# 計算可能性の基礎についての論理的考察

哲学教室　田　畑　博　敏



## はじめに

論理的な証明が機械的かつ実効的なものであることは望ましいことである。よく知られているように，古典命題論理はその定理性を決定する手続きを持つ（論理式$\phi$がｎ個の原子文とｍ個の部分文から成るならば，$m \cdot 2^n$個の調査項目を持つ真理表が描ける）。実効性＝エフェクティヴネス(effectiveness）の概念の必要性は，記号言語の体系という問題場面で論理の研究に登場する。記号列が構文論的に正しく文を形成しているかどうか，をテストするためのアルゴリズムが存在することは，構文論を定式化するための必要不可欠の条件である。こうして，決定可能な構文論によって形式言語の研究をすることが一般に認められるやり方になった(1)。言い換えると，形式的な言語研究において，計算または計算可能性の要素の重要さが認識されるようになった。

アルゴリズムの探究は論理や数学を機械的規則による記述とみなす形式主義によっても動機づけられている。その歴史的な先駆者は，算術の決定方法を求めたヒルベルトである。彼は「命題ｐが算術Ｚで証明可能である」ということがエフェクティヴに決定される方法を求めた。この問題提起はゲーデルの不完全性の結果を生み出すとともに，アルゴリズムの概念を正確に論理的に定式化しようとする1930年代におけるさまざまな提案へと導いた。そして，提案されたアルゴリズムの定式化が実質的に同等であることが判明することによって，アルゴリズムと計算可能性の概念が安定したものであることが信じられるようになった。

小論の目的は，このような言語，論理，数学の基礎に関わるアルゴリズムと計算可能性の基本概念の発展過程を追跡し，その論理的な含意を考察することである。まず，アルゴリズムの基本的事例としてチューリング機械を取り上げ，その発想と基本的メカニズムを考察する（§１）。続いて，計算ステップの最大数を予め予測できる計算可能な関数としての原始帰納的関数を考察し（§２），さらにそれを越えるアルゴリズムの表現としての帰納的部分関数の検討に進む。最後にゲーデルの

定理や各種の決定問題へのこれらの成果の応用を考察し，アルゴリズムの理論としての帰納理論の諸問題を展望する。

## §1．発想と動機

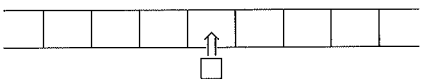
計算の具体的手続きとしてのアルゴリズム（algorithm）には由緒ある歴史がある。例えば，最大公約数を求めるユークリッドの互除法[2]や回文（palindrome）の決定法[3]等がその単純な例である。これらのアルゴリズムには三つの特徴がある：

(1) 手続きが終了するとき，求められている事を当のアルゴリズムが実行していることの証明が与えられる。ユークリッドの互除法は，任意の自然数の対に対して常にそれらの最大公約数を与える。そして，そのことは証明できる。

(2) 手続きは，明確な指令によって，有限的な操作で機械的に実行できる。操作の実行にあたって，単純な記号操作以外に洞察や経験による勘といったものは必要ではない。

(3) 手続きは，有限回のステップを経た後に必ず終了する。

われわれはこれらのアルゴリズムの本質を，A．チューリングが考案した理論上の計算機であるテューリング機械[4]によって実感することから始める。チューリング機械は，有限個の内部状態 $q_1,\cdots, q_n$ を持ち，読み取りと書き込みの装置と，それらを記録し記憶する（可能的に無限に長い）テープを備えた抽象的機械と考えることができる。そのテープは四角のマス目に分割され，機械は一時に1マスだけ左または右方向に移動できる。機械はマス目上に書かれた有限個の記号：$S_1,\cdots, S_n$ を読み取り，かつ印字する，とわれわれは仮定する。機械の動作はきわめて局所的であり，次の形の指令に従って動く：「状態 $q_i$ で記号 $S_j$ を読み取ったとき，記号 $S_k$ を書き込み，状態 $q_h$ となり左または右に移れ」。われわれはこの指令を記号列　$q_i\ S_j\ S_k\ q_h\ X$　で表現する（ここでXはL＝左，またはR＝右である）。適宜，いくつかの規約を定めることによって[5]，機械をスムーズにコントロールできる。さてこれから，このチューリング機械による計算のいくつかの具体例を観察する。



### 1．1　回文判定

まず，与えられた記号列が回文であるかどうかを判定するチューリング機械を考える。機械は，終端記号をチェックしながら左右に走ることになる。両終端記号が同一ならばそれらを消し，次の記号に進む。回文を発見したとき，機械はすべての記号を消すかまたはただ一つの非空白のマス目を残して停止すると規約する。こうして，回文であるか否か，YESかNOかの解答をわれわれは得ることになる。いま，簡単のため，与えられる記号はａ，ｂおよびＢ（「空白」のマス目を表す）とする。テープは最初，・・・ＢＢａａｂａｂａａＢＢ・・・という形をしているとする。機械は最初の内部状態 $q_0$ において左端の記号ａを読み取ることから開始する。これを以下のように図示する：

・・・Ｂａａｂａｂａａ Ｂ・・・
$q_0$

さて，第一の記号がａであることを記憶しながら，機械は右に一マス動いて$q_a$の状態になる（添え字のａが現在の記号列の先頭記号がａであることを記憶することになる)。右方向へさらに移動し続け，最初にＢ（＝空白 Blank）に出会うとき，与えられた記号列を右方向に通り過ぎたことがわかる。そこで左へ一マス戻り，一つ前の記号（つまり当該記号列の最後の記号）がａであるか否かを，$q_a$の記憶を頼りにチェックする。

ここまでに必要な指令は次のものである：

$q_0$ a a $q_a$ R　：状態$q_0$でａを読むとき，そのままで状態$q_a$となり右に１マス移れ；
$q_a$ a a $q_a$ R　：状態$q_a$でａを読むとき，そのままで状態$q_a$となり右に１マス移れ；
$q_a$ b b $q_a$ R　：状態$q_a$でｂを読むとき，そのままで状態$q_a$となり右に１マス移れ；
$q_a$ B B $q_1$ L　：状態$q_a$でＢを読むとき，そのままで状態$q_1$となり左に１マス移れ；
$q_1$ a B $q_2$ L　：状態$q_1$でａを読むとき，ａを消して状態$q_2$となり左に１マス移れ。

これから機械は記号列の先頭に戻ることになる。そこまでのテープの変化を，テープの非空白部分と状態記号による一連の状況記述（state descriptions)として図示する。

Ｂａａｂａｂａａ$q_0$Ｂ→Ｂａａ$q_a$ｂａｂａａＢ→Ｂａａｂａ$q_a$ｂａａＢ→ …

Ｂａａｂａｂａａ$q_a$Ｂ→Ｂａａｂａｂａ$q_1$ａＢ→Ｂａａｂａｂ$q_2$ａＢＢ→ …

Ｂａａｂａｂａ$q_2$Ｂ

機械は再び記号列の先頭に行き，最初の記号を消す。そして次の記号に応じて同様の手続きを繰り返す。そこで，上の指令以外に必要な指令は次のものである：

記号列の先頭に移れ
- $q_2$ a a $q_2$ L
- $q_2$ b b $q_2$ L
- $q_2$ B B $q_3$ R

先頭の記号を消し，次回の記号列の先頭に移れ
- $q_3$ a B $q_0$ R
- $q_3$ b B $q_0$ R

記号ｂを見たら，右に移動して最後の記号をチェックせよ
- $q_0$ b b $q_b$ R
- $q_b$ b b $q_b$ R
- $q_b$ a a $q_b$ R
- $q_b$ B B $q_4$ L
- $q_4$ b B $q_2$ L

機械は計算を続行した後[6]，ＢＢＢとなったところで停止する。なぜなら，“$q_3$ Ｂ”で始まるいかなる指令も無いからである。このとき，テープは空白であるから，与えられた記号列は回文であった。もし回文でなければ，機械は先頭記号がａかｂに応じて状態$q_1$または$q_4$で，記号列の右終端記号を眺めたままで停止し，空白でないテープが残る。

こうして，チューリング機械の「計算」とは一連の状況記述で表現される手続きであり，それは指令の集合によって実行される。従って，チューリング機械そのものを指令の集合と考えることができる。――尚，以後，状況記述の表現において，眺めている記号の直前に状態記号を書く。

### 1. 2　いくつかの算術的演算

次に，以下の節で原始帰納的関数または帰納的関数として分類することになるいくつかの算術的演算を計算するチューリング機械を考える。ここでの目的は，直観的な機械的計算の具体化であるチューリング機械による計算によって，算術に現れる基本的計算がどのように実行されるかを見ることである。ここで取り扱う計算の対象は0を含む自然数である。そこで，通常の規約に従い，n＋1個の縦棒“｜”によって自然数nを表現する。自然数のペアは一個の空白マスで隔てられた，縦棒から成る二つの列によって表現される。またn＋1個の縦棒を$\overline{n}$で表す。さらにアウトプットを読む規約として，「機械が停止したときの縦棒の個数そのものを当のアウトプット，つまり計算結果とする」ということを定める。

**1. 2. 1　同一性関数** (identity function)：$f(x)=x$

上の規約により，縦棒を一個消せばよい。よって，指令は$q_0$｜B$q_0$ Lとなる。計算は，

$$B\,q_0\,|\,|\,|\cdots|\,B\rightarrow q_0\,B\,B\,|\,|\cdots|\,B$$

のようになる。

**1. 2. 2　後者関数** (successor function)：$f(x)=x+1$

これは「何もしない」ということに等しい（上の規約による）。それゆえ，機械は次のダミーの指令を与えるだけでよい：$q_0$ B B $q_0$ R。機械はスタートと同時に停止する。

**1. 2. 3　加　　法** (addition)：$f(x,y)=x+y$

二本の縦棒を消さねばならない。最初の縦棒の列が一本の場合（0＋nの計算）を考慮して，最初の｜を消して状態$q_1$となり，右に進んで最初の｜を消して停止すればよい。よって，指令は，

$$q_0\,|\,B\,q_1\,R$$
$$q_1\,|\,B\,q_2\,R$$
$$q_1\,B\,B\,q_1\,R$$

となる。2＋1＝3の計算例を次に示す：

$$q_0\,|\,|\,|\,B\,|\,|\rightarrow B\,q_1\,|\,|\,B\,|\,|\rightarrow B\,B\,q_2\,|\,B\,|\,|.$$

また，0＋2＝2の計算例は次のようになる：

$$q_0\,|\,B\,|\,|\,|\rightarrow B\,q_1\,B\,|\,|\,|\rightarrow B\,B\,q_1\,|\,|\,|\rightarrow B\,B\,B\,q_2\,|\,|.$$

**1. 2. 4　減　　法** (subtraction)：$f(x,y)=x-y$

$x-y$はここでは$x\geqq y$のときのみ定義されるので，部分関数である。xとyから交互に｜を一本ずつ消し，yがxより先に尽きれば機械は停止する。指令は次のようになる：

$q_0\,|\,|\,q_0\,R$
$q_0\,B\,B\,q_1\,R$
$q_1\,|\,|\,q_1\,R$
$q_1\,B\,B\,q_2\,L$
(＊) $q_2\,|\,a\,q_3\,L$

— 右に動いて，yの最後の｜を消し，マーカー“a”を書け；

$q_3\,|\,|\,q_3\,L$
$q_3\,B\,B\,q_4\,L$
(★) $q_4\,B\,B\,q_4\,L$
$q_4\,|\,B\,q_5\,R$

— xの右端の｜に出会うまで左に動け。出会ったら，それを消して再び右へ動け；

$q_5\ B\ B\ q_5\ R$
$q_5\ |\ |\ q_5\ R$　　マーカーａに出会うまで右に動け。出会ったら，ａの左の｜を消せ。
$q_5\ a\ a\ q_2\ L$　　((＊) による)

例えば，$B\ q_0\ |\ |\ |\ B\ |\ |$ Bから出発するとき，機械はいくつかの計算ステップを経た後[7]，$B\ |\ B\ B\ q_2\ B\ a\ a\ B$で停止する（2－1＝1の計算例)。もしｘがｙより先に尽きてしまうならば，(★) によって，機械は左に移動し続け，決して停止しない。つまり，ｘ＜ｙに対して，アウトプットは無い。

**1. 2. 5　射影関数** (projection function)：$U^n_i\ (x_0,\cdots,x_n)=x_i\ \ (0\leqq i\leqq n)$

機械は最初の｜の列から始めて，$j\neq i$であるｊ＋１番目の列のすべての棒を消し，さらにｉ＋１番目の列からも一本の｜を消す[8]。

ここで，われわれは，テープの非空白の有限部分の左右の終端をそれぞれ表す終端記号（例えば$\$_1$と$\$_2$)を導入できるものとする。そして，$\$_1\ \overline{n}\ \$_2$のような状況記述から出発する計算を一種の予備計算として実行することができるものとする。(引き続き，自然数ｎを表示するｎ＋１本の縦棒｜の列を$\overline{n}$で表す。)

**1. 2. 6　代　　入** (substitution)

チューリング機械$M_1$と$M_2$がそれぞれ関数ｆとｇの計算を実行すると仮定する。そのとき，いかにしてわれわれは，チューリング機械によって，関数：

$$h(x)=f(g(x))$$

を計算させるかを考えなければならない。$M_1$と$M_2$の状態の集合は互いに素だとする。ポイントは，インプットｘに対する$M_2$の計算において，$M_2$が停止するとき$M_2$を終端状態$q_t$へと動かす余分の指令を追加することである。ついで，縦棒を一つにまとめて，機械が$M_1$の初期状態で最左端の｜を眺めるようにさせればよい。よって，われわれは機械$M_1$，$M_2$に次のような演算を実行させる指令を作ればよい：

(1)　$M_2$と同じ計算を実行するが，終端記号の間で行うような機械$M_2'$に変換する。
(2)　さらに$M_2'$を，すべての｜を一まとめにして｜の最左端を眺めて終端状態$q_t$で停止する機械$M_2''$へと変換する。
(3)　$M_1$の内部状態$q_0,\cdots,q_m$を，状態$q_t,\cdots,q_{t+m}$へと数え直し，その結果生じる機械を$M_1'$とする。

こうしてできる$M_2''$と$M_1'$が結合することにより，関数ｈに対して要求される機械が定義される。

**1. 2. 7　原始帰納** (primitive recursion)

新しいアルゴリズムを定義する基本的テクニックの一つが原始帰納の方法である。ここでは，パラメータのない単純なケースを考察する。もしｇが与えられたアルゴリズム（かつ全体関数）ならば，以下の図式で定義される関数ｆも同様のアルゴリズムを与える：

