は次のように定義される全単射である。

$$\begin{aligned}\pi(m, m) &= 2^m(2n+1)-1\\ \zeta(m, m, q) &= \pi(\pi(m-1, n-1), q-1)\end{aligned}$$

π はもちろん計算可能であるから ζ も計算可能。π^{-1} については，$x=2^m(2n+1)-1$ ならば，m は $x+1$ を一意に素因数分解したときの1番目の素数2の指数である。これを $\pi_1(x)=(x+1)_1$ と書く

と，

$$\pi^{-1}(x)=(\pi_1(x),\ \pi_2(x))\ ;\ \pi_2(x)=(1/2)((x+1)/2^{\pi1(X)}-1)$$

となるから，π^{-1} は計算可能である。ζ^{-1} については，

$$x=\zeta(m,\ n,\ q)=2^{\pi(m-1,\ n-1)}(2(q-1)+1)-1$$

とすると $\pi(m-1,\ n-1)=\pi_1(x), (q-1)=\pi(x)$ だから，$m-1=\pi(\pi(x)), n-1=\pi_2(\pi_1(x))$. ゆえに，

$$\zeta^{-1}(x)=(\pi_1(\pi_1(x))+1,\ \pi_2(\pi_1(x))+1,\ \pi_2(x)+1)$$

となって，ζ^{-1} もまた計算可能となる。

こうして，$\beta:I\rightarrow N$ は任意の URM 命令のゲーデル数を与える。さて，URM プログラム $\boldsymbol{P}$ は有限個の URM 命令の列 $I_1, I_2, \ldots, I_s$ であるから，列 $\beta(I_1), \beta(I_2), \ldots, \beta(I_s)$ を自然数に写す全単射 $\tau: \cup_{k>0}N^k\rightarrow N$ で，しかも τ も τ^{-1} も計算可能なものをつくることができれば目的は達成される。このような τ として，

$$\tau(a_1,\ \ldots,\ a_k)=2^{a1}+2^{a1+a2+1}+2^{a1+a2+a3+2}+\cdots+2^{a1+a2+\cdots+ak+k-1}-1$$

をとることができる。これが計算可能であることは明らかである。また，一般に任意の自然数 x に対して，

$$x+1=2^{b1}+2^{b2}+\cdots+2^{bk}$$

をみたす $k\geq 1, 0\leq b_1<b_2<\ldots<b_k$ を一意に決定することができるので，$a_1=b_1, a_{1+1}=b_{1+1}-b_1-1$, $(1\leq i<k)$ とおけば

$$\tau^{-1}(x)=(a_1,\ \ldots,\ a_k)$$

となることから，τ が全単射であることと τ^{-1} が計算可能であることがわかる。このようにして，

$$\gamma(\boldsymbol{P})=\tau(\beta(I_1),\ \ldots,\ \beta(I_s))$$

と定義すれば，関数 $\gamma:\Sigma\rightarrow N$ が求めるものである。

ゲーデル数は URM プログラムやチューリング・マシンの集合だけでなく，一般に有限な対象の集合に対していくつかの方法で定義することができる。ゲーデル自身は，論理式に対してこのような符号化を実行して不完全性定理を導いたことはよく知られている。

重要なことは，ひとつの符号化のもとで，すべての自然数は文字どおり数であると同時に，プログラムとして確定した意味をもつことである。それゆえ，どの URM プログラムも自分自身をデータとして取り込むことは，全く自然な行動であるばかりでなく，基本的に重要な結果に導くことが後で明らかになる。

3．4　標準形定理と万能プログラム

クリーネの**標準形定理**（normal form theorem）とは，次のように計算可能関数がチューリング・マシンや URM プログラムから計算されることを形式的に表現する定理である。(10)

定理 3．4．1　ある原始帰納的関数 $U(x)$ と，各 $n \geq 1$ について原始帰納的述語 T_n が存在して，任意の n 変数部分帰納的関数 φ^n に対して次の 1，2 が成立するような番号 e がある：

$$1.\ \varphi^n(x_1, ..., x_n)\downarrow \ \Leftrightarrow\ \exists y T_n(e, x_1, ..., x_n, y)$$

$$2.\ \varphi^n(x_1, ..., x_n) \simeq \underline{U}(\mu y T_n(e, x_1, ..., x_n, y)).$$

記号 $\varphi^n(\boldsymbol{x})\downarrow$ は $\boldsymbol{x}$ において関数 φ が定義されていることをあらわす。述語 T_n は，「$(y)_1$ は，入力 $x_1, ..., x_n$ のもとでの，ゲーデル数 e の URM プログラムによる $(y)_2$ 以下のステップでの計算結果」[(11)] ということを述べる決定可能な関係で，**T-述語**の名で知られている。それゆえ，この定理は，あるプログラム e が入力 $x_1, ..., x_n$ に対してある数 y を算出するときおよびこのときに限って，関数 φ^n は値をもち，その値は関数 U によって計算結果 $(y)_1$ として y から抽出されることを述べている。このプログラムのゲーデル数 e を，計算可能関数 φ^n の**指標**（index）という。

標準形定理からえられる次の**列挙定理**（enumeration theorem）によれば，すべての計算可能関数を（重複を許して）列挙することが可能になる。

定理 3．4．2　各 $n \geq 1$ について，すべての n 変数部分帰納的関数は計算可能な手順で列挙することができる。すなわち，次の 1，2，および 3 が成立するような列 $\{\varphi_e^n\}_{e \in N}$ がある。