$f(0)=n$　　(ｎは自然数)
$f(x+1)=g(f(x),\ x)$

われわれは，ｎとｘをテープ上に，例えば，$\$_1\ \overline{n}\ \$_2\ \overline{x}\ \$_3$の形で用意する。次に，$\overline{x}$から｜を一本消す。それにより，$\$_2$と$\$_3$の間に何も残らないならば，そのとき，われわれはｎの棒を一本消して停止する。そうでなければ，テープの内容を$\$_1\ \overline{n}\ B\ |\ \$_2\ \overline{x-1}\ \$_3\ \overline{n}\ B\ |$に変換して，$\$_3$の右側に対して$M_1$（ｇを計算する機械）を適用する。そこから生じる縦棒はまとめられて，

$\$_1\overline{n}B\mid\$_2\overline{x-1}\$_3\overline{m}$（ここでm＝f(1)）となる。次に，$\overline{n}$を$\overline{m}$で置き換える。それから，$\$_2$と$\$_3$の間の縦棒をさらにもう一本消す。もしそこに何も残っていないならば$\$_1$と$\$_2$の間の棒を二本消し，かつ$\$_2$の右にある棒をすべて消して，その後に停止する。もしそうでなければ，そのとき$\$_2$の直ぐ左にもう一本棒を追加して，テープ内容を　$\$_1\overline{m}B\mid\mid\$_2\overline{x-2}\$_3\overline{m}B\mid\mid$に変換して，同じ動作を繰り返す。機械は，x個のステップの後に，f(x)個の縦棒をテープ上に残して停止する。余分なパラメータの付加は，テープ上に便宜上，パラメータを蓄えておくという問題にすぎなくなる。

**1. 2. 8　非有界の最小化**（unbounded minimalization）

いま，全体関数g(x, y)を計算するチューリング機械Mが存在すると仮定する。このとき，与えられた任意のyに対して，g(x, y)＝0となるような最小のxを探す機械$M_1$を見出すことができるだろうか？本質的にわれわれは，連続的にg(0, y)，g(1, y)，g(2, y)，…を計算して，アウトプット0（ゼロ）が生み出されるや停止する機械を求めている。よって，次のようにすればよい：

(1)　…B$\$_1\mid B\overline{y}\$_2\$_3$B…という形のテープ状況から出発して，最初の｜を読む；

(2)　$\$_1$と$\$_2$の間の記号列を$\$_2$と$\$_3$の間にコピーする；

(3)　Mを$\$_2$と$\$_3$の間の記号列に働かせる；

(4)　$\$_2$と$\$_3$の間に｜が残っているかどうか（つまりg(x, y)＝0かどうか）をテストする指令を追加し，もし残っていなければ$\overline{y}$を消し，さらに$\$_1$の右の｜を一本消して停止する。もし$\$_2$と$\$_3$の間に｜が残っていれば，$\$_3$から左へ移動しながら$\$_2$と$\$_3$の間の｜をすべて消して，更に左に移動し，$\$_1$の右に｜を一本追加する。

(5)　(2)を繰り返す。

明らかに，もしこの新しい機械$M_1$が停止すれば，テープ内容は望ましいものを生み出している。しかし，機械は，望ましいアウトプットが生み出されるか不明なままで計算し続ける可能性がある。つまり，非有界の最小化の演算は，通常のアルゴリズムの領域を越える。

## 1. 3　万能チューリング機械

チューリング機械（の族）が持つ著しい特徴の一つは，あらゆるチューリング機械を模倣する「親」の機械（いわゆる**万能機械** universal machine）が存在することである。これは，すでにチューリング自身が彼のオリジナルの論文で扱っている主題である[9]。これは大雑把には次のようなことである：もしチューリング機械Mのすべての指令とすべてのインプットが与えられたならば，Mの計算を模倣して同じアウトプットを生み出すチューリング機械が存在する。

このシミュレーションの過程を加法の機械（1.2.3節）の単純なケースで素描する。テープ上に，指令と，適切な記号で分離されたテープ状況が次のように印刷されているとする（0＋1＝1の計算）：

$$\$_1\ q_0\mid B\,q_1\ R*q_1\mid B\,q_2\ R*q_1\ BB\,q_1\ R\,\$_2\ q_0\mid B\mid\mid\$_3$$

加法機械の状況と機械の記号表現およびRとLが新しい機械に対する記号として働いている，ということが重要である。さて，機械をスタートさせ，$\$_2$の右の記号を読ませ，さらに1マス右に動いて“$q_0\mid$”を記憶させる（内部状態として蓄える）。それから，$\$_2$の左のペアの“$q_0\mid$”を求めて左に移動する。ペアが見つかれば，右の3個の記号“B$q_1$R”を再び記憶する。右に動いて，$\$_2$の後に続く“$q_0\mid$”を“B$q_1$”で置き換える。そして，機械は同様の手続きを繰り返す。こうして，機械は元の加法の計算過程を模倣する[10]。

いわゆる**万能チューリング機械**（universal Turing machine）に関する定理が次のように定式化できる：

> 各チューリング機械Mに対して，コード化されたMの指令の列とxとから成るインプットが提示されたとき，インプットxに対するMの計算を模倣できるようなチューリング機械Uが存在する。

この定理の帰結を考えよう。機械Mのコード化された指令の列eをMのインデックスと呼び，インプットxに対するMのアウトプットを$\phi e(x)$で表す。明らかに，万能チューリング機械Uはそれ自身のインデックスを持っている。あるコーディングによって，Uは任意のチューリング機械（のインデックス）に働くことができるが，特に自分自身に働きかけることができる。これはある種の自己言及（self-reference）であり，自己適用（self-application）である。

チューリング機械はアルゴリズミックに動く。すなわち，インプットが与えられたとき，十分に決定されたステップの列をエフェクティヴに実行し，停止するとき一定のアウトプットを生み出す。よって，チューリング機械は決定手続きのモデルとなる。ある問いに対して，YES か NO かの解答を要求する問題は決定問題と呼ばれる。そこで，例えば，YES のときにアウトプット 1 を NO のときアウトプット 0 を産出するようなチューリング機械を設計することで，解答を機械的に引き出す手続きをチューリング機械に実行させることができる。一つの問題に対して，1 または 0 の解答を生み出すチューリング機械が存在するとき，その問題は決定可能な問題と呼ばれる。それでは，決定不可能な問題が存在するか？ある意味では，これはトリヴィアルに YES である。というのは，（以下で考察する）コード化の方法によって，この種の問題は，自然数 n が自然数全体の集合の部分集合Xの要素であるかどうか（$n \in X \subseteq \{0,1,2,\cdots\}$）に掛かっている。ところが，チューリング機械のインデックスは可算無限個しかない，すなわちチューリング機械は可算無限個しかないのに対して，そのようなXは非可算無限個存在するからである。

そこで，問いを限定して，チューリングの停止問題[11]と呼ばれる問題を考える。それは次の問いである：

> インプット（e，x）に対して，もしインデックスeとインプットxを持つチューリング機械が停止すればアウトプット 1 を生み出し，停止しなければアウトプット 0 を生み出すようなチューリング機械が存在するか？

これに対しては，次のようなカントールの対角線論法によって否定的な解答が与えられる。まず，求められた能力をもった，インデックス$e_0$を持つ停止判定機械$M_0$が存在すると仮定する。このとき，$M_0$は以下の条件を満たす関数$\phi e_0$であることになる：

$$\phi e_0(e,x)=\begin{cases}1, & \text{インデックスeの機械Meが計算する関数値}\phi e(x)\text{が存在するとき；}\\ 0, & \text{インデックスeの機械がインプットxに対してアウトプットを持たないとき.}\end{cases}$$

さてわれわれは，この機械$M_0$を少し変形して新しい機械$M_1$を作る。それは，インプット（x，x）に対する$\phi e_0$の値が 0 のとき 1 を取り，1 のとき左に動き続けるという機械である。すなわち$M_1$は次の条件を満たす関数$\phi e_1$である：

$$\phi e_1(x)=\begin{cases}1, & \phi e_0(x,x)=0\text{のとき；}\\ \text{未定義}, & \phi e_0(x,x)=1\text{のとき.}\end{cases}$$

さて，$M_1$ に対して，インプット $e_1$ を適用すると，

$$\phi e_0(e_1, e_1)=0 \quad \Leftrightarrow \quad \phi e_1(e_1)=1 \quad \Leftrightarrow \quad \phi e_0(e_1, e_1)=1$$

となって矛盾である。よって，機械$M_0$は存在しない。つまり停止問題は決定不可能である。

このように，停止問題に典型的に現れるような，機械的に解答できない問題が存在する。しかし，今日のコンピュータの発達が示唆するように，近似的にでも機械的な計算が要求される問題も多数存在する。ところで，そのような機械的計算手続きとしてのアルゴリズムを定式化する標準的な様式があるだろうか。本節の始めで考えたようなアルゴリズムに必要な特徴を表現しうる有限的で離散的な表現手段として自然数$\mathbb{N}=\{0,1,2,\cdots\}$を考えることは，しごく当然であろう。すなわち，アルゴリズムを自然数によってコード化することが，基本的なテクニックと認められる。ゲーデルによるこのアイディアは単純であるが強力なものである。ゲーデルはまず（可能的に可算無限個の）アルファベットに，重複を避けながらただ一つの自然数（ゲーデルに因んでゲーデル数と呼ばれる）を体系的に配分する。次に，アルファベットの記号列に，同じく重複を避けながら，ただ一つの自然数を配分する。つまり，最小素数2から連続的に増大する素数列を取り，当該アルファベットの各ゲーデル数を順次素数の巾指数としてそれらの積をその記号列のコードとする。同様の方法で，記号列の記号列のコード，さらにそれらの記号列のコード，等々が，限りなく，しかも重複なく与えられる。そして，素因数分解の一意性により復コード化のアルゴリズムが存在する。こうして，自然数によるアルゴリズムのコード化（算術化）により，機械的な計算に関わる問題が，ある枠づけられた算術の問題圏に含まれることになる。しかも，この問題圏は，万能チューリング機械に関わる議論で見られたように，数としてコード化されたアルゴリズムそのものが，計算（別のアルゴリズム）の対象となるという重層性を持ちながら，同時に，自然数の部分集合という共通基盤に帰着するという構造を有している。

すると，決定問題の一般形が，自然数の集合Xに対して，$n\in X$？という形になることもより明確に理解される。数とは限らない離散的対象aが性質Aを持つかどうかの問題も，対象とその集合の自然数へのコード化#を考えることにより，$A^{\#}=\{x\in\mathbb{N}\mid A^{\#}(x)\}$（ここで，$A^{\#}(x)$は性質Aを表現する述語）に対して，“$\#(a)\in A^{\#}$？”という問いに帰着するからである。

従って，理論的な目的にとって，アルゴリズム一般の考察は自然数の諸集合に対する決定問題として定式化できることになる。すなわち，自然数の集合Xに対して以下のようなアルゴリズム（Xの特徴関数と呼ばれる）Fが存在するときXは決定可能である：

$$F(n)=\begin{cases}1, & \text{もし } n\in X;\\ 0, & \text{もし } n\notin X.\end{cases}$$

そこで，節を改めて，ある基本的なアルゴリズに対応する，自然数から自然数への関数の構造に関する研究を考察しよう。

## §2．原始帰納的関数

前節で，あらゆるアリゴリズムをシミュレートできるものとして，自然数のアルゴリズムを取ることができることが明らかとなった。そのような自然数のアルゴリズムのクラスとして歴史的かつ方法的に重要であるものが**原始帰納的関数**（primitive recursive function）の族である。原始帰納的関数は，予め承認されたストック関数から出発して，代入（合成）と原始帰納という操作を有限回

適用することによって得られる。すでにわれわれは，代入と原始帰納がチューリング機械からチューリング機械を導くことを見たが，これらの操作によってアルゴリズムを持つ関数から再びアルゴリズムを持つ関数が得られるということを，いくつかの数論的関数において見ることにする。原始帰納的関数は自然数と同じほどに確実で基本的なものと考えられているが，同時に驚くほどの広範囲のアルゴリズムのストックを有している。以下，帰納的定義 (inductive definition)の形で与えられる原始帰納的関数の定義へと進もう。

最初のストックとなる**初期関数** (initial functions) は次の 3 個である。まず，**定数値関数** (constant functions)：$C^k_m(n_1, \cdots, n_k) = m$ (mは定数)；次に，**後者関数** (successor function)：$S(n) = n+1$；そして，**射影関数** (projection functions)：$P^k_i(n_1, \cdots, n_k) = n_i$ $(1 \leqq i \leqq k)$ である。これらのストックから新しい原始帰納的関数を生み出す二つの手続きが認められている。まず**代入** (substitution) または**合成** (composition)である。すなわち，

$$f(n_1, \cdots, n_k) = g(h_1(n_1, \cdots, n_k), \cdots, h_p(n_1, \cdots, n_k))$$

のとき，関数 f が関数 g，$h_1$，…，$h_p$ から代入または合成によって得られると言われる。もう一つの手続きは，**原始帰納** (primitive recursion) である。すなわち，

$$\begin{cases} f(0, n_1, \cdots, n_k) = g(n_1, \cdots, n_k) \\ f(m+1, n_1, \cdots, n_k) = h(f(m, n_1, \cdots, n_k),\ n_1, \cdots, n_k, m) \end{cases}$$

のとき，関数 f が関数 g と h から原始帰納によって得られると言われる。

## 2. 1　定義と例

**原始帰納的関数** (primitive recursive functions) のクラスは，初期関数を含み，代入と原始帰納に関して閉じた最小クラスとして定義される。以下，簡便のため，ゼロを含む自然数の列：$n_1$，$n_2$，…，$n_k$ を $\vec{n}$ と短縮表記する。

原始帰納的関数の例を与える。

①$x+y$　すなわち，

$$\begin{cases} x+0 = x \\ x+(y+1) = (x+y)+1 \end{cases}$$

この定義は，$x+y$ が原始帰納的であることを直接に示す以下の定義の形に書き改めることができる：

$$\begin{cases} +(0, x) = P^1_1(x) \quad \cdots\cdots \text{射影関数} \\ +(y+1, x) = S(P^3_1(+(y, x), x, y)) \quad \cdots\cdots P^3_1 \text{とSの合成.} \end{cases}$$

②$x \cdot y$，③$x^y$，④前者関数 $p(x)$，⑤切断減法 $x \dot{-} y$，⑥階乗 $n!$，⑦記号関数 $sg(x)$，⑧逆記号関数 $\overline{sg}(x)$，⑨絶対値 $|x-y|$，⑩有限和 $\Sigma^y_{i=0} g(\vec{x}, i)$，⑪有限積 $\Pi^y_{i=0} g(\vec{x}, i)$。これらは原始帰納的であることが明示的に理解できるように定義を書き換えることができる[12]。さらに，⑫f が原始帰納的であり，$\pi$ が集合 $\{1, \cdots, n\}$ の置換 (permutation) であれば，$g(x_1, \cdots, x_n) = f(x_{\pi_1}, \cdots, x_{\pi_n})$ である関数 g も原始帰納である。というのは，$g(\vec{x}) = f(P^n_{\pi_1}(\vec{x}), \cdots, P^n_{\pi_n}(\vec{x}))$ だからである (合成)。

次に原始帰納的述語を定義する。その特徴関数が原始帰納的関数である述語が**原始帰納的述語** (primitive recursive predicates) であると定義される。この定義は，それが当てはまる対象である自然数の集合のメンバーシップにより述語をテストするというアイディアに対応している。すなわち，$K_R$ を述語 R (それに対応する集合 R) の特徴関数とすれば，

$$\vec{n} \in R \quad \Leftrightarrow \quad K_R(n_1, \cdots, n_k) = 1$$

であることが分かる。

以下の集合（述語）は原始帰納的である：①空集合$\phi$，②偶数の集合$E=\{0,2,4,\cdots\}$，③同一性関係$=$，④大小関係$<$。これらの特徴関数が原始帰納的であることは容易に確かめ得る[13]。

## 2. 2　いくつかの補題と定理

これから，以後の議論のために原始帰納性に関するいくつかの補題と定理を論じる。

**2. 2. 1　補題**：原始帰納的関係は（集合として捉えた場合）演算$\cap$，$\cup$，$^c$および有界量化の下で閉じている。証明はこれらの演算に対応する特徴関数の原始帰納性による[14]。

ここでは，限定された束縛範囲を持つ量化，すなわち有界量化の場合を考える。$R(n_1,\cdots,n_k,m):=Qx\leqq m\,S(n_1,\cdots,n_k,x)$（ここでQは量化子$\forall$または$\exists$）であるとき，関係Rは関係Sから**有界量化**（bounded quantification）によって得られる，と言われる。いま，有界存在量化：$R(\vec{x},n):=\exists y\leqq n\ S(\vec{x},y)$を考える。そのとき，Rの特徴関数を取ると，$K_R(\vec{x},n)=\mathrm{sg}\,\Sigma y\leqq n\ K_S(\vec{x},y)$であるから，Sが原始帰納的ならば，Rも原始帰納的である。また有界普遍量化：$R(\vec{x},n):=\forall y\leqq n\ S(\vec{x},y)$を考える。そのとき，$K_R(\vec{x},n)=\mathrm{sg}\,\Pi y\leqq n\ K_S(\vec{x},y)$となるから，Sが原始帰納的ならばRも原始帰納的である（Q. E. D）。

**2. 2. 2　補題**：原始帰納的関係は原始帰納的代入の下で閉じている。すなわち，もし$f_1,\cdots,f_p$およびRが原始帰納的ならば，$S(x_1,\cdots,x_n):=R(f_1(\vec{x}),\cdots,f_p(\vec{x}))$で定義されるSも原始帰納的である[15]。

**2. 2. 3　補題（場合分けによる定義）**

$R_1,\cdots,R_p$が相互に排反的（mutually exclusive）な原始帰納的述語であり，かつ$\forall\vec{x}\,(R_1(\vec{x})\vee\cdots\vee R_p(\vec{x}))$であり，しかも$g_1,\cdots,g_p$が原始帰納的関数であるとすると，そのとき，

$$f(\vec{x})=\begin{cases} g_1(\vec{x}), & \text{もし}\quad R_1(\vec{x})\ \text{のとき；}\\ \quad\vdots & \qquad\vdots\\ g_p(\vec{x}), & \text{もし}\quad R_p(\vec{x})\ \text{のとき，}\end{cases}$$

として定義される関数$f(\vec{x})$は原始帰納的である[16]。

自然数の持つ著しい性質の一つに，非空な部分集合がすべて最小要素を持つ（$\mathbb{N}$の整列可能性）ということがある。すると，この最小要素をエフェクティヴに見出す方法があるか，と問うことは自然であろう。一般的には，この答えは否定的であるが，考察している集合が非空でかつ原始帰納的であれば，非空を保証する要素を取り出すことができ，それより小さい数が要素か否かを確かめることができる。そこで，新しい記号法：$\mu y\,R(\vec{x},y)$を導入する。これは，もし存在すれば，$R(\vec{x},y)$が成り立つような最小数yを表す。また，$\mu y<m\ R(\vec{x},y)$は，もし存在すれば$R(\vec{x},y)$が成り立つような最小数y（$<$m）を表し，もし存在しないならば，単にmを表す。前者は**μ作用素**，後者は**有界μ作用素**（bounded $\mu$-operator）と呼ばれる。

**2. 2. 4　補題**：もしRが原始帰納的であれば，$\mu y<m\ R(\vec{x},y)$も原始帰納的である[17]。

こうして，多数の関数・関係の原始帰納性を確立できる準備が整ってきた。

**2. 2. 5　例**：以下のものは原始帰納的である。①素数の集合。というのは，xは素数（prime）である$\Leftrightarrow\forall y\leqq x\ \forall z\leqq x\ (x=y\cdot z\supset y=1\vee z=1)\wedge x\neq 1$だからである。②割り切れるという関係：$x\mid y\Leftrightarrow\exists z\leqq y\ (x\cdot z=y)$。③xの素因数分解中での素数pの巾指数（exponent）：$f(x)=\mu y\leqq x\ (p^y\mid x\wedge\neg p^{y+1}\mid x)$。④“n番目の素数”関数$p_n$：$p_1=2$，$p_{n+1}=\mu x\leqq(\Pi_{i=1}^{n}p_i)+1\ [x\text{は素数である}\wedge x>p_n]$。

われわれは，これまでの原始帰納的関数のストックを，自然数の有限列を自然数にコード化する

ために用いることができる：

$$(n_1, \cdots, n_k) \longmapsto 2^{n_1+1} \cdot 3^{n_1+1} \cdot \cdots \cdot p_i^{ni+1} \cdot \cdots \cdot p_k^{n_k+1}。$$

述語：Seq(n) を「nは**系列数** (sequent number)である」とすると，この述語は明らかに原始帰納的である。というのは，自然数の有限列の上述のコード化により，この述語は「一つの素数がnを割り切るならば，それより小さいすべての素数がnを割り切る」と同等となるからである：

$$\forall p \leqq n \quad \forall q \leqq n \ [Prim(p) \wedge Prim(q) \wedge q < p \wedge p \mid n \supset q \mid n] \wedge n \neq 1$$

("Prim(x)" ：xは素数である)。もしnが列 $\langle a_1, \cdots, a_k \rangle$ の系列数であるならば，その長さ lth(n) (つまりk) を見出すことができる：

$$lth(n) := [\mu x \leqq n+1 \ (\neg p_x \mid n)] - 1。$$

特に lth(1) = 0である。また，われわれは系列数nを**解読する** (decode) ことができる：

(n)i := (系列数nの素因数分解におけるi番目の素数の巾指数) − 1。(上の例③参照)

(n)iは系列数nの系列でのi番目の記号のコードを意味する。lth(n) も (n)iも原始帰納的である。というのは，$(a_1, \cdots, a_k) \longmapsto \Pi_{i=1}^{k} p_i^{ai+1}$ は原始帰納的だからである。ここで，$\langle a_1, \cdots, a_k \rangle := \Pi_{i=1}^{k} p_i^{ai+1}$ とする (つまり列表現をそのままその列の系列数表現に流用する)。

われわれは二つの系列数の**「連結」** (concatenation) をコード化できる。nを $\langle a_1, \cdots, a_k \rangle$ のコード，mを＝ $\langle b_1, \cdots, b_p \rangle$ のコードとするとき，二列の連結 $\langle a_1, \cdots, a_k, b_1, \cdots, b_p \rangle$ のコード：n＊mは次のように定義できる：$n * m := n \cdot \Pi_{i=1}^{lth(m)} p_{lth(n)+i}^{(m)_i+1}$

### 2. 3　累積帰納

もう一つの帰納の形式として，値が，先行するすべての値に依存するという累積帰納を考える。それの正確な定義のために，関数 f (y, $\vec{x}$) に対して，これの**累積値関数** (course of value function) $\overline{f}(y, \vec{x})$ を次のように定義する：

$$\begin{cases} \overline{f}(0, \vec{x}) = 1 \\ \overline{f}(y+1, \vec{x}) = \overline{f}(y, \vec{x}) \cdot p_{y+1}^{f(y, \vec{x})+1} \end{cases}$$

例えば，f (0) = 1，f (1) = 0，f (2) = 7ならば，$\overline{f}(0) = 1$，$\overline{f}(1) = 2^{1+1}$，$\overline{f}(2) = 2^2 \cdot 3^{0+1}$，$\overline{f}(3) = 2^2 \cdot 3^1 \cdot 5^8$である。明らかに，もしfが原始帰納的であれば，$\overline{f}$も原始帰納的である。$\overline{f}$ (n+1) は第n番目までのfに関する言わばすべての情報をコード化するので，$\overline{f}$を，累積帰納法を定式化するために用いることができる。

#### 2. 3. 1　定　　理

もしgが原始帰納的であり，f (y, $\vec{x}$) = g ($\overline{f}$ (y, $\vec{x}$), y, $\vec{x}$) であるならば，fも原始帰納的である。

《証明》最初に$\overline{f}$を改めて定義し直す。

$$\begin{cases} \overline{f}(0, \vec{x}) = 1 \\ \overline{f}(y+1, \vec{x}) = \overline{f}(y, \vec{x}) * \langle g(\overline{f}(y, \vec{x}), y, \vec{x}) \rangle \end{cases}$$

これは原始帰納の図式により，$\overline{f}$が原始帰納的であることを示している。ところが，仮定より，

必要なf(y, $\vec{x}$) という情報は$\overline{f}$(y+1, $\vec{x}$) の列の最後尾にあることになる。すなわち, f(y, $\vec{x}$) = $(\overline{f}\ (y+1, \vec{x}))_{y+1}$。ゆえにfは原始帰納的である (Q. E. D.)。

これまで, さまざまな原始帰納的関数・述語を考えたが, それ以上のアルゴリズムがあるのかどうか, ということは問うに値する問いであろう。答えはＹＥＳであり, 以下の議論がそれを示す。各原始帰納的関数fは, 各関数fiが初期関数かまたは先行する関数に代入または原始帰納の手続きを適用して得られる関数列：$f_1, f_2, \cdots, f_n (=f)$ によって定義される。このような定義の全体を, すべての情報がエフェクティヴに引き出せるような形で自然数へとコード化することは厄介であるがルーティンにすぎない。そこで, $f_x$ がコードxを持つ原始帰納的関数であるとき, F(x, y) $=f_x(y)$であるような関数Fを定義できることが示される。いま, D(x) $=$F(x, x) $+1$を考える。Dが原始帰納的であると仮定すると, あるnに対して, $D=f_n$。そのとき, $D(n)=F(n,n)+1=f_n(n)+1=f_n(n)$。矛盾。こうして, 対角線論法により, 原始帰納的関数全体の外に出るが, しかし, Dに対してアルゴリズムの性格は未だ残した。よって, より広いアルゴリズムのクラスを考えねばならない。節を改めてそれに向かう。

## § 3.　帰納的部分関数

これから, われわれはアルゴリズムのクラスを拡張する。これまでと異なり, そのクラスは部分関数を含むことになる。つまり, われわれが導入するアルゴリズムは**帰納的部分関数** (partial recursive functions) である。われわれは万能チューリング機械に対応する, ある種の万能関数を考える。そのために, 各アルゴリズムにインデックスと呼ばれるコード数を与える。そして,「インデックスeを持つアルゴリズムはインプット ($x_1, \cdots, x_n$) に対してアウトプットyを生み出す」を記号で,

$$\{e\}(x_1, \cdots, x_n) \simeq y$$

と表現する[18]。

**3. 1　定義**：関係 $\{e\}(\vec{x}) \simeq y$ は, 以下のように帰納的に (inductively) 定義される。

**R 1**　$\{\langle 0, n, q\rangle\}(m_1, \cdots, m_n) \simeq q$

**R 2**　$\{\langle 1, n, i\rangle\}(m_1, \cdots, m_n) \simeq m_i$　($1 \leqq i \leqq n$に対して)

**R 3**　$\{\langle 2, n, i\rangle\}(m_1, \cdots, m_n) \simeq m_i + 1$　($1 \leqq i \leqq n$に対して)

**R 4**　$\{\langle 3, n+4\rangle\}(p, q, r, s, m_1, \cdots, m_n) \simeq p$　(もしr＝sならば)

　　$\{\langle 3, n+4\rangle\}(p, q, r, s, m_1, \cdots, m_n) \simeq q$　(もしr≠sならば)

**R 5**　$\{\langle 4, n, b, c_1, \cdots, c_k\rangle\}(m_1, \cdots, m_n) \simeq p$

　　(もし, $\{c_i\}(m_1, \cdots, m_n) \simeq q_i (1 \leqq i \leqq k)$ かつ $\{b\}(q_1, \cdots, q_k) \simeq p$であるような$q_1, \cdots, q_k$が存在すれば)

**R 6**　$\{\langle 5, n+2\rangle\}(p, q, m_1, \cdots, m_n) = S^1_n(p, q)$

**R 7**　$\{\langle 6, n+1\rangle\}(b, m_1, \cdots, m_n) \simeq p$　(もし $\{b\}(m_1, \cdots, m_n) \simeq p$ならば)

以上の図式については, $\{e\}(\vec{x})$ の読み方に従って, 次のようにパラフレーズできる：

Ｒ１：インデックス $\langle 0, n, q\rangle$ を持つ機械はインプット ($m_1, \cdots, m_n$) に対して, アウトプットqを生み出す (**定数値関数**)。