1. 各番号 e について，φ_e^nは n 変数部分帰納的関数である
2. もし ψ が n 変数部分帰納的関数ならば，番号 e が存在して

$$\psi = \varphi_e^n$$

3. $n+1$ 変数の部分帰納的関数 φ が存在して，

$$\varphi(e, \boldsymbol{x}) \simeq \varphi_e^n(\boldsymbol{x}).$$

標準形定理から，$\varphi(e, \boldsymbol{x}) \simeq U(\mu y T_n(e, \boldsymbol{x}, y))$ とおけば証明できる。3 によって，番号 e を指定すれば，計算可能関数 φ_e^n がえられる。もちろん，この e は計算可能関数 $\psi = \varphi_e^n$ の指標である。ひとつの計算可能関数は無限個の指標をもっている。とくに，結果に影響しない無害な命令を一定の手順で付け足していくことにより，もとの関数と同じ関数の指標を無限個，しかも計算可能な方法で生成できることに注意しておこう。

さて，計算可能関数のいくつかの変数を固定した関数も計算可能であることはいうまでもないが，こうしてえられる計算可能関数を計算するプログラムは固定した変数の値から計算可能な手順で見

(10)　証明は Cutland (1980) あるいは Davis (1982) など参照。

(11)　$(y)_i$ は，前節と同じく，$y>0$ を一意に素因数分解したときの i 番目の素数の指数である。また，$(0)_1=0$ とする。なお，クリーネが標準形定理を証明したのは URM プログラムが発表される前であるから，この定理はもともとチューリング・マシンによる計算を前提にしている。

つけ出すことができる。これを示すのが，次の**パラメター定理**あるいは **S_n^m定理**とよばれる定理である。[12]

定理 3．4．3 任意に与えられた m, n に対し，原始帰納的関数 $S_n^m(e, x_1, \ldots, x_n)$ が存在して

$$\varphi_{S_n^m(e,\, x_1,\, \cdots,\, x_n)}(y_1, \ldots, y_m) \simeq \varphi_e(x_1, \ldots, x_n, y_1, \ldots, y_m).$$

こうして帰納的関数 S_n^m によって，データ $x_1, \ldots, x_n$ から作成されるプログラム $S_n^m(e, x_1, \ldots, x_n)$ は，$y_1, y_2, \ldots, y_m$ が与えられれば，もとの関数と同じ値を計算する。

列挙定理では指標が変数の役割を果たし，パラメター定理では変数が指標の役割を演じていることからわかるように，列挙定理とパラメター定理は，指標と変数の間の双対関係を明らかにしている。

列挙定理はもうひとつの重要な事実を明らかにする。条件 3 は

$$\varphi_U(e, \boldsymbol{x}) \simeq \varphi_e^n(\boldsymbol{x})$$

をみたす $n+1$ 変数の計算可能関数 φ_U の存在を記している。この関数 φ_U は，番号 e と変数 $\boldsymbol{x}=(x_1, \ldots, x_n)$ の値を与えられると，指標 e をもつ n 変数計算可能関数の値 $\varphi_e^n(\boldsymbol{x})$ を計算する。つまり，任意の計算可能関数 φ_e^n をシミュレートすることができる。この意味で，関数 φ_U を n 変数計算可能関数に対する**万能関数** (universal function) といい，万能関数を計算する URM プログラムやチューリング・マシンを各々**万能プログラム** (universal program)，**万能チューリング・マシン** (universal Turing machine) という。

万能プログラムは，もちろん，各プログラム e を内臓しているわけではなく，与えられたプログラム e を解読する能力をもっているだけである。しかし，この能力はそのままゲームにおいて与えられた相手の戦略＝プログラムをシミュレートし，それに対する最適反応を実行するという場合に不可欠な役割を果たすことが容易にわかる。[13]

今日普及しているパーソナルコンピュータが，個々のアプリケーション・プログラムをデータとして読み込んだうえで，それを実行できるのはまさにオペレーティング・システムという万能プログラムを備えているからにほかならない。目的にあわせて特別に配線を変えてハードウェアを構成し直すという初期の非効率な計算機から出発し，ゲーデルの仕事に精通していたフォン・ノイマンの，各種のプログラムを符号化してデータと同様に格納しておく**プログラム内蔵計算機** (stored-program computer) を経て，今日に至ってようやくチューリングのアイディアの実現にたどり着いたというわけである。[14]

(12) この証明も Cutland (1980), または，Odifreddi, P. (1992) など参照。

4．決定問題と帰納的可算集合

4. 1　停止問題

3. 1で述べたように，述語 $M(\boldsymbol{x})$ が決定可能であるとは，その特性関数が計算可能となることであった。そうでないとき，述語 $M(\boldsymbol{x})$ は**決定不能**である（undecidable）といわれる。また，次に述べるように，決定可能に準ずるといえるような場合がある。すなわち，

$$g(\boldsymbol{x})=\begin{cases}1 & M(\boldsymbol{x})\text{であるとき}\\ \text{未定義} & M(\boldsymbol{x})\text{でないとき}\end{cases}$$

で与えられる**半特性関数**（partial characteristic function）g が計算可能であるとき，述語 $M(\boldsymbol{x})$ は**半決定可能**（partially decidable）である，とか**半計算可能**（semi-computable）である，などという[15]。決定可能である場合との違いは，$M(\boldsymbol{x})$ でない $\boldsymbol{x}$ に対しては g を計算するどんなアルゴリズムも停止しないという点である。また，決定可能な $M(\boldsymbol{x})$ は，半決定可能であることに注意しよう。それは，$M(\boldsymbol{x})$ でない $\boldsymbol{x}$ に対してはループに入り込むように，特性関数 c_M を計算するもとのプログラムを書きかえることができるからである。さらに，次の事実も決定可能性との強い関連を示している。

定理4．1．1　$M(\boldsymbol{x})$ とその否定 $\neg M(\boldsymbol{x})$ がいずれも半決定可能であるならば $M(\boldsymbol{x})$ は決定可能である。