Ｒ２：インデックス $\langle 1, n, i\rangle$ を持つ機械はインプット$\vec{m}$に対してアウトプット$m_i$を生み出す

(**射影関数** $P^{n}_{i}$)。

R 3：インデックス〈2, n, i〉を持つ機械はインプット $\vec{m}$ に対してアウトプット $m_i+1$ を生み出す（i 番目の引数(アーギュメント) $m_i$ の**後者関数**)。

R 4：インデックス〈3, n+4〉を持つ機械は，インプットの第 3 番目と第 4 番目の引数が同じか違うかに応じて，第 1 または第 2 番目の引数をアウトプットとして生み出す（**識別関数**)。

R 5：インデックス〈4, n, b, $c_1, \cdots, c_k$〉を持つ機械は，最初に，インデックス $c_1, \cdots, c_k$ を持つ機械をインプット $\vec{m}$ に関してシミュレートして，それから，アウトプットの列（$q_1, \cdots, q_k$）をインプットとして利用して，インデックス b を持つ機械をシミュレートする（**代入**)。

R 6：後の $S^{m}_{n}$ 定理で特定する。

R 7：インデックス〈6, n+1〉を持つ機械は，与えられたインプット b, $m_1, \cdots, m_n$ に対して，インデックス b とインプット $\vec{m}$ を持つ機械をシミュレートする（**反射**)。つまり，この機械は n 項インプットのすべての機械に対して万能機械として働く。

パラフレーズされた関数としてのこれらの機械のインデックスは，関連する情報のすべてを含んでいる。すなわち，第 1 座標はどの条項を用いるべきかを告げ，第 2 座標は引数の個数を与える。残りの座標は各機械に特有な情報を与える。R 7 はきわめて強力であり，一定数の引数を持つすべての機械を枚挙する。これは対角化のために必要な機械である。それは，各機械のインデックスがエフェクティヴに認知できるならば，それを実行し，認知した各機械の動きをシミュレートする。

上の定義は，われわれが何を計算と見なすべきかを示唆している。すなわち，$\{e\}(\vec{x})$ を計算するために，われわれはまず e を見る。e の第一記載項目が 0 か 1 か 2 ならば，対応する初期関数によってアウトプットを計算する。もし第一記載項目が 3 ならば，r = s か否かにより計算する。件の項目が 4 ならばインデックス $c_1, \cdots, c_k$ に関わる予備計算を実行した後，インデックス b による部分計算を行って，R 5 により最終アウトプットを見出す。件の項目が 5 ならば $S^{m}_{n}$ 定理によって示される仕方で計算する。件の項目が 6 のとき，インデックス b による部分計算にジャンプする。

R 7 を認めることによって，計算過程の停止はもはや一般的には保証されていない。なぜなら，以下のような単純なループに巻き込まれる可能性があるからである。R 7 により，$\{e\}(x) = \{x\}(x)$ であるような e が存在する。引数 e に対する計算を行うため，R 7 に従って，右辺に進む。すなわち，$\{e\}(e)$ を計算せねばならない。しかし，停止問題の事例が教えるように，これに対するアウトプットは必ずしも保証されない。ループや非停止の計算はインプットに対して未定義であることを意味する。つまり，関数として見たとき部分関数となる。以後，帰納的部分関数を $\phi$, $\psi$, …で，全体関数は f, g, h, …で表すことにする。また，“$\simeq$”を表すにも，通常の同一性を表すにも，区別せずに“=”を用いる。

以下の重要な定理は，巧妙な機械に関する動機に裏打ちされている。二つの引数 x, y に対して働く，インデックス e の機械を考える。x を固定したとき，われわれは y に対して働く機械を得るとする。そうした機械のインデックスは x に依存するか？答えは YES である。次の定理がこのことを保証する。

### 3. 2　$S^m_n$ 定理

帰納的部分関数のインデックスeが与えられたとき，$0<m<n$であるすべてのm，nに対して

$$\{S^m_n(e, x_1, \cdots, x_m)\}(x_{m+1}, \cdots, x_n) = \{e\}(\vec{x})$$

であるような原始帰納的関数$S^m_n$が存在する。

《証明》$S^m_n$は$S^1_n$（R6）をm回適用することによって得られる[19]。

$S^m_n$定理は，帰納的部分関数の一様性（uniformity）の特性を表現している。例えば，帰納的部分関数$\phi(x, y)$に対して，xに定項nを代入して得られる関数$\phi(n, y)$が部分帰納的であることは明らかであるが，このことは，$\lambda y\, \phi(x, y)$が体系的かつ一様な仕方で計算可能であることを示してはいない。

$S^m_n$定理の一つの応用を考える。$\phi(x)$を$\phi(x) = \{e\}(x) + \{f\}(x)$と定義する。そのとき，$\phi$は部分帰納的である。そこで，$\phi$のインデックスをeとfの関数として表現したい。いま，$\psi(e, f, x) = \{e\}(x) + \{f\}(x)$を考える。$\psi$は部分帰納的である。それゆえ，$\psi$はインデックスnを持っている。すなわち，$\{n\}(e, f, x) = \{e\}(x) + \{f\}(x)$である。$S^m_n$定理により，

$$\{n\}(e, f, x) = \{h(n, e, f)\}(x)$$

であるような原始帰納的関数hが存在する。従って，$g(e, f) = h(n, e, f)$は要求された関数である。

### 3. 3　帰納定理（recursion theorem）

$$\{rc(e)\}(\vec{x}) = \{e\}(rc(e), \vec{x})$$

であるような，原始帰納的関数rcが存在する。

この定理を証明する前に，この定理の意義を考える。これは，$\phi(\vec{x}) = \{e\}(\cdots\phi\cdots, \vec{x})$であるような帰納的部分関数$\phi$が欲しいという問題を解決する。帰納的部分関数を求めるということは，その関数のインデックスを求めることに等しい。よって，$\phi$を$\{z\}$で置き換える（ここで，zは未だ知られていないインデックスである）。すると，$\{z\}(\vec{x}) = \{e\}(\cdots\{z\}\cdots, \vec{x}) = \{e'\}(z, \vec{x})$。rcが原始帰納的であるから，$rc(e')$が$\phi$に対して求められたインデックスを与える。

《証明》[20] $\phi(m, e, \vec{x}) = \{e\}(S^2_{n+2}(m, m, e), \vec{x})$とし（※），pを$\phi$のインデックスとする。そして，$rc(e) = S^2_{n+2}(p, p, e)$とおく（★）。そのとき，

$\{rc(e)\}(\vec{x}) = \{S^2_{n+2}(p, p, e)\}(\vec{x}) = \{p\}(p, e, \vec{x})$　……$S^m_n$定理より

$= \phi(p, e, \vec{x})$　……pが$\phi$のインデックス

$= \{e\}(S^2_{n+2}(p, p, e), \vec{x})$　……（※）より

$= \{e\}(rc(e), \vec{x})$　(Q. E. D.)　……（★）より

《例》**アッカーマン関数**（the Ackermann function）：以下の関数列を考える：

$$\phi_0(m, n) = n + m$$
$$\phi_1(m, n) = n \cdot m$$
$$\phi_2(m, n) = n^m$$
$$\vdots$$
$$\begin{cases} \phi_{k+1}(0, n) = n \\ \phi_{k+1}(m+1, n) = \phi_k(\phi_{k+1}(m, n), n) \end{cases} \quad (2 \leqq k)。$$

この列は，次第に一層早く増大する関数の列から成っている。われわれは，それらすべての関数を次の関数にまとめることができる：

$$\phi(k, m, n) = \phi_k(m, n)。$$

上の方程式の列は，次のように要約される：

$$\begin{cases} \phi(0, m, n) = n+m \\ \phi(k+1, 0, n) = \begin{cases} 0, & \text{もし } k=0 \text{ のとき；} \\ 1, & \text{もし } k=1 \text{ のとき；} \\ n, & \text{それ以外のとき（すなわち } 1<k \text{ のとき）；} \end{cases} \\ \phi(k+1, m+1, n) = \phi(k, \phi(k+1, m, n), n)。 \end{cases}$$

これが（一般化された）アッカーマン関数である。第二の方程式は，$\phi_{k+1}$ が乗法，巾，一般のケース（$k \geqq 2$）に応じて場合を分けねばならないことを示している。すべての原始帰納的関数は帰納的であるという事実（以下の定理３．６の系）を用いて，われわれは三つのケースを，ある適切な関数 f に対して，$\{e\}(k, m, n) = f(e, k, m, n)$ という形の一つの方程式に書き直すことができる。よって，帰納定理により，上の方程式を満たすインデックス e を持つ帰納的関数 f が存在する。アッカーマンは，関数 $\phi(n, n, n)$ がどんな原始帰納的関数よりも早く増大することを示した。

**3. 4　標準形定理**（normal form theorem）

$$\{e\}(\vec{x}) = (\mu z\, T(e, \langle \vec{x} \rangle, z))_1$$

であるような，原始帰納的述語 T が存在する。

《証明》略[21]。

この述語 T は，「z は，インデックス e を持つ帰納的部分関数（機械）の，インプット $\langle \vec{x} \rangle$ に対する計算である」という言明を形式化したものである。ただし，ここでの「計算」では，その第一の射影（projection）がアウトプットであるように定義されているものとする（それが右辺の添え字“1”の意味である）。応用の場面では，述語 T の正確な構造は重要ではない。$S^m_n$ 定理，帰納定理および標準形定理により，非決定性に関する多くの結果が得られる。

**3. 5　定理**：f を帰納的関数とする。そのとき，関数：

$$\phi(\vec{x}) = \mu y\, [f(y, \vec{x}) = 0]$$

は部分帰納的である。

《証明》帰納定理を利用して行う[22]。

**3. 6　定理**：帰納的関数は原始帰納の下で閉じている。

《証明》いま g と h が帰納的関数であるとする。示すべきことは，

$$\begin{cases} f(0, \vec{x}) = g(\vec{x}) \\ f(y+1, \vec{x}) = h(f(y, \vec{x}), \vec{x}, y) \end{cases}$$

として定義された関数 f も帰納的である，ということである。上の図式を，以下のような場合分けの定義に書き改める：

$$f(y, \vec{x}) = \begin{cases} g(\vec{x}), & y = 0 \text{ のとき；} \\ h(f(y-1, \vec{x}), \vec{x}, y-1), & y > 0 \text{ のとき。} \end{cases}$$

先行する関数値が帰納的に計算できるから，関数fのインデックスeを用いた，

$$\{e\}(y, \vec{x}) = \{a\}(y, \vec{x}, e)$$

という方程式が成り立つ（ここで，aはg，hとfの先行関数値から計算される）。帰納定理により，方程式は解$e_0$を持つ。y上の帰納法から$\{e_0\}$が全体的であることにより，fは帰納的であると結論される。　　（Q．E．D.）

**系**：すべての原始帰納的関数は帰納的である。

### 3．7　定　　義

ここで，いくつかの定義を与える：

(i)集合（または関係）が（帰納的）に決定可能(decidable）であるのは，それが帰納的である場合である。

(ii)集合は，もしそれが帰納的部分関数の定義域（domain）となっているならば，帰納的に枚挙可能である（recursively enumerable)と言われる。この述語を“ＲＥ”と略記する。

(iii)$W_e^k = \{\vec{x} \in \mathbb{N}^k \mid \exists_y (\{e\}(\vec{x}) = y)\}$とする。すなわち，$W_e^k$は帰納的部分関数$\{e\}$の定義域とする。eを$W_e^k$のＲＥインデックスと呼ぶ。

抽象的機械に受け入れられる集合として帰納的に枚挙可能な集合（＝ＲＥ集合）を考えることができる。0，1，2，…と自然数を提示するとき，機械は，それらをインプットとして受け入れ可能な場合にアウトプットを生み出す。このことは，ある良い発見法を構成する。もし集合Ａｉが機械Ｍｉ（$i=0,1$）によって受け入れられているならば，平行して働くことにより$M_0$と$M_1$を模倣する機械Ｍを作ることができる。よって，もし自然数ｎが$M_0$または$M_1$に受け入れられているならば，ｎはＭによっても受け入れられる。こうして，ＲＥ集合の合併集合もＲＥである。

ＲＥ集合の例として次のものが基本的である：

(i)$\mathbb{N}$＝定数値関数の定義域

(ii)$\phi$＝空な関数の定義域

(iii)すべての帰納的集合はＲＥである。実際，Ａを帰納的集合とする。$\vec{x} \in A$のとき，$\psi(\vec{x}) = 0$，$\vec{x} \notin A$のとき，$\psi(\vec{x}) = \varnothing(\vec{x})$（ここで関数$\varnothing$は空）と定義すると，$\mathrm{Dom}\,\psi = A$（“Dom”は「定義域」と読む)。

帰納的に枚挙可能な集合は，それらの要素が帰納的部分関数によって生み出される，つまりアルゴリズムによって表現できる，という点で重要である。事実，論理学における重要な関係（集合）がＲＥ（帰納的に枚挙可能）である。例えば，算術や述語論理における証明可能な文の集合はＲＥである。ＲＥは決定可能な集合を越える第一のステップである。

### 3．8　ＲＥ集合に関連したいくつかの定理

**定理**：以下の言明は同値である。

(i)ある帰納的部分関数$\phi$に対して，$A = \mathrm{Dom}(\phi)$。

(ii)ある帰納的部分関数$\phi$に対して，$A = \mathrm{Ran}(\phi)$。(“Ran”は「値域」)

(iii)ある帰納的部分関係Ｒに対して，$A = \{x \mid \exists y\, R(x, y)\}$。

**定理：**

(i)集合A，BがREならば，集合A∪B，A∩BもREである。

(ii)関係R(y，$\vec{x}$) がREならば，∃yR(y，$\vec{x}$) もREである。

(iii)R(y，$\vec{x}$) がREであり，$\phi$が部分帰納的であれば，R($\phi$ ($\vec{x}$，$z_1$,…,$z_n$)，$\vec{x}$) もREである。

(iv)R(y，$\vec{x}$) がREならば，∀y＜zR(y，$\vec{x}$) も∃y＜zR(y，$\vec{x}$) もREである。

**定理：**集合Aが帰納的である⇔AもA$^c$ もREである。

この最後の定理は，RE集合によって帰納的集合を特徴づけている。AとA$^c$がともにREとすると，これらの集合を枚挙する二つの機械がある。両方の機械を同時に動かし，各自然数nについて，二つの機械のアウトプットとしてnが生み出されるかどうかを見ればよい。これは有限個のステップの後に判明する。というのは，排中律により，n∈Aまたはn∈A$^c$だからである。よって，Aのメンバー性をテストするエフェクティヴなテストの方法が存在することになる。

### 3. 9　決定不可能な問題の例

(1)　**停止問題**

K＝{x | ∃zT(x，x，z)} を考える（ここで，Tは3.4節の標準形定理で登場した原始帰納的述語である)。Kは帰納的関係の射影である。よって，KはREである。そのとき，K$^c$もREであると仮定する。そのとき，あるインデックスeに対して，x∈K$^c$⇔∃zT(e，x，z)。さて，e∈K⇔∃zT(e，e，z) ⇔e∈K$^c$。これは矛盾である。よって，K$^c$はREでない。よって，上の定理より，Kは帰納的でない。

Kに対する決定問題は停止問題と呼ばれる。なぜなら，この問題は，「インプットとしてxが提示されたとき，有限個のステップの後に停止するような計算を実行する，インデックスxを持つ機械を決定せよ」としてパラフレーズできるからである。こうして，「インデックスxを持つ機械がインプットyに対して停止するかどうか」は決定不可能である。

(2)　インデックスxを持った機械（＝関数）{x} が全体関数であるかどうかは決定可能でない。それが決定可能だと仮定する。そのとき，f(x)＝0⇔{x} が全体的である，であるような帰納的関数fを持つことになる。さて，このとき，

$$\phi(x,\ y) := \begin{cases} 0, & \text{もしx∈Kのとき；} \\ \text{未定義}, & \text{もしx∉Kのとき，} \end{cases}$$

を考える。$S^m_n$定理により，{h(x)} (y)＝$\phi$(x，y) であるような帰納的関数hが存在する。ところで，{h(x)} が全体的であるのはx∈Kであるときかつそのときにかぎるから，f(h(x))＝0⇔x∈K。すなわち，Kに対する帰納的特徴関数sg(f(h(x))) がある。上の(1)の結果より，これは矛盾である。よって，そのようなfは存在しない。つまり，集合{x | {x} は全体的である} は帰納的ではない。

(3)　「Weは有限であるか」という問題は帰納的には可解でない。f(e)＝0⇔Weは有限である，ような帰納的関数fが存在すると仮定せよ。h(x) を上の(2)で定義されたものとする。明らかに，$W_{h(x)}$＝Dom {h(x)} ＝∅⇔x∉K，そして，x∈Kなるxに対して$W_{h(x)}$は無限である。f(h(x))＝0⇔x∉K，それゆえ，sg(f(h(x))) はKに対する帰納的特徴関数である。矛盾。

(4) ＲＥ集合間の同一性は決定不可能である。すなわち，$\{(x, y) \mid W_x = W_y\}$ は帰納的でない。この問題は，$W_y = \varnothing$ を選ぶことによって，(3)の解へと還元される。

(5) We が帰納的かどうかは決定できない。$\phi(x, y) = \{x\}(x) \cdot \{y\}(y)$ とおく。そのとき，一定の帰納的 h に対して $\phi(x, y) = \{h(x)\}(y)$，および

$$\mathrm{Dom}\{h(x)\} = \begin{cases} K, & x \in K \text{のとき；} \\ \varnothing (= \{\ \}), & x \notin K \text{のとき。} \end{cases}$$

$f(x) = 0 \Leftrightarrow W_x$ は帰納的である，が成り立つような帰納的関数 f が存在すると仮定せよ。そのとき，$f(h(x)) = 0 \Leftrightarrow x \notin K$。これより，Kは帰納的であることになり，矛盾。

### 3. 10　チャーチの提唱

帰納的アルゴリズム以外のアルゴリズムがあるか？この問題は解かれていない。しかし，原始帰納的関数についての同じ問いは肯定的に答えられる（すなわち，原始帰納的関数として表現しうるアルゴリズム以外のアルゴリズムが存在する）。対角化によって，エフェクティヴな仕方で原始帰納的関数の外に出ることができるからである。しかし，その手続きは帰納的関数には当てはまらない。

アルゴリズミックな初期関数から出発して，代入と射影の下での閉包がアルゴリズムからアルゴリズムへと導くことを承認するならば，すべての帰納的関数がアルゴリズミックであることの帰納的（インダクティブ）証明が存在する。あるいは，帰納的部分関数を考慮に入れるならば，すべての帰納的部分関数はアルゴリズミックである。しかし，この逆の問いはむずかしい：すべてのアルゴリズムが帰納的か？帰納的でないアルゴリズムが存在するか（否定形）？

帰納的でないアルゴリズムが発見されるならば，この問題は否定的に解決されることになる。肯定的に解決するには，未だ欠けているだろう，アルゴリズムの全クラスの正確な特徴づけが要求される。実際，帰納的部分関数はまさにこの目的のために導入された。アルゴリズムとは，われわれがエフェクティヴに計算可能である（effectively computable）と認めるところの関数である。よって，一方に帰納的部分関数の正確な定義があり，他方に，アルゴリズムに関する直観的・主観的概念がある。

1936年にアロンゾ・チャーチはこの二つの概念を同一視しようという提案を行った。以後この提案はチャーチの提唱（Church's thesis）と呼ばれる[23]。それは，次のように表現できる：

（数論的）関数がアルゴリズミックである　⇔　それは帰納的である。

同様の提案はチューリング等によってもなされた[24]。

さて，チャーチの提唱を支持するいくつかの議論がある。

(1) プラグマティックな議論：これまでに知られているアルゴリズムがすべて帰納的であるという経験的事実を根拠とする議論がある。これまでの種々のアルゴリズムの経験において，帰納理論の技術を応用・実践した人々のすべてがこの提唱を受け入れるようになった。その結果，「チャーチの提唱に基づく証明」の伝統が生まれている[25]。それは，以下の形を取る：

何らかの方法で一定の関数が計算可能であることを確かめる。次いで，それが（部分）帰納的であるという結論に飛躍する。

(2) 計算可能性に関する概念分析による議論：これの実例はアラン・チューリングの基本論文にある（Turing [1936]）。チューリングは，人間の計算の手続きを抽象的コンピュータである

チューリング機械によって定式化された単純なステップに分解した。ガンディはこのチューリングの分析の方向を追究している[26]。

(3) 安定性による議論：さまざまな観点から提案された多数の計算可能性の概念[27]がすべて同等（同値）であることが示されている。同一概念について，互いに独立ではあるが同等である多数の定式化が存在するということは，当の概念の自然さと安定性を示唆すると考えられる。

ところで，チャーチの提唱において言及されているアルゴリズムは，本性上，機械的（mechanical）なものでなければならない。すなわち，それらは人間の実行者に対して創造性や発明の才を要求してはならない。その点で直観主義論理とのある種の親近性を有する。また，ここで問題にしている計算可能性の概念は抽象的なものであり，実行可能性（feasibility）の問題はチャーチの提唱と直接の関係はないが，計算機科学にとっては基本的に重要である。すなわち，時間またはテープの複雑性（complexity）は計算機の実行的計算能力と直結する問題である。多項式時間（polynomial time）による計算は実践的観点から望ましいものであるが，多くの重要な決定方法は指数関数的時間（exponential time）を要求する。しかし，帰納的関数の概念が歴史的に見てもきわめて重要なものであることに変わりはない。ゲーデルはその意義をこう述べている：

> 「［一般帰納性またはチューリング計算可能性の概念の］重要さは，この概念によって初めて，われわれが興味深い認識論的概念についての絶対的定義，すなわち選ばれた形式に依存しない定義を与えることに成功したということに大部分起因するものである，と私には思われる。」[28]

## § 4．応用と展望

計算可能性に関する理論としての帰納理論（リカーションセオリー）は算術の研究とともに発展したと言える。そこで，算術研究の応用面から考察していきたい。ゲーデルは，算術の十分な部分を含む理論は不完全であるということを示すために，帰納理論のメカニズムを利用した。その後の研究によって，算術を含む多くの理論が決定不可能であることが示された。それらに関連する方法と結果には以下のようなものがある。

### 4．1　算術の理論

算術の第一階理論**ＰＡ**（ペアノ算術：Peano's arithmetic）は，Ｓ(後者)，＋，・および**0**を含む言語を持ち，その公理は次のものである：

$$\mathrm{S}x \neq \mathbf{0}$$
$$\mathrm{S}x = \mathrm{S}y \supset x = y$$
$$x + \mathbf{0} = x$$
$$x + \mathrm{S}y = \mathrm{S}(x + y)$$
$$x \cdot \mathbf{0} = \mathbf{0}$$
$$x \cdot \mathrm{S}y = x \cdot y + x$$
$$\phi(\mathbf{0}) \wedge \forall x\,(\phi(x) \supset \phi(\mathrm{S}x)) \supset \forall x\,\phi(x) \qquad \text{(帰納法の図式)}$$

**ＰＡ**において，われわれは順序関係　$x < y :\Leftrightarrow \exists z\,(x + \mathrm{S}z = y)$ を定義でき，この関係が持つよく知られた性質を証明できる[29]。個々の自然数を表す記号は，$1 = \mathrm{S}\mathbf{0}$，$2 = \mathrm{SS}\mathbf{0}$，$3 = \mathrm{SSS}\mathbf{0}$，… として定義される。Ｒ．ロビンソンは帰納法の図式を一つの公理：

$x \neq 0 \supset \exists y\,(x = Sy)$

で置き換えて，有限的に公理化された**PA**の部分体系**Q**を導入した[30]。シェーンフィールドも有限的に公理化された**PA**の部分体系**N**を導入した[31]。この体系は，“<”を原始記号としており，帰納法の図式は次の公理によって置き換えられる：

$\neg(x < 0)$

$x < Sy \supset x < y \vee x = y$

$x < y \vee x = y \vee y < x$。

### 4. 2　算術化

構文論的性質がコードの原始帰納的述語となるような仕方で，われわれは算術の表現を自然数としてコード化できる。コード化を実際に行う方法は多数存在するが，われわれは§2でのコード化に基づくものを考える。伝統に従い，コードを**ゲーデル数**（Gödel numbers）と呼ぶ。

(1)　ゲーデル数をアルファベットの諸記号に配分する：

$x_i \longmapsto 2_i$，　$0 \longmapsto 1$，　$\vee \longmapsto 3$，　$\neg \longmapsto 5$，　$\exists \longmapsto 7$，　$S \longmapsto 9$，　$+ \longmapsto 11$，

$\cdot \longmapsto 13$，　$= \longmapsto 15$　　（**N**を考えるときは，$< \longmapsto 17$）

(2)　項（terms）tのゲーデル数$\ulcorner t \urcorner$は次のように定義される：

$\ulcorner x_i \urcorner = \langle 2_i \rangle$，$\ulcorner 0 \urcorner = \langle 1 \rangle$，$\ulcorner S\,t \urcorner = \langle 9, \ulcorner t \urcorner \rangle$，$\ulcorner (t + s) \urcorner = \langle 11, \ulcorner t \urcorner, \ulcorner s \urcorner \rangle$，$\ulcorner t \cdot s \urcorner = \langle 13, \ulcorner t \urcorner, \ulcorner s \urcorner \rangle$

(3)　式（formula）Aのゲーデル数$\ulcorner A \urcorner$は，次のように定義される：

$\ulcorner (t = s) \urcorner = \langle 15, \ulcorner t \urcorner, \ulcorner s \urcorner \rangle$，$\ulcorner (\phi \vee \psi) \urcorner = \langle 3, \ulcorner \phi \urcorner, \ulcorner \psi \urcorner \rangle$，

$\ulcorner \neg \phi \urcorner = \langle 5, \ulcorner \phi \urcorner \rangle$，$\ulcorner (\exists x_i \phi) \urcorner = \langle 7, \ulcorner x_i \urcorner, \ulcorner \phi \urcorner \rangle$。

表現にゲーデル数を配分する上の関数は原始帰納によって定義できる。よって，次の補題が成り立つ：

(4)　補題：以下の述語は原始帰納的である[32]。

(a)　nは変項のゲーデル数である，

(b)　nは項のゲーデル数である，

(c)　nは式のゲーデル数である。

(5)　一定の場所で，項や式における自由変項を追跡することが重要な場合がある。以下の述語は，「xはAにおいて自由である」ということを表現している（ここで，Aは項または式）：

「xはAそのものであるか，またはAは∃に依らずに二つの部分から構成されておりxはそれらの少なくとも一方で自由であるか，またはAは∃によって二つの部分から構成されていてxは第一の部分と同一ではなく第二の部分で自由である，等」

よって，次のように書ける：

$$Fr(m,n) \Leftrightarrow (m = n \wedge Vble(m)) \vee \exists x,y < n\,(n = \langle 3,x,y \rangle \wedge (Fr(m,x) \vee Fr(m,y))) \vee (\exists x,y < n\,(n = \langle 11, \cdots \rangle \cdots)) \vee (\cdots n = \langle 13, \cdots \rangle \cdots) \vee (\cdots n = \langle 15, \cdots \rangle \cdots) \vee \cdots \vee \exists x,y < n\,(n = \langle 7,x,y \rangle \wedge m \neq x \wedge Fr(m,y))。$$

再び，Frは原始帰納的である。

(6)　代入という操作を模倣する数論的関数Subを定義する。関数Subは，$Sub(\ulcorner A \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner) = \ulcorner A[t/x] \urcorner$を満たす（ここでAは項または式である）。

$$\mathrm{Sub}(m,\ n,\ k) = \begin{cases} k, & \mathrm{Vble}(m) \wedge m = n \text{ のとき；} \\ \langle 9,\ \mathrm{Sub}(p, n, k)\rangle, & m = \langle 9,\ p\rangle \text{ のとき；} \\ \langle 5,\ \mathrm{Sub}(p, n, k)\rangle, & m = \langle 5,\ p\rangle \text{ のとき；} \\ \langle a,\ \mathrm{Sub}(p, n, k),\ \mathrm{Sub}(q, n, k)\rangle, & m = \langle a, p, q\rangle \text{ で } a = 3, 11, 13, 15 \text{ のとき；} \\ \langle 7,\ p, \mathrm{Sub}(q, n, k)\rangle, & m = \langle 7, p, q\rangle \text{ かつ } p \neq n \text{ のとき；} \\ m, & \text{以上のどれでもないとき。} \end{cases}$$