何故なら，$M(\boldsymbol{x})$ を決定するアルゴリズムの各命令と $\neg M(\boldsymbol{x})$ を決定するアルゴリズムの各命令を交互に実行してゆくアルゴリズムを走らせれば，これは必ず停止するので決定可能となるからである。逆も成立することはいうまでもない。

さて，与えられた述語の決定可能性を問う問題を一般に**決定問題**（decision problem）とよぶことにしよう。標題の**停止問題**（the halting problem）とは，「任意のプログラムが任意の入力に対して停止するか否か」を決定する単一のアルゴリズムの存在を問う有名な決定問題であり，**半決定可**

---

(13)　たとえば，Rubinstein, A. (1998). *Modeling Bounded Rationality.* MIT Press: Cambridge, Massachusetts の第10章や，Anderlini (1990), Knoblauch (1994) など参照。

(14)　Enderton, H. B. (1977). Elements of Recursion Theory. In J. Barwise ed: *Handbook of Mathematical Logic.* North-Holland: Amsterdam にこのような記述がある。

(15)　決定可能な述語は帰納的ともよばれることに対応して，半決定可能な述語は後で導入する帰納的可算という名でよばれることもある。

能であって決定可能ではない問題の代表例として基本的に重要である。

列 $\{\varphi_e\}_{e\in N}$ によってすべての1変数計算可能関数を列挙したものをあらわそう。すると，

定理4．1．2 決定問題‘$\varphi_x(y)\downarrow$’は半決定可能であるが，決定可能ではない。

証明 ‘$\varphi_x(y)\downarrow$’が半決定可能であることは，すべての x を1に写す関数を $\mathbf{1}$ とすれば，計算可能関数の合成によって，その半特性関数

$$g(x, y)\simeq\mathbf{1}(\varphi_U(x, y))$$

が計算可能であることからしたがう。また，もし‘$\varphi_x(y)\downarrow$’が決定可能であったとしたら，その否定‘$\varphi_x(y)\uparrow$’も決定可能であるから

$$f(x)=\begin{cases}1 & \varphi_x(x)\uparrow \text{のとき}\\ \text{未定義} & \varphi_x(x)\downarrow \text{のとき}\end{cases}$$

で与えられる関数 $f(x)$ は計算可能。ゆえに，ある番号 e があって $\varphi_e=f$ となる。すると $\varphi_e(e)\downarrow \Leftrightarrow \varphi_e(e)\uparrow$ となって矛盾する。□

証明の中で矛盾を導いた方法は，**対角化**（diagonalization）とよばれる，カントールにさかのぼる重要な論法である。

次のライスの定理が示すように，実際にはほとんどの問題は決定可能ではない。[16]

定理4．1．3 集合 B をすべての1変数計算可能関数の，空でない真部分集合とすると，決定問題‘$\varphi_e\in B$’は決定可能でない。

つまり，「φ_x は全域で定義されている」とか「$\varphi_x(y)=0$」などのいくつかの関数だけがもっている性質を意味のある性質とよべば，任意に与えられた関数が意味のある性質をもっているかどうかは一般に決定不能である。これがゲームのマシン・プレイヤーに対してもつ含意のひとつは，任意の相手プレイヤーのゲーデル数が与えられたとき，相手の行動＝そのプログラムが計算する関数が定められた性質をもっているか否かを決定できるマシン・プレイヤーは存在しないことである。

4．2 帰納的可算集合

ある自然数がある集合のメンバーであるか否かという決定問題を考えることにより，重要な集合

（16） 証明はたとえば Cutland (1980) または，Odifreddi (1992) など参照。

を定義することができる。

定義 4．2．1 集合 $S \subseteq N$ について，決定問題 '$x \in S$' が半決定可能であるとき，S を**帰納的可算集合** (recursively enumerable set)，あるいは短く **r. e.** という。また，'$x \in S$' が決定可能であるとき，S を**帰納的集合** (recursive set) という。

この定義と定理 4．1．1 より，集合 S が帰納的集合であることと，S および補集合 S^c がともに帰納的可算であることは同値である。

N や空集合 ϕ，有限集合などは帰納的である。帰納的集合はもちろん帰納的可算である。しかし，停止問題が示すように逆は正しくない。集合 K を

$$K := \{x \mid x \in W_x\} = \{x \mid \varphi_x(x)\downarrow\}$$

と定義すると，K は帰納的可算であるが帰納的ではない。集合 K は自分自身のゲーデル数を入力されたときにある値を出力するプログラムの番号の集合で，多くの重要な役割を果たす。

ある計算可能関数 g について，決定問題 '$x \in Dom(g)$' を考えると

$$x \in Dom(g) \Leftrightarrow \mathbf{1}(g(x))$$

であるから集合 $Dom(g)$ は帰納的可算である。このように，帰納的可算集合を**計算可能関数の定義域**として定義することもできる。計算可能関数の列挙定理から，$W_e = Dom(\varphi_e)$ と書くと，すべての帰納的可算集合を

$$W_0,\ W_1,\ W_2,\ W_3,\ \ldots$$

のように（重複を許して）列挙することができる。$S = W_e$ のとき，e を帰納的可算集合 S の指標という。

この事実と次の定理は，帰納的可算集合の標準形定理を与えている。

定理 4．2．1 集合 S が r. e. である必要十分条件は，ある決定可能な述語 $R(x, y)$ に対し，

$$x \in S \ \Leftrightarrow\ \exists y R(x, y)$$

となることである。[17]

証明 S が r. e. ならば，S の指標 e に対して標準形定理 3．4．1 から，

$$x \in S \ \Leftrightarrow\ x \in W_e \ \Leftrightarrow\ \varphi_e(x)\downarrow \ \Leftrightarrow\ \exists y T_1(e, x, y).$$

そこで，$R(x, y) = T_1(e, x, y)$ とおけば R は決定可能で，$x \in S \ \Leftrightarrow\ \exists y R(x, y)$ となる。逆に，決

(17) それゆえ，述語 $\exists y R(x, y)$ は半決定可能である。

定可能な述語 $R(x,y)$ に対し，$x \in S \Leftrightarrow \exists y R(x,y)$ ならば，関数 $g(x) \simeq \mu y R(x,y)$ は計算可能で[18] $S = Dom(g)$ となる。□

帰納的可算集合という名称は，次の定理に由来する。

定理 4．2．2 次の（a），（b）および（c）は同値である。

（a）　S は r. e.