Sub は原始帰納的である。

(7) **PA**の論理的基底（例えば，ヒルベルトタイプの体系か，シーケンス計算か等）に応じて，「n はゲーデル数mを持つ式の証明のゲーデル数である」ということを表現する原始帰納的述語：

$$\mathrm{Prov}(m,\ n)$$

を見出すことができる。

(8) 述語：$\mathrm{Thm}(m) :\Leftrightarrow \exists x\, \mathrm{Prov}(m, x)$ は，mが**PA**の定理のゲーデル数であることを表現している。存在量化子は，Thm を帰納的に可算である（recursively enumerable）とするが，帰納的（recursive）にはしない。

(9) 自然数は言わば二重生活を送っている。すなわち，算術の（形式的）理論に現れると同時にわれわれの現実的な直観世界にも生きている。算術の理論において，それは記号として，つまり数詞という姿で出現する。数nを表す数詞は$\overline{n}$で表される。特に$\overline{0} = \mathbf{0}$。(**PA**は**0**を定項記号として持っていた。) また，$\overline{n+1} = S(\overline{n})$。さて，数詞は記号である。それゆえ，数詞はゲーデル数を持つ。われわれは数nに数詞$\overline{n}$のゲーデル数を結びつける関数 Num をこう定義する：

$\mathrm{Num}(0) = \langle 1\rangle$

$\mathrm{Num}(n+1) = \langle 9,\ \mathrm{Num}(n)\rangle$。

それゆえ，$\mathrm{Num}(n) = \ulcorner \overline{n} \urcorner$。

(10) 次のクレイグの定理により，不必要な制限を避けることができる：

定理：すべての公理化可能な理論は，公理の帰納的集合によって公理化されうる。

ここで，理論が公理化可能である（axiomatizable）のは，その理論の公理の集合がRE（recursively enumerable）である場合である。$\phi_0, \phi_1, \phi_2, \cdots$を理論Tの諸公理の実効的枚挙（effective enumeration）とせよ（無限のリストを仮定しても何の制約もない）。そのとき，

$$\phi_0,\quad \phi_0 \wedge \phi_1,\quad \phi_0 \wedge \phi_1 \wedge \phi_2,\quad \cdots$$

もまたTを公理化する。よって，われわれは与えられた文$\sigma$がこの集合に属するかどうかをエフェクティヴにテストできる。というのは，公理の長さは正確に増大しているから，これらの公理の有限個のものがリストに挙げられた後，$\sigma$の長さを越える長さを持つ公理がリストに挙がった時点で，もしそのリスト中に$\sigma$が出現していないならば，それ以後リストに加わる公理の長さは$\sigma$の長さより大であるから，今後決して$\sigma$はリストに挙がらないことが分かる。公理化可能な算術の理論はRE理論と呼ばれる。公理化可能でない理論は事実上決定不可能である。なぜなら，それらの定理のクラスすらREでないからである。

### 4. 3　帰納的関数と述語の表現可能性

これまでの節（§2，3）では算術と論理を帰納理論に還元したが，これから逆に，帰納理論を算術の形式的理論に還元する。われわれは，**PA**内部で帰納的関数や関係について語ることができ

る（ただし，いく分複雑な仕方で）。便宜上，**Q**または**N**の公理化可能な拡大である理論Tを考察することによって，三つの算術理論**PA**，**Q**，**N**を同じ土俵上で扱う。

**4. 3. 1 定　義**

n項関数fがn＋1個の変項$x_1, \cdots, x_n, y$を持つ式$\phi$によって**表現される**（represented）のは，すべての$k_1, \cdots, k_n, m$に対して，

$$f(k_1, \cdots, k_n) = m \Leftrightarrow T \vdash \phi(\overline{k_1}, \cdots, \overline{k_n}, y) \equiv \overline{m} = y$$

の場合である。また，n項関係Rがn個の自由変項を持つ式$\phi$によって**表現される**のは，

$$\begin{cases} R(k_1, \cdots, k_n) \Rightarrow T \vdash \phi(\overline{k_1}, \cdots, \overline{k_n}), \\ R(k_1, \cdots, k_n) \text{ でない} \Rightarrow T \vdash \neg\phi(\overline{k_1}, \cdots, \overline{k_n}). \end{cases}$$

の場合である。この定義に関して次の基本的定理が成り立つ：

**定理**：すべての帰納的関数・関係は，体系**PA**，**Q**，**N**のいずれにおいても（よって，いかなるTにおいても）表現可能である。（証明略[33]）

この定理により，4.2節で導入した諸述語を上の意味で「表現する」，理論Tにおける式を手に入れることができる。特に，述語Prov を表現する式が存在する。便宜上，表現式に対しても述語と同じ記号を用いる（どちらを問題にしているかは文脈によって容易に判断できる）。

ところで，述語論理に対する健全性定理は，理論Tの定理がTのすべてのモデルで真となることを告げている。Tの証明可能な文がすべて標準モデルで真であるならば，われわれは$\mathbb{N}$において真である文を単に「真である」と呼ぶ。それでは逆に，すべての真なる文は**PA**で証明可能であろうか？1920年代末まではそのように希望され期待されていた。しかし，1931年にゲーデルがその期待を打ち破った。にも関わらず，この逆命題が重要な文のクラスに対して確立されることが望ましい場合もある。以下の定理がそのようなクラスの例を与える。その前に一つの規約を定める。われわれは，一つの式が，**有界量化子**（bounded quantifier）のみを含む式を従えた，存在（または全称）量化子の冠頭式であるとき，これを，$\Sigma^0_1$（または$\Pi^0_1$）と呼ぶ。

**4. 3. 2 定理（$\Sigma^0_1$完全性定理）**：$\Sigma^0_1$文$\phi$に対して，$\mathbb{N} \models \phi \Rightarrow \mathbf{PA} \vdash \phi$。

証明は$\phi$の構造に関する帰納法によって進む[34]。$\Pi^0_1$文に対しては，以下で見るように，真理性は証明可能性を含意しない。

われわれの算術の形式的体系**PA**では，関数記号としてはS，＋，・のみを原始記号とした。それならば，平方，巾，階乗といった関数を表す記号をなぜ導入しないのか？その理由は，そういった原始帰納的関数は，一般に算術の言語で定式化できるからである。すなわち，各原始帰納的関数fに対して，$f(\vec{m}) = n \Leftrightarrow \mathbf{PA} \vdash \phi(\vec{m}, y) \equiv y = n$であるばかりか，$\mathbf{PA} \vdash \forall \vec{x} \exists ! y \phi(\vec{x}, y)$でもある表現式$\phi(\vec{x}, y)$を見出すことができるからである。言語に関数記号Fと公理$\forall \vec{x} \phi(\vec{x}, F(\vec{x}))$を追加しても，本質的にその理論を強めたことにならない。それは，つぎの事実として表現される：拡大理論$T^*$が元の理論Tに対して**保存的である**（conservative）であるのは，Fを含まない式$\psi$に対して，$T \vdash \psi \Leftrightarrow T^* \vdash \psi$であるとき，かつそのときにかぎる。このとき，記号Fを消すために，$T^* \vdash \psi \Leftrightarrow \{\psi^0\} \cup T^* \vdash \psi \Leftrightarrow T \vdash \psi^0$となるような翻訳“ $^0$ ”を構成できる[35]。要するに，われわれはすべての原始帰納的関数を表すための関数記号と公理とを，**保存拡大**の形で**PA**に追加できる。メタ数学的観点からすれば，原始帰納的関数による保存拡大（conservative extension）を考えることは何ら害をもたらさない。例えば，決定可能性と完全性の観点からTとその保存拡大$T^*$は正確に同様に振る舞う。すなわち，$T^*$が決定可能（完全）である⇔Tが決定可能（完全）である。原始帰納的関数を表す関数記号が存在することによって，いくつかの結果の提

示がスムーズに行われる場合，算術の**定義的拡大** (definitional extension)[36]を利用できる。われわれはそのような拡大をもＰＡと呼ぶことにする。

### 4. 4　(第一) 不完全性定理とＰＡの決定不可能性

ゲーデルは嘘つきのパラドクスと類比性を持つ文を定式化した[37]。日常言語でパラフレーズすれば，それは「私はＰＡで証明できない」と語っている。この種の自己言及文の構造と導出可能性とを正確に定式化するために，ＰＡの持つ能力が最大限に活用される。その便利な手段が次の定理である：

**4. 4. 1　不動点定理** (fixed point theorem)

$\phi(x_0)$を，ただ一つの自由変項$x_0$を持つ式とする。そのとき，文$\psi$で，

$$\mathbf{PA} \vdash \psi \equiv \phi(\ulcorner \psi \urcorner)$$

であるものが存在する。

《証明》代入関数と表現可能性定理を利用する[38]。

この不動点定理は，ＰＡの任意の公理化可能な拡大体系に対して成り立つ。さて，われわれは，不動点定理を述語$\neg \exists y\, \mathrm{Prov}(x, y)$に適用することによって，直ちに**不完全性定理**を得る。すなわち，ある文$\psi$（不動点定理での$\phi(x)$を$\neg \exists y\, \mathrm{Prov}(x, y)$とするときの$\psi$）に対して，

$$\mathbf{PA} \vdash \psi \equiv \neg \exists y\, \mathrm{Prov}(\ulcorner \psi \urcorner, y) \qquad \cdots\cdots (※)$$

である。さて，$\mathbf{PA} \vdash \psi$とする。すると，Provの表現可能性より，$\mathbf{PA} \vdash \mathrm{Prov}(\ulcorner \psi \urcorner, \overline{k})$。ここで，kは$\psi$の実際の証明のゲーデル数である。よって，$\mathbf{PA} \vdash \exists y\, \mathrm{Prov}(\ulcorner \psi \urcorner, y)$。しかし，仮定と（※）から，$\mathbf{PA} \vdash \neg \exists y\, \mathrm{Prov}(\ulcorner \psi \urcorner, y)$。ゆえに，ＰＡは矛盾することになる。それゆえ，ＰＡが無矛盾であると仮定すると，$\mathbf{PA} \nvdash \psi$でなければならない。

もし$\mathbf{PA} \vdash \neg \psi$とする。ところで，$\mathbf{PA} \nvdash \psi$であるから，どんな$n \in \mathbb{N}$についても$\mathrm{Prov}(\ulcorner \psi \urcorner, n)$ではない。よって，任意の$n \in \mathbb{N}$で，$\mathbf{PA} \vdash \neg \mathrm{Prov}(\ulcorner \psi \urcorner, \overline{n})$。ところが，仮定と（※）より$\mathbf{PA} \vdash \exists y\, \mathrm{Prov}(\ulcorner \psi \urcorner, y)$，すなわち$\mathbf{PA} \vdash \neg \forall y \neg \mathrm{Prov}(\ulcorner \psi \urcorner, y)$。ここで，ＰＡが$\omega$無矛盾である（つまり，“任意の$n \in \mathbb{N}$について$\mathbf{PA} \vdash \sigma(\overline{n})$，しかし$\mathbf{PA} \vdash \neg \forall x\, \sigma(x)$”ということがない）と仮定すると，$\sigma(x)$として$\neg \mathrm{Prov}(\ulcorner \psi \urcorner, x)$を取ると$\omega$矛盾である。よって，$\mathbf{PA} \nvdash \neg \psi$。$\omega$無矛盾性は無矛盾性を含意するから，ＰＡが$\omega$無矛盾であると仮定すると，$\psi$も$\neg \psi$もＰＡでは証明できない。よって，ＰＡは不完全である。(ここで，上のゲーデル文$\psi$が$\Pi^0_1$であることに注意しよう。)

**4. 4. 2　考　　察**

(1)　ＰＡはＱやＮの拡大体系であるから，われわれはＱやＮの不完全性をも確立したことになる。原始帰納的関数が全く用いられていないようなゲーデル文の言い換えによって，すべての無矛盾な公理化可能なＱ（またはＮ）の拡大Ｔが不完全であることを示すこともできる。

(2)　ロッサーは不動点定理を別の式に適用することによって，$\omega$無矛盾性への依存を無矛盾性への依存へと弱めた[39]。その別の式が意味するのは「私の否定形の証明が存在し，私のいかなる証明もそれに先行しない」である。記号では，$\exists y\, [\mathrm{Prov}(\mathrm{neg}x, y) \wedge \forall z < y \neg \mathrm{Prov}(x, z)]$ (ここで，negは，$\mathrm{neg}(\ulcorner \phi \urcorner) = \ulcorner \neg \phi \urcorner$であるような原始帰納的関数である)。この式全体はＰＡの保存拡大の言語で定式化される。この式を$R(x)$として，不動点定理を適用する：$\mathbf{PA} \vdash \psi \equiv R(\ulcorner \psi \urcorner)$。読みやすさのためにＰＡへの言及は省略する。すると，

$$\vdash \psi \equiv \exists y\, [\mathrm{Prov}(\ulcorner \neg \psi \urcorner, y) \wedge \forall z < y \neg \mathrm{Prov}(\ulcorner \psi \urcorner, z)]。 \qquad (1)$$

$\vdash \psi$と仮定する。すると，ある自然数nに対してProv($\overline{\ulcorner\psi\urcorner}$, n)が成り立つ。それゆえ，

$\vdash$Prov ($\overline{\ulcorner\psi\urcorner}$, $\overline{n}$)。　(2)

また，(1)と仮定から，

$\vdash \exists y$ (Prov($\overline{\ulcorner\neg\psi\urcorner}$, y)$\wedge\forall z<y \neg$Prov($\overline{\ulcorner\psi\urcorner}$, z)。　(3)

(2)と(3)から次が導かれる：

$\vdash \exists y<n$ Prov($\overline{\ulcorner\neg\psi\urcorner}$, y) 。　(4)

$\vdash y<\overline{n} \equiv y=0 \vee y=1 \vee\cdots\vee y=\overline{n-1}$を用いることによって，われわれは$\vdash$Prov($\overline{\ulcorner\neg\psi\urcorner}$, $\overline{0}$)$\vee\cdots\vee$ Prov($\overline{\ulcorner\neg\psi\urcorner}$, $\overline{n-1}$)を見出す。有界量化子のみを持つ文$\sigma$と$\tau$に対して，容易に，$\vdash\sigma\vee\tau\Rightarrow\vdash\sigma$または$\vdash\tau$を確立しうる。これにより，あるm<nに対して$\vdash$Prov($\overline{\ulcorner\neg\psi\urcorner}$, $\overline{m}$) を生み出す。このことから，$\vdash\neg\psi$。しかし，仮定$\vdash\psi$を考慮すると，これは**PA**の無矛盾性に反する。よって，$\vdash\neg\psi$とする。すると，$\nvdash\psi$，すなわち，

任意のnに対して　$\vdash\neg$Prov ($\overline{\ulcorner\psi\urcorner}$, $\overline{n}$)。　(5)

他方，$\vdash\neg\psi$より，(1)から$\vdash\forall y$ (Prov ($\overline{\ulcorner\neg\psi\urcorner}$, y)$\supset\exists z<y$ Prov ($\overline{\ulcorner\psi\urcorner}$, z))，および，ある自然数mについて，$\vdash$Prov($\overline{\ulcorner\neg\psi\urcorner}$, $\overline{m}$)が得られる。これより，$\vdash\exists z<m$ Prov($\overline{\ulcorner\psi\urcorner}$, z)，よって以前と同様，あるk<mにつき，$\vdash$Prov ($\overline{\ulcorner\psi\urcorner}$, $\overline{k}$) を得る。これは(5)と結びつくことにより，**PA**の無矛盾性に反する。従って，$\nvdash\psi$　かつ　$\nvdash\neg\psi$　が結論となる。

(3) ゲーデル文$\psi$を標準モデル$\mathbb{N}$で解釈することによって，$\psi$が真であることが分かる。よって，ゲーデル文は，「真ではあるが証明できない文」の一例となる。

(4) 上の不完全性定理から，$\{\overline{\ulcorner\phi\urcorner} \mid \mathbb{N}\models\phi\}$，すなわち算術の真なる文（のゲーデル数）の集合は決定可能（つまり帰納的）ではない，と結論できる。というのは，もし仮に決定可能だとすると，それらの真なる文を公理として用いることができるが，この新しい体系Trに対して不完全性定理を繰り返すことができる，すなわち　Tr$\nvdash\phi$　かつ　Tr$\nvdash\neg\phi$　であるような文$\phi$が存在する。しかし，これは不可能である。なぜなら，真なる文と偽なる文が文のすべてを尽くすからである。よって，Trは帰納的（決定可能）ではない。

(5) 証明可能性の述語によって，**Q**または**N**の公理化可能な拡大において表現可能な述語・関数は帰納的である，ということを容易に示すことができる。すなわち，$\vdash\overline{m}=\overline{n}\Rightarrow m=n$であるような，**0**とSを持つ公理化可能な理論において，表現可能な述語・関数は帰納的である。このことは，帰納的関数のもう一つの特徴づけを与える。つまり，帰納的関数のクラスは**PA**で表現可能な関数のクラスと一致する。この条件を満たす理論は数値的（numerical）と呼ばれる。

(6) 算術化のテクニックと帰納的関数の表現可能性によって，われわれは**Q**と**N**（よって**PA**）の拡大の決定不可能性を証明できる。**Q**と**N**は本質的に決定不可能であると呼ばれる。

同じ方法によって，われわれは一般的結果を証明できる：すべての帰納的関数が表現可能であるような数値的理論は決定不可能である。さらに，われわれは算術における真理の定義不可能性についての，以下のタルスキの定理を導出できる。もし，T$\vdash\phi\equiv\tau$($\overline{\ulcorner\phi\urcorner}$) ならば，数値的理論Tにおいて，式$\tau$は**真理定義**（truth definition）であると言われる。

**4. 4. 3　タルスキの定理：**

**Q**または**N**の，無矛盾で公理化可能などんな拡大も，真理定義を持たない。

《証明》不動点定理を利用して行うことができる(40)。

**4. 4. 4　定理**：もしTが**Q**（または**N**）の無矛盾な拡大ならば，そのときTは決定不可能である。

**系（チャーチの定理）**：第一階の述語論理は決定不可能である。

《証明》少なくとも算術の言語を含む第一階言語を考える。**Q**は有限的に公理化可能であるから$\mathbf{Q}\vdash\phi\Rightarrow\vdash\alpha\supset\phi$が成り立つ（ここで，$\alpha$は**Q**の公理の連言とする）。このとき，明らかに，第一階述語論理に対する任意の決定手続きが**Q**に対する決定手続きを与える。しかるに，**Q**に対する決定手続きは存在しない。よって，第一階述語論理の決定手続きも存在しない（Q. E. D.）。

### 4. 5　決定可能理論と決定不可能理論

今日，多くの理論が決定不可能であることが知られている(41)。この場合，基底にある論理が決定可能性の結果においてある重要な役割を果たすことがあることに注意すべきである。例えば，古典的単項述語論理は決定可能であるのに対して，直観主義的単項述語論理は決定可能ではない(42)。述語論理の決定可能性の局面は広く研究されている。最も古い結果の一つが単項述語論理の決定可能性である（Löwenheim：1915, Behmann：1922)。そして，その頂点が述語論理の決定不可能性である（Church：1936)。この特定の領域では決定問題に関する可解と非可解のケースがよく研究されている。

構文論的基準によって与えられる式のクラスΓで，各式$\phi$に，$\vdash\phi\Leftrightarrow\vdash\phi^{*}$であるような$\phi^{*}\in\Gamma$を結びつける還元手続きによってΓの決定不可能性が確立されているようなものが存在する。そのようなクラスΓは「証明可能性（または妥当性）の観点からの還元クラス」と呼ばれる。このとき，使われる構文論的基準は，例えば①述語の引数の個数，②述語の個数，③冠頭標準形での接頭量化子の個数，④同じ形式での量化子の変化の数，などである。ここで，ある表記法を導入する：$Q_1{}^{n_1}\cdots Q_m{}^{n_m}$は，$n_1$個の量化子$Q_1$，$n_2$個の量化子$Q_2$，…，$n_m$個の量化子$Q_m$を接頭辞として持つすべての冠頭式のクラスを表すとする。右肩指数$^{\infty}$は，任意有限の長さの量化子の区画を表示するとする。述語記号の性質に関する制限は有限列で示す。例えば，(0, 2, 1) は0個の単項述語，2個の二項述語，1個の三項述語を表す。この二つの表記法を結びつけることで，$\forall^{\infty}\exists_2(0,1)$や$\exists^{2}\forall^{1}\exists^{3}(2,1)$といった明確なクラスの表現を得る（$^{\infty}$は「すべての有限なn」を意味する)。身近な例として，スコーレム標準形によって与えられる$\forall^{\infty}\exists^{\infty}$式のクラスは充足可能性に対する還元クラスである。つまりこのクラスは存在量化子を従えた全称量化子を持つ冠頭式から成る。すでに多くの還元クラスの例が研究されている(43)し，量化式一般の非可解クラスの研究もある(44)。

他方，理論の決定可能性を示す方法として量化子消去の方法がある。理論Tが量化子を消去可能であるのは，自由変項$x_1,\cdots,x_n$を持つ任意の式　$\phi(x_1,\cdots,x_n)$　に対して，

$$\vdash\phi(x_1,\cdots,x_n)\equiv\psi(x_1,\cdots,x_n)$$

であるような開いた（よって量化子を含まない）式$\psi(x_1,\cdots,x_n)$が存在する場合である。それゆえ，量化子が消去できる理論に対しては，ある式に対応する開いた式の導出可能性を問題とすればよい。通常の導出問題よりこれは容易であるので，量化子消去によって決定手続きが生み出されることがある。初期の目覚ましい結果は，プレスブルガー（1930）が，後者関数と加法のみを持つ算術理論が量化子消去によって決定可能であることを示したことである。後に，タルスキは実数の閉じた体に対して量化子消去から成る決定方法が存在することを示した(45)。以来，量化子消去は決定手続き構成の有力な方法となっている(46)。

### 4. 6　算術的階層

算術化という符号化（コーディング）によって「nは集合Xに属するか？」という決定問題は，$\mathbb{N}=\{0,1,2,\cdots\}$の部分集合の決定問題として提示しうるが，通常は，一定のエフェクティヴに生成された集合の要

素が他の同様な集合の要素であるかを決定せよ，という形で考えられる。つまり，われわれの関心は，$\mathbb{N}$の部分集合の間である仕方でエフェクティヴに記述される一定集合（およびその間の関係）に存する。われわれが既に出会ったそのような集合は，原始帰納的集合，帰納的集合，帰納的に枚挙可能な（RE）集合である。一般に，一つの理論または構造において定義可能な自然数の集合を考察するのが自然である。そのような集合の最初の候補は算術の標準モデルである$\mathbb{N}$である。$\mathbb{N}$で定義可能な集合は**算術的**（arithmetic）と呼ばれ，$\{n \mid \mathbb{N} \models \phi(\overline{n})\}$という形をしている。帰納的集合とRE集合は算術的である。さらに，算術的でない集合が既に登場した。それは算術の真なる文（のゲーデル数）の集合である（タルスキの定理）。

そこで，算術的集合のクラスがある構造，特に複雑さの度合いの観点からの構造を持つかどうかが問題となる。われわれは，算術的集合を構文論的観点から分類できる。**算術的階層**（arithmetical hierarchy）と呼ばれるその分類は，階層を定義する式の冠頭標準形に基づいている。階層は次のように定義される：

$\Sigma^0_1$式　は　$\exists \vec{y}\, \phi\ (\vec{x}, \vec{y})$という形をしている；

$\Pi^0_1$式　は　$\forall \vec{y}\, \phi\ (\vec{x}, \vec{y})$という形をしている（$\phi$は有界量化子しか含まない）。

もし$\phi\ (\vec{x}, \vec{y})$が$\Sigma^0_n$式ならば，$\forall \vec{y}\, \phi\ (\vec{x}, \vec{y})$は$\Pi^0_{n+1}$式である；

もし$\phi\ (\vec{x}, \vec{y})$が$\Pi^0_n$式ならば，$\exists \vec{y}\, \phi\ (\vec{x}, \vec{y})$は$\Sigma^0_{n+1}$式である。

$\Sigma^0_n$と$\Pi^0_n$は対応する式の外延として定義され，集合が$\Delta^0_n$であるのは，それが$\Sigma^0_n$でも$\Pi^0_n$でもある場合である。

以下の包含が確立されている：

$$\begin{array}{ccccccccccccc}
 & \subsetneq & \Sigma^0_1 & \subsetneq & & \subsetneq & \Sigma^0_2 & \subsetneq & & \subsetneq & \Sigma^0_n & \subsetneq & & \subsetneq & \Sigma^0_{n+1} & \subsetneq & \\
\Delta^0_1 & & & & \Delta^0_2 & & & & \cdots\cdots & & & & \Delta^0_{n+1} & & & & \cdots \\
 & \supsetneq & \Pi^0_1 & \subsetneq & & \supsetneq & \Pi^0_2 & \subsetneq & & \supsetneq & \Pi^0_n & \subsetneq & & \supsetneq & \Pi^0_{n+1} & \subsetneq &
\end{array}$$

例えば，もし$X \in \Sigma^0_1$ならば，Xは式$\exists \vec{y}\, \phi\ (x, \vec{y})$で定義され，ダミーの変項と量化子を追加することで$\Pi^0_2$式である$\forall z \exists \vec{y}\, \phi\ (x, \vec{y})$が得られる。

上の分類は構文論的基準に基づいているが，帰納理論による分類も存在する。それによる階層の末端において，次の対応がある：

$\Delta^0_1$ …… 帰納的集合，

$\Sigma^0_1$ …… RE集合，

$\Pi^0_1$ …… RE集合の補集合（$\Sigma^0_1 \cap \Pi^0_1 = \Delta^0_1$）。

モデルの複雑さもまた算術的階層によって測られる。すなわち，モデルの宇宙（対象領域）がすべての自然数の集合で，関係が$\Sigma^0_n$（または$\Pi^0_n$，$\Delta^0_n$）のとき，そのモデルは$\Sigma^0_n$（または$\Pi^0_n$，$\Delta^0_n$）と呼ばれる。

最後に，いくつかの結果に言及して小論を閉じることにする。

（i）ヒルベルト＝ベルナイス完全性定理：（可算な言語による）理論が公理のRE集合を持ち，かつ無矛盾ならば，その理論は$\Delta^0_2$モデルを持つ。

（ii）決定可能な理論は帰納的モデルを持つ。

（iii）算術の唯一の帰納的モデルは標準モデルである。

（iv）1970年にマチャセヴィッツはヒルベルトの第10問題に否定的解を与えた(47)：
与えられた任意のディオファントス型方程式（すなわち，整数係数を持つ$p(x_1, \cdots, x_n) = 0$という形の多項式）が解を持つかどうかを決定するアルゴリズムは存在しない。

# 註

(1) 哲学的な側面については Carnap [1937] の議論を参照。また，Fraenkel et al. [1973]（特に280頁以下）参照。

(2) 自然数 a，b（a＞b）に対して，a を b で割った余りを r とし，次に b を r で割った余りを $r_1$ とし，さらに r を $r_1$ で割った余りを $r_2$ とし，…，以下同様な割り算を実行する。すなわち，

$$a = b \cdot q + r \qquad (0 \leqq r < b)$$
$$b = r \cdot q_1 + r_1 \qquad (0 \leqq r_1 < r)$$
$$r = r_1 \cdot q_2 + r_2 \qquad (0 \leqq r_2 < r_1)$$
$$\cdots \qquad \cdots$$
$$r_{n-2} = r_{n-1} \cdot q_n + r_n \qquad (0 \leqq r_n < r_{n-1})$$
$$r_{n-1} = r_n \cdot q_{n+1} + r_{n+1} \qquad (0 \leqq r_{n+1} < r_n)$$
$$r_n = r_{n+1} \cdot q_{n+2} + r_{n+2} \qquad (0 \leqq r_{n+2} < r_{n+1})$$

この計算過程の最後の余り $r_{n+2}$ が 0 であれば，a と b の最大公約数（g．c．d．＝greatest common divisor）は $r_{n+1}$ である。これがユークリッドの互除法である。$r_{n+2} = 0$ のとき，$r_{n+1}$ が a，b の公約数であることはこの計算過程を逆に辿ることによって示され，また最大であることも証明できる。さらに，この過程で，被除数が除数より常に大であり，除数が剰余より常に大であることから，$a > b > r > r_1 > \cdots > 1 > 0$ となり，最大 r＋1 個の割り算を実行した後，計算は停止することになる。具体的な数値例として，4458 と 1724 の最大公約数を求める：

4458＝1724・2＋1010，　　1724＝1010・1＋714，
1010＝714・1＋296，　　714＝296・2＋122
296＝122・2＋52，　　122＝52・2＋18，
52＝18・2＋16，　　18＝16・1＋2，　　16＝2・8＋0。