（b）　$S=\phi$, または S はある帰納的関数の値域である。

（c）　S はある部分帰納的関数の値域である。

証明　（a）→（b）だけを示そう[19]。$S=Dom(f) \neq \phi$, f を計算する URM プログラムを $\boldsymbol{P}$，さらに，$a \in S$ として，帰納的関数 g を次のように定義する。

$$g(x,t)=\begin{cases} x & \text{if } \boldsymbol{P}(x)\downarrow \text{in } t \text{ steps} \\ a & \text{otherwise} \end{cases}$$

（全域で定義されていることは明らか。また，計算可能であることはチャーチの提言から。）１変数の帰納的関数 h を，$h(0)=0$ および $z>0$ ならば

$$h(z)=g((z)_1,(z)_2)$$

で定義すると，関数 h と g の値域はともに S に一致する。□

この定理によって，もし S が r. e. ならば，S はある帰納的関数 h の値域 $Ran(h)$ であるから

$$S=Ran(h)=\{h(0), h(1), h(2), h(3), \ldots\}$$

のように，S の各メンバーを残らず（重複を許して）列挙することができる。これが帰納的可算の意味である。

帰納的可算集合のメンバーを列挙する帰納的関数の性質をいくつかあげておこう。

定理 4．2．3　（a）無限集合 S は帰納的可算集合 $\Leftrightarrow$ S は１対１帰納的関数の値域である。（b）無限集合 S は帰納的集合 $\Leftrightarrow$ S は単調増加帰納的関数の値域である。（c）任意の帰納的可算無限集合は帰納的無限部分集合をもつ。

（18）　部分帰納的関数の定義 3．1．1 の 4 で，$\psi(x,y) \simeq 0$ を $1-c_R(x,y) \simeq 0$ とする。ここに，$c_R(x,y)$ は決定可能述語 $R(x,y)$ の特性関数である。

（19）　その他は Cutland (1980) や Odifreddi (1992) 参照。

証明 (a) 無限集合 S は帰納的可算であるとし，ある帰納的関数 h によって $S=Ran(h)$ であるとする。関数 g を $g(n)=h(k(n))$，ただし

$$k(0)=0, k(n+1)=\mu y(h(y)\neq h(k(0)), h(k(1)), \ldots, h(k(n)))$$

と原始帰納的に定義すると，k は帰納的関数となるから g も帰納的で $Ran(g)=Ran(h)=S$。しかも，k の作り方から g は1対1である。逆は定理4. 2. 2。

(b) 無限集合 S は帰納的集合とし，ある帰納的関数 h によって $S=Ran(h)$ とする。関数 g を

$$g(0)=\mu y(y\in S), g(n+1)=\mu y(y\in S \text{ and } y>g(n))$$

と原始帰納的に定義すると，述語‘$y\in S$’は帰納的であるから g も帰納的で $Ran(g)=Ran(h)=S$。しかも，g は単調増加関数である。逆に，単調増加な帰納的関数 g について $S=Ran(g)$ であるとすると，$y=g(n)$ ならば $n\leq y$ だから

$$y\in S \Leftrightarrow y\in Ran(g) \Leftrightarrow \exists n\leq y(g(n)=y).$$

述語‘$g(n)=y$’は帰納的だから，S は帰納的集合である。[20]

(c) 集合 S は帰納的可算無限集合とし，ある帰納的関数 h によって $S=Ran(h)$ とする。関数 g を $g(n)=h(k(n))$，ただし

$$k(0)=0, k(n+1)=\mu y(h(y)>h(k(n)))$$

と原始帰納的に定義すれば，S は無限集合だから，g は単調増加な帰納的関数で，しかも $Ran(g)\subseteq Ran(h)$。よって，(b) より，$Ran(g)$ は S の帰納的部分集合である。□

集合が1対1の帰納的関数で列挙可能であっても，値は離散的であるから，ある数がその関数の**値域に入っていない**ことを確定できるとは限らない。これが一般の帰納的可算集合の場合である。しかし，その関数値を小さいものから順に重複せずに（選択的に）指定していく計算可能な手順が存在し，これによって列挙される数は帰納的な部分集合を生成する。これが (c) の内容である。また，増加関数で列挙可能ならば，与えられた数がその関数値になりうるか否かを確定できることは明らかである。

4. 3 多項式表現

帰納的可算集合や帰納的可算な述語は，整数係数の多項式の整数解の存在と密接な関連をもっている。

---

(20) 述語 $\exists n\leq y(g(n)=y)$ が帰納的であるのは，$(g(0)=y$ or $g(1)=y$ or…or $g(y)=y)$ と同値であり $g(n)=y$ が帰納的だからである。しかし，$\exists n(g(n)=y)$ とすると，帰納的であるとは限らない（定理4. 2. 1）。

定義4．3．1　述語 $R(x_1, ..., x_m)$ に対してある整数係数の多項式※ $P(x_1, ..., x_m; y_1, ..., y_n)$ が存在して，

$$R(x_1, ..., x_m) \Leftrightarrow \exists y_1 ... \exists y_n (P(x_1, ..., x_m; y_1, ..., y_n) = 0)$$

となるとき，述語 $R(x_1, ..., x_m)$ は**ディオファントスである** (diophantine) という。[21]

集合 $S \subseteq N$ に対し，$R(x) \Leftrightarrow x \in S$ としてこの定義が成立つとき，S を**ディオファントス集合** (diophantine set) という。ディオファントス集合 S は帰納的可算集合である。それは，n 個の変数 $y_1, ..., y_n$ を1個の変数 y に符号化して[22]

$$M(x, y) \Leftrightarrow P(x, (y)_1, ..., (y)_n) = 0$$

とおけば述語 $M(x, y)$ はもちろん決定可能で，$x \in S \Leftrightarrow \exists y M(x, y)$。よって定理4．2．1より S は帰納的可算となる。逆も成立つことは1970年になってようやくマチヤセビッチによって証明された。[23]

定理4．3．1　すべての帰納的可算集合はディオファントス集合である。

この定理の威力は，有名な**ヒルベルトの10番問題** (Hilbert's tenth problem) の否定的解決を与えることである。マチヤセビッチはデイビス=パトナム=ロビンソンの研究以来10年近く解決の1歩手前に留まっていたこの問題に，定理4．3．1を証明することによって最終的解決を与えた。この否定的解答は，次のように簡単に述べることができる。指標 x をもつ帰納的可算集合 W_x に対して $R(x) \Leftrightarrow x \in W_x$ とすれば，ある整数係数の多項式 $P(x, y_1, ..., y_n)$ が存在して

$$x \in W_x \Leftrightarrow \exists y_1 ... \exists y_n (P(x, y_1, ..., y_n) = 0)$$

となるが，左辺は決定可能ではない[24]ので右辺も同様。つまり，あるディオファントス方程式については，それが解をもつか否かは決定不能である。それゆえ，すべてのディオファントス方程式が解をもつか否かを決定する単一のアルゴリズムは存在しないことが示されたことになる。[25]

定理4．3．1からえられるもうひとつの重要な帰結は，すべての帰納的可算集合を生成する単一の多項式，**万能多項式** (universal polynomial) の存在である。すなわち，ある多項式 $P(z, x, y_1,$

(21)　整数解 $y_1, ..., y_n$ に限定される方程式 $P(x_1, ..., x_m; y_1, ..., y_n) = 0$ をディオファンスト方程式という。

(22)　$y = p_1^{y_1} p_2^{y_2} \cdots p_n^{y_n}$. ただし，$p_1 = 2$; p_k は k 番目の素数である。

(23)　Matijasevič, J. (1970). Enumerable sets are diophantine. *Dokl. Acad. Nauk* 191, 279-282.

(24)　停止問題の決定不能性。

(25)　ヒルベルトの10番問題の解とは整数解であるが，任意の自然数は整数の2乗の4個の和として表現できるので，自然数の解で決定不能ならば整数解で決定不能である。

$\dots, y_n)$があって，すべての番号 e に対して，

$$x \in W_e \Leftrightarrow \exists y_1, \dots, \exists y_n (P(e, x, y_1, \dots, y_n) = 0)$$

となる。これは，述語 $R(e, x)$ を $R(e, x) \Leftrightarrow x \in W_e$ と定義すると，定義4. 3. 1とマチヤセビッチの定理からしたがう。

万能チューリング・マシンと同様，万能多項式の存在の意味するものはまさに壮大である。単一の方程式が，すべての素数の集合，フィボナッチ数の集合，完全数の集合，ゴールドバッハの仮説をみたす数の集合，フェルマーの式をみたす数の集合など，すべての帰納的可算集合を生成するだけでなく，驚くべきことに，それらを定義するすべてのディオファントス方程式と，変数の数や次数の高さにかかわらず，同等なのである。ジョーンズは67変数76次の万能多項式を実際に作り上げた[(26)]。これは貴重なので，以下に再現しておくことにしよう。

定理4. 3. 2 すべての正の整数 x と n について，$x \in W_n$ であるためには，次の方程式が自然数の解をもつことが必要十分である：

$$
\begin{aligned}
&(2n-[(u+\nu)^2+3\nu+u])^2 \\
&+(\alpha-\theta\beta-\theta)^2+(h+h\beta+h\beta\nu-[x+\rho+\rho\beta+\rho\beta u])^2 \\
&+((h+h\beta+h\beta\nu-\alpha)^2+x^2+\gamma+1-\beta)^2+(3n+\alpha^3-z)^2 \\
&+(z^{18}(z^6+2)(r+1)^2+1-\psi^2)^2+(t+e+e\beta+e\beta s-\alpha-q\varphi)^2 \\
&+(p+b+b\beta+b\beta w-\alpha-q\phi)^2 \\
&+(\{[3(s+w)^2+pw+3s-2r]^2 \\
&\qquad +[(1+\beta+r\beta)^2(1+(\alpha-t-p)^2)(\beta-t^2-p^2)-(g+1)(1+\beta+r\beta)^2]^2\} \\
&\qquad\quad \{[3(\delta+w)^2+9w+3s+2-2r]^2+[(1+\beta+\beta r)^2(1+(\alpha-tp)^2)(\beta-t^2-p^2) \\
&\qquad\qquad -(\alpha-tp)^2-(g+1)(1+\beta+\beta r)^2]^2\}\{3g+2-r\}\{3n+g-r\}-q\pi)^2 \\
&+(r+q+b+e+g+s+w+f'+g'+h'+i'+j'-\omega)^2 \\
&+(\omega^3(\omega+2)(\sigma+1)^2+1-\delta^2)^2+(\sigma r+\sigma-\eta)^2+(b+(q'+\sigma)a'-\eta)^2 \\
&+(e+(r'+\sigma)b'-\eta)^2+(g+(s'+\sigma)\mathrm{c}'-\eta)^2+(s+(t'+\sigma)d'-\eta)^2 \\
&+(w+(u'+\sigma)e'-\eta)^2+(\eta^3(\eta+2)(\zeta+1)^2+1-\epsilon^2)^2 \\
&+(\zeta(\eta-r)(q'+\sigma)(r'+\sigma)(s'+\sigma)(t'+\sigma)(u'+\sigma)-\chi)^2 \\
&+(q+(1+\xi)(\eta-r)-y)^2+(qf'+(q'+\sigma)k'-y)^2+(qg'+(r'+\sigma)\iota'-y)^2 \\
&+(qh'+(s'+\sigma)m'-y)^2+(qi'+(t'+\sigma)n'-y)^2+(qj'+(u'+\sigma)p'-y)^2
\end{aligned}
$$

(26) Jones, J. P. (1978). Three universal representations of recursively enumerable sets. *Jour. Symbolic Logic* 43, 335-351.

$$
\begin{aligned}
&+((\mu^2-1)\chi^2+1-\nu^2)^2\\
&+((\mu^2\chi^2-1)\lambda^2+1-\upsilon^2)^2+(5(c-\chi\lambda y)^2+\iota-\chi^2\lambda^2)^2\\
&+(9\eta\chi y-\mu)^2+(\eta-z+1+k(\mu-1)-\chi)^2+(z+1+\iota(\mu\chi-1)-\lambda)^2\\
&+(\mu\chi+\mu-a)^2+(m+\eta+1-c)^2+((a^2-1)c^2+1-d^2)^2\\
&+(4(a^2-1)\iota^2c^4+1-f^2)^2\\
&+([(a+f^2(f^2-a))^2-1](\eta+1+2jc)^2+1-(d+\tau f)^2)^2\\
&=0
\end{aligned}
$$

5．算術ゲームにおける決定不能性

完全情報2人勝ち負けゲームという古典的なゲームにおける決定不能問題は Rabin (1957) によって最初に考察された。Jones (1982) はさらに，帰納的可算集合を整数係数多項式の整数解と関連づける彼自身の研究にもとづいて，算術ゲームを定義しいくつかの決定不能な問題について述べている。

算術ゲーム (arithmetical games) とは，次に述べるように多項式をペイオフにもつゲームである。変数の数が k の整数係数多項式 $P(x_1, x_2, \ldots, x_k)$ が与えられている。この k をゲームの**長さ** (length) という。まず，プレイヤー1 (J_1) が自然数 x_1 を選び，次にこれを知ってプレイヤー2 (J_2) が自然数 x_2 を選ぶ。次にこれを知って J_1 が x_3 を選び，J_2 がこれを知って x_4 を選ぶ。以下同様。最後に選択するプレイヤー，つまり x_k を選ぶプレイヤーは $P(x_1, x_2, \ldots, x_k)$ をゼロにできれば勝ち，他のプレイヤーはそれをゼロ以外の値にできれば勝つ。

長さ k は偶数であると仮定すると

$$J_1 \text{が必勝戦略をもつ} \Leftrightarrow \exists x_1 \forall x_2 \exists x_3 \cdots \forall x_k (P(x_1, x_2, \ldots, x_k) \neq 0);$$

$$J_2 \text{が必勝戦略をもつ} \Leftrightarrow \forall x_1 \exists x_2 \forall x_3 \cdots \exists x_k (P(x_1, x_2, \ldots, x_k) = 0).$$

もちろん，どちらかのプレイヤーはかならず必勝戦略をもつ。これは $\neg(\forall x_1 \exists x_2 \forall x_3 \cdots \exists x_k (P=0)) \Leftrightarrow \exists x_1(\neg(\exists x_2 \cdots (P=0))) \Leftrightarrow \exists x_1 \forall x_2 (\neg(\forall x_3 \cdots (P=0))) \Leftrightarrow \cdots \Leftrightarrow \exists x_1 \forall x_2 \exists x_3 \cdots \forall x_k \neg(P=0) \Leftrightarrow \exists x_1 \forall x_2 \exists x_3 \cdots \forall x_k (P \neq 0)$ のように純粋に形式的に示すことができる。

必勝戦略の存在は，ツェルメロやフォン・ノイマン[(27)]によって保証されているけれども，どのゲームでどのプレイヤーがそれをもつかということは一般には知られていない[(28)]。算術ゲームについては，

(27) Zermelo (1912), および, Neumann. J. von. (1928). Zur theorie der gesellschaftsspiele. *Mathematische Annalen* 100, 295-320.

(28) ラビンのゲームのような，理論的なゲームは別にして，たとえば，Nash が考案した Hex という2人ゲームではつねに後手が必勝戦略をもつことが知られている。

実はこれは決定不能な問題であるというのが，最初の決定可能性についての結果である。

定理 5．1　任意に与えられた算術ゲームについて，どのプレイヤーが必勝戦略をもつか，という問題は決定不能である。実際，必勝プレイヤーを決定できない長さ 4 の算術ゲームのクラスがある。

この定理の証明には，帰納的可算集合と，必勝戦略の定義のように限定詞が交互に現われる形の述語との関連を示すジョーンズ自身による次の結果（Jones，[1981，Lemma 2]）が必要である：

任意の帰納的可算集合 W に対して整数係数の多項式 $Q(n, x_1, x_2, x_3, x_4)$ が存在して

$$n \in W \Leftrightarrow \exists x_1 \forall x_2 \exists x_3 \forall x_4 (\,Q(n, x_1, x_2, x_3, x_4) \neq 0)$$

となる。

定理 5．1 の証明　$W = K = \{x | x \in W_x\}$ とする。K はすでに述べたように，帰納的ではない帰納的可算集合である。各 n に対し，$Q(n, x_1, x_2, x_3, x_4)$ をペイオフとする算術ゲーム G_n を考えると，すべての n に対して，

$$J_1 \text{ が } G_n \text{ において必勝戦略をもつ} \Leftrightarrow n \in K$$

であるから，左辺は決定不能である。□

定理 5．2　どのプレイヤーも計算可能な必勝戦略をもたない，長さ 6 の算術ゲームがある。

証明　すでに述べたように，ラビンのゲームの本質は次のような完全情報の 2 人勝ち負けゲームである。

- J_1 が先に $x_1 \in N$ を選び，これを知って J_2 が $x_2 \in N$ を選ぶ。
- 与えられた単純集合 $S \subseteq N$ について，$x_1 + x_2 \in S$ ならば J_1 が勝ち，$x_1 + x_2 \notin S$ ならば J_2 が勝つ。

単純集合 S の補集合 S_c は無限集合だから，J_2 が必勝戦略をもつ。しかし，同時に補集合 S^c は免疫性をもつ，すなわち，いかなる無限部分集合も帰納的可算ではない。それゆえ，すべての無限集合 $X \subseteq S^c$ について，決定問題‘$x_1 + x_2 \in X$’は決定不能でしかも半決定可能でさえない。こうして，J_2 の必勝戦略は計算可能ではなく，J_1 はもちろん必勝戦略をもたない。

さて，集合 S は帰納的可算だから，上に述べたジョーンズの結果（Jones，[1981，Lemma 2]）によって，すべての x_1, x_2 に対して，

$$x_1 + x_2 \in S \Leftrightarrow \exists x_3 \forall x_4 \exists x_5 \forall x_6 (Q(x_1 + x_2, x_3, x_4, x_5, x_6) \neq 0)$$

を成立させる整数係数の多項式 Q が存在する。したがって，

$$\forall x_1 \exists x_2 (x_1 + x_2 \notin S) \Leftrightarrow \forall x_1 \exists x_2 \forall x_3 \exists x_4 \forall x_5 \exists x_6 (Q(x_1 + x_2, x_3, x_4, x_5, x_6) = 0).$$

左辺はラビンのゲームにおいてJ_2が必勝戦略をもつこと，右辺はこの多項式Qについての算術ゲームにおいてJ_2が必勝戦略をもつことを記述する。ゆえに，この算術ゲームでは，どのプレイヤーも計算可能な必勝戦略をもたない。□

最後に，メタ数学的な結果を述べよう。Tで算術の記号をもつ，**帰納的に公理化される**（recursively axiomatizable），**サウンド**（sound）な理論[(29)]をあらわす。帰納的に公理化できるとは，公理があてはまる論理式（命題）の個々のケースの集合が帰納的であることをいう。また，サウンドな理論においては，**証明可能**（provable）な論理式，つまり，定理はすべてつねに真である。論理式pが証明可能であるとはpの**証明**（proof）が存在すること，すなわち，最後尾にpをもつ有限個の論理式の列で，各論理式は公理があてはまるケースかまたは列の前に位置するいくつかの論理式から推論規則を用いてえられた結果となっているようなものが存在することである。これは普通におこなう数学の証明を形式化したものであることはいうまでもない。

述語$Pr(x, y)$で「xはyの証明である」という関係をあらわそう[(30)]。列xが論理式yの証明であるか否かは，列の各論理式を推論規則と公理があてはまるケースの帰納的集合に照らして機械的に点検することによって確定できるので，$Pr(x, y)$は決定可能な述語である。それゆえ，証明可能な論理式の集合をTと書くと，

$$y \in T \Leftrightarrow \exists x Pr(x, y)$$

であることに注意すれば，定理4．2．1によって，定理の集合Tは帰納的可算である。

以上の準備のもとに次の定理が証明される。

定理5．3　次の（i），（ii），および（iii）が成立するような，長さ4の算術ゲームGがある：

（i）命題「J_1はGにおいて必勝戦略をもつ，またはJ_2はGにおいて必勝戦略をもつ」は理論Tにおいて証明可能である。

（ii）命題「J_1はGにおいて必勝戦略をもつ」は理論Tにおいて証明可能ではない。

（iii）命題「J_2はGにおいて必勝戦略をもつ」は理論Tにおいて証明可能ではない。

証明　まず，任意の論理式pについて，論理式「pまたは$\neg p$」は理論Tの定理であるから，pを「J_1はGにおいて必勝戦略をもつ」とすれば（i）がしたがう。

次に，再び$W = K = \{x | x \in W_x\}$とすると，$W$は帰納的でない帰納的可算集合である。すると定理5．1と同様にして，ある整数係数の多項式Qが存在し，すべての自然数$n \in N$について，

---

(29)　詳細については，Boolos, G. and R. Jeffrey (1974) など参照。

(30)　すべての論理式が符号化されていると仮定する。

$$n \in W \Leftrightarrow \exists x_1 \forall x_2 \exists x_3 \forall x_4 (Q(n, x_1, x_2, x_3, x_4) \neq 0) \tag{*}$$

となる。集合 W_+ と W_- を次のように定義する。

$W_+ = \{n \in N|$命題「J_1 は G において必勝戦略をもつ」が理論 T で証明可能$\}$

$W_- = \{n \in N|$命題「J_2 は G において必勝戦略をもつ」が理論 T で証明可能$\}$

理論 T はサウンドだから $W_+ \cap W_- = \phi$ であるが，さらに上の (*) に注意すれば

$$W_+ \subseteq W \quad \text{and} \quad W_- \subseteq W^c.$$

W は帰納的ではないので，定理4. 1. 1より W^c は帰納的可算ではない。また，集合 W_- は定理の集合だから，上で述べたように帰納的可算である。ゆえに，

$$W_- \subsetneq W^c.$$

すなわち，$\exists n \in W^c - W_-$. この n について，

$$n \notin W_- \quad \text{and} \quad n \notin W_+.$$

これより（ii)，(iii）がしたがう。□

これは有名なゲーデルの第一不完全性定理の形をした定理である。ゲーデルの定理は，公理化された**無矛盾な**（consistent）算術の理論には，証明もその否定の証明も不可能な論理式が存在することを述べているが，[31] 理論 T の命題「J_1 は G において必勝戦略をもつ」がまさにそのような論理式であることが（ii)と（iii）からわかる。ここで，無矛盾な理論とは，定理の否定は定理にはならない理論のことであり，[32] サウンドな理論はもちろん無矛盾である。

こうして，どのプレイヤーが必勝戦略をもつのかについての合理的推論，すなわち，形式論理にもとづく推論も，手続きとしては完全性を欠いたものとならざるをえないのである。

6. おわりに

手続きの合理性を考察するための形式的基礎理論のひとつとして，計算可能性理論とその応用例である古典的算術ゲームにおける決定可能性について紹介し，存在が保証された戦略や政策も一連の手順＝アルゴリズムとして実行できるとは限らないという結論を，そこに到達するための「一連の手順」とともに理解することを試みた。

決定的な役割を果たしているのは，帰納的可算集合がある算術ゲームの必勝戦略の存在によって定義できる，というJones (1981) の結果である。これはもともと，マチヤセビッチによるディオファントス方程式の決定問題の研究に端を発し，決定可能な述語に限定詞 $\forall$ と $\exists$ を交互に付与し

(31) たとえば，Boolos and Jeffrey (1974) 参照。
(32) 公理が矛盾を含むと，いかなる論理式も定理となる。

てえられる述語の決定可能性に発展した研究からの副産物であるが，ゲーム理論の根幹に関わる問題に貢献することになった。また，単純集合も，計算論において「集合の複雑さ＝決定問題の困難さ」を計る度合いの研究の必要上，ポストによって導入されたものである。[33] 本稿の議論では単純集合の具体的な形は不必要であるが，コルモゴロフによる**ランダム数**（random number）を用いる興味深い自然な具体例がある。[34] ランダム数とゲーム理論との関わりは，無限ゲームのナッシュ均衡の存在が決定不能であるばかりでなく，コインを投げて決定するのと同じ意味でランダムであることを論じた Prasad（1991）の結果が最初であろう。これもジョーンズとマチヤセビッチの仕事，[35] および，アルゴリズム的情報理論の創始者であるチャイティンの仕事[36]に大きく依存する結果である。これについては別の機会にあらためて紹介したい。

本稿で述べた結果は，いずれも限界を画すものという意味では不可能性定理である。しかし，これらを不毛な結果の羅列であるとみるのは誤りであろう。最初にも述べたように，認識，推論および行動というプレイヤー＝経済主体の基本的行為の手続きとしての合理性という視点からすると，これらの結果はこれまでの経済学やゲーム理論で当然とみなされてきた合理性の哲学には，少なくとも反省の余地があることを意味するだけでなく，実行の手順や手続きの重要性をこれまで以上に意識した理論の必要性を示唆するものといえるのである。

（経済学部教授）

参考文献

Anderlini, L. (1990). Some notes on Church's thesis and the theory of games. *Theory and Decision* 29. 19-52.

Binmore, K. (1987). Modelling rational players Ⅰ and Ⅱ. *Economics and Philosophy* 3. 179-214 and 4. 9-55.

Boolos, G. and R. Jeffrey (1974). *Computability and Logic*. Cambridge University Press： Cambridge.

Bridges, D. S. (1994). *Computability*. Springer-Verlag： London.

Canning, D. (1992). Rationality, computability, and Nash equilibrium. *Econometrica* 60. 877-888.

Cutland, N. J. (1980). *Computability*. Cambridge University Press：Cambridge.

Davis, M. (1982). *Computability and Unsolvability*. Dover Publications： New York.

---

(33) Post (1944).

(34) Kolmogorov, A. (1963). On tables of random numbers. *Ind. J. Stat.* 25, 369-376.

(35) Jones, J. P. and Y. V. Matijasevič (1984). Register machine proof of the theorem on exponential diophantine representation of enumerable sets. *Journal of Symbolic Logic* 49, 818-829.

(36) Chaitin, G. (1987). Incompleteness theorems for random reals. *Advances in Applied Mathematics* 8, 119-146.

Jones, J. P. (1981). Classification of quantifier prefixes over diophantine equations. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 27. 403-410.

——— (1982). Some undecidable determined games. *International Journal of Game Theory* 11. 63-70.

Kalai, E. (1990). Bounded rationality and strategic complexity. In T. Ichiishi et al. eds. *Game Theory and Applications.* Academic Press : New York.

Knoblauch, V. (1994). Computable strategies for repeated Prisoners' Dilemma. *Games and Economic Behavior* 7. 381-389.

Marks, R. (1992). Repeated games and finite automata. In J. Creedy et al. eds. *Recent Developments in Game Theory.* Edward Elgar : Vermont.

Nachbar, J. H. and W. R. Zame (1996). Non-computable strategies and discounted repeated games. *Economic Theory* 8. 103-122.

Odifreddi, P. (1992). *Classical Recursion Theory.* North-Holland : Amsterdam.

Post, E. L. (1944). Recursively enumerable sets of positive integers and their decision problems. *Bulletin of the American Mathematical Society.* 50, 284-316.

Prasad, K. (1991). Computability and randomness of Nash equilibrium in infinite games. *Journal of Mathematical Economics* 20. 429-442.

Rabin, M. O. (1957). Effective computability of winning strategies. In M. Dresher et al. eds. *Contributions to the Theory of Games, Annals of Math. Studies.* Princeton University Press : Princeton.

Zermelo, E. (1912). Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. *Proceedings of the Fifth International Congress of Mathematicians* II, 501-504.