よって，2が4458と1724の最大公約数である。これを逆にたどる：16＝2・8，18＝2・8＋2＝2a（ここで，8＋1を a とおく。以下同様に2を括り出す），52＝2a・2＋16＝2b，122＝2b・2＋2a＝2c，296＝2c・2＋2b＝2d，714＝2d・2＋2c＝2e，1010＝2e・1＋2d＝2f，1724＝2f・1＋2e＝2g，4458＝2g・2＋2f＝2h。こうして，2が4458と1724の公約数であることが示された。最大性は次のように示される。$2 < \alpha$ である公約数 $\alpha$ が存在すると仮定する。4458＝$\alpha$h，1724＝$\alpha$j とおいて，互除法の計算過程に沿った推論を行う：
4458＝$\alpha$h＝1724・2＋1010＝$\alpha$j＋1010，∴1010＝$\alpha k_1$ とおける。
1724＝$\alpha$j＝1010・1＋714＝$\alpha k_1$＋714，∴714＝$\alpha k_2$ とおける。
1010＝$\alpha k_1$＝714・1＋296＝$\alpha k_2$＋296，∴296＝$\alpha k_3$ とおける。
714＝$\alpha k_2$＝296・2＋122＝$\alpha k_3$・2＋122，∴122＝$\alpha k_4$ とおける。
296＝$\alpha k_3$＝122・2＋52＝$\alpha k_4$・2＋52，∴52＝$\alpha k_5$ とおける。
122＝$\alpha k_4$＝52・2＋18＝$\alpha k_5$・2＋18，∴18＝$\alpha k_6$ とおける。
52＝$\alpha k_5$＝18・2＋16＝$\alpha k_6$・2＋16，∴16＝$\alpha k_7$ とおける。
18＝$\alpha k_6$＝16・1＋2＝$\alpha k_7$＋2，∴2＝$\alpha k_8$ とおける。つまり，$\alpha$ は2の約数でもある。よって，$\alpha \leqq 2$。これは，$2 < \alpha$ という仮定と矛盾する。よって，2が最大公約数である。

(3) 回文（または回語）とは，“bob”，“boaob”，“たけやぶやけた”，“Able was I ere I saw Elba” のように，左から読んでも右から読んでも（上から読んでも下から読んでも）同一の文（または語）となる文字列である。与えられた（例えば5000文字から成る）記号の有限列が回文であるかどうかの判定手続きは次のようなものである。「最初の記号と最後の記号が同一かどうかを確認せよ。もし同一であれば，それらを消し，短くなった記号列について同様のことを行え。もし同一でなければ，回文ではない。」この操作は有限回で停止する。すべての記号が消されるか，ただ一つの記号が残れば，最初の記号列は回文であり，そうでなければ回文ではない。この操作が正しいことは回文の定義から明らかである。

(4) チューリング機械は Turing [1936] により初めて定式化された。この論文は Davis [1965] に再録されている。

(5) 規約の必要性は，例えば，機械が無限に操作し続けることなく計算終了後に停止するように機械を調整する，等の

場面で表面化する。以下に主要な規約を挙げる。

(1) 当初，有限個のマス目を除いて，テープは空白の状態で提示される。

(2) 計算を開始するとき，機械は空白でないマス目の左端を眺めている。

(3) ある状態で，いかなる指令も実行できない記号を読み取ったとき，機械は停止する。

(4) 任意の状態および読み取られた記号に対して，適用される指令はたかだか一個である，すなわち機械は決定論的（deterministic）に動く。

(6) それ以降の計算過程を一連の状況記述として示すと以下のようになる：

$$\mathrm{B\,a\,a\,b\,a\,b\,a\,B}\rightarrow\mathrm{B\,a\,a\,b\,a\,b\,a\,B}\rightarrow\mathrm{B\,B\,a\,b\,a\,b\,a\,B}\rightarrow\mathrm{B\,a\,b\,a\,b\,a\,B}\rightarrow\ \cdots$$
$$\mathrm{q_2}\qquad\qquad\mathrm{q_3}\qquad\qquad\mathrm{q_0}\qquad\qquad\mathrm{q_a}$$
$$\rightarrow\mathrm{B\,a\,b\,a\,b\,a\,B}\rightarrow\mathrm{B\,a\,b\,a\,b\,a\,B}\rightarrow\mathrm{B\,a\,b\,a\,b\,B\,B}\rightarrow\ \cdots\rightarrow$$
$$\mathrm{q_a}\qquad\qquad\mathrm{q_1}\qquad\qquad\mathrm{q_2}$$
$$\mathrm{B\,a\,b\,a\,b\,B}\rightarrow\mathrm{B\,a\,b\,a\,b\,B}\rightarrow\mathrm{B\,B\,b\,a\,b\,B}\rightarrow\mathrm{B\,b\,a\,b\,B}\rightarrow\ \cdots\rightarrow\mathrm{B\,b\,a\,b\,B}\rightarrow$$
$$\mathrm{q_2}\qquad\qquad\mathrm{q_3}\qquad\qquad\mathrm{q_0}\qquad\qquad\mathrm{q_b}\qquad\qquad\mathrm{q_b}$$
$$\mathrm{B\,b\,a\,b\,B}\rightarrow\mathrm{B\,b\,a\,B\,B}\rightarrow\mathrm{B\,b\,a\,B}\rightarrow\mathrm{B\,b\,a\,B}\rightarrow\mathrm{B\,b\,a\,B}\rightarrow\mathrm{B\,B\,a\,B}\rightarrow\mathrm{B\,a\,B}\rightarrow$$
$$\mathrm{q_4}\qquad\mathrm{q_2}\qquad\mathrm{q_2}\qquad\mathrm{q_2}\qquad\mathrm{q_3}\qquad\mathrm{q_0}\qquad\mathrm{q_a}$$
$$\mathrm{B\,a\,B}\rightarrow\mathrm{B\,B\,B}\rightarrow\mathrm{B\,B\,B}\ (\mathrm{END}).$$
$$\mathrm{q_1}\quad\mathrm{q_2}\qquad\mathrm{q_3}$$

(7) 途中の計算過程は次のようになる（“…→”で→の有限個のステップを表す）：

$$\mathrm{B\,q_0\,|\,|\,|\,B\,|\,|\,B}\cdots\rightarrow\mathrm{B\,|\,|\,|\,B\,q_1\,|\,|\,B}\cdots\rightarrow\mathrm{B\,|\,|\,|\,B\,|\,|\,q_1\,B}\rightarrow$$
$$\mathrm{B\,|\,|\,|\,B\,|\,q_2\,|\,B}\rightarrow\mathrm{B\,|\,|\,|\,B\,q_3\,|\,a\,B}\rightarrow\mathrm{B\,|\,|\,|\,q_3\,B\,|\,a\,B}\rightarrow$$
$$\mathrm{B\,|\,|\,q_4\,|\,B\,|\,a\,B}\rightarrow\mathrm{B\,|\,|\,B\,q_5\,B\,|\,a\,B}\cdots\rightarrow\mathrm{B\,|\,|\,B\,B\,q_2\,|\,a\,B}\cdots\rightarrow$$
$$\mathrm{B\,|\,B\,B\,B\,q_5\,a\,a\,B}\rightarrow\mathrm{B\,|\,B\,B\,q_2\,B\,a\,a\,B}\ (\mathrm{END}).$$

(8) 射影関数を表すチューリング機械の指令は以下のようなものになる：

$\mathrm{q_0\,|\,B\,q_0\,R}$
$\mathrm{q_0\,B\,B\,q_1\,R}$ ——1番目の|の列をすべて消し，右側の2番目の列に移れ；

$\mathrm{q_1\,|\,B\,q_1\,R}$
$\mathrm{q_1\,B\,B\,q_2\,R}$ ——2番目の|の列をすべて消し，右側の3番目の列に移れ；

$\vdots$

$\mathrm{q\,i\ \ |\,B\,q\,i'\,R}$
$\mathrm{q\,i'\ |\,|\,q\,i'\,R}$
$\mathrm{q\,i'\,B\,B\,q_{i+1}\,R}$ ——i＋1番目の列からは一本だけ|を消し，i＋2番目の列に移れ；

$\vdots$

$\mathrm{q_n\,|\,B\,q_n\,R}$ ——n＋1番目の列のすべての|を消して停止せよ。

(9) Turing [1936] §§6，7参照。

(10) 元の加法計算で0＋1＝1の計算過程は次のようになる：

$$\mathrm{q_0\,|\,B\,|\,|}\rightarrow\mathrm{B\,q_1\,B\,|\,|}\rightarrow\mathrm{B\,B\,q_1\,|\,|}\rightarrow\mathrm{B\,B\,B\,q_2\,|}\ (\mathrm{END}).$$

これを新しい機械は次のような計算過程によって模倣する：

$$\$_1-\$_2\ \overline{q_0}\ q_0\ |\ B\ \|\ \$_3\cdots\rightarrow\$_1-\$_2\ B\ \overline{q_k}\ q_1\ B\ \|\ \$_3\cdots\rightarrow\ \cdots\ \cdots\rightarrow\$_1-\$_2$$
$$B\,B\,\overline{q\,i}\ q_1\ \|\ \$_3\cdots\rightarrow\$_1-\$_2\ B\,B\,B\,\overline{q\,j}\ q_2\ |\ \$_3\cdots\rightarrow\$_2\ B\,B\,B\,\overline{q\,j}\ q_2\ |\ \$_3.$$

(11) Turing [1936] §8参照。

(12) ②$x\cdot y$には，$x\cdot 0=0$，$x\cdot(y+1)=x\cdot y+x$，という原始帰納による定義を与えることができ，これをさらに，$\times(0,x)=C^1_0(x)=0$，$\times(y+1,x)=+(P^3_1(\times(y,x),x,y),P^3_2(\times(u,x),x,y))$と明示的に書き直すことができる（定義図式でhに当たる関数は＋と$P^3_1$，$P^3_2$の合成である）。以下，明示的ではなく伝統的な定義を与える。③$x^y$は，$x^0=1$，$x^{y+1}=x^y\cdot x$，④$p(0)=0$，$p(x+1)=x$，⑤$x\dot{-}0=x$，$x\dot{-}(y+1)=p((x\dot{-}y))$ ⑥$0!=1$，$(x+1)!=(x!)\cdot(x+1)$，⑦$sg(0)=0$，$sg(x+1)=1$，⑧$\overline{sg}(x)=1\dot{-}sg(x)$，⑨$|x-y|=(x\dot{-}y)+(y\dot{-}x)$，⑩$\Sigma^y_{i=0}g(\vec{x},i)=g(\vec{x},0)+\cdots+g(\vec{x},y)$，⑪$\Pi^y_{i=0}g(\vec{x},i)=g(\vec{x},0)\cdot\cdots\cdot g(\vec{x},y)$。

(13) 実際，①$K_{\phi}(x)=0$，②$K_E(0)=1$，$K_E(x+1)=\overline{sg}(K_E(x))$，③$K_{=}(x,y)=\overline{sg}(|x-y|)$，④$K_{<}(x,y)=sg(y \dot{-} x)$であり，これらの特徴関数はいずれも原始帰納的である。

(14) いまC＝A∩Bとおく。そのとき，$x\in C \Leftrightarrow x\in A \wedge x\in B$。そこで，これらの特徴関数を取ると，$K_C(x)=1 \Leftrightarrow K_A(x)=1 \cdot K_B(x)=1$であるから，$K_C(x)=K_A(x)\cdot K_B(x)$と定義できる。明らかに，特徴関数$K_C$つまり$K_{A\cap B}$は原始帰納的である。合併集合に対しては，$K_{A\cup B}(x)=sg(K_A(x)+K_B(x))$とし，補集合に対しては$\overline{sg}(K_A(x))$とおけばよい。

(15) 証明：$K_S(\vec{x})=K_R(f_1(\vec{x}),\cdots, f_p(\vec{x}))$。

(16) 証明：$R_1,\cdots, R_p$が相互排反的だから，$K_{Ri}(\vec{x})=1$のとき，他の関係のすべての特徴関数は$\vec{x}$に対して値0を取る。ゆえに，$f(\vec{x})=g_1(\vec{x})\cdot K_{R1}(\vec{x})+\cdots+g_p(\vec{x})\cdot K_{Rp}(\vec{x})$。

(17) 証明：以下の表を考える。

| R | $R(\vec{x},0)$ | $R(\vec{x},1)$ | … | $R(\vec{x},i-1)$ | $R(\vec{x},i)$ | $R(\vec{x},i+1)$ | … | $R(\vec{x},m)$ |
|---|---|---|---|---|---|---|---|---|
| $K_R$ | 0 | 0 | … | 0 | 1 | 0 | … | 1 |
| g | 0 | 0 | … | 0 | 1 | 1 | … | 1 |
| h | 1 | 1 | … | 1 | 0 | 0 | … | 0 |
| f | 1 | 2 | … | i | i | i | … | i |

水平線より下の第一行目に，$0\leqq i\leqq m$に対する特徴関数の値　$K_R(\vec{x},i)$を書く。第二行目でそこまでの値の単調集積：$g(\vec{x},i)=sg\,\Sigma^{i}_{j=0}K_R(\vec{x},j)$を作る。第三行目で0と1を交換する：$h(\vec{x},i)=\overline{sg}(g(\vec{x},i))$。そして，第四行目でそこまでのhの値を足し合わせる：$f(\vec{x},i)=\Sigma^{i}_{j=0}h(\vec{x},j)$。もし$R(\vec{x},j)$がj＝iで初めて成り立つとき，$f(\vec{x},m-1)=i$である。そして，もしj＜mであるどんなjに対しても$R(\vec{x},j)$が成り立たないとき，$f(\vec{x},m-1)=m$　となる。それゆえ，$\mu y<m\ R(\vec{x},m)=f(\vec{x},m-1)$，よって有界最小化は原始帰納的関数を生み出す。

(18) これはクリーニに基づく。彼は未定義項の文脈における同等性を表す記号として記号“≃”を導入した。≃を支配する規則は次のものである：t≃sならば，tとsは同時に定義されており同一であるか，またはそれら（tとs）は同時に未定義である。tとsの少なくとも一方が未定義であれば，t＝sは未定義であるが，t≃sは，一方が未定義のとき他方が未定義ならば真，定義されておれば偽である，という点が＝と≃の違いである。クリーニは≃を「完全同等性」，＝を「弱い同等性」と呼ぶ（Kleene［1952］327-8頁）。

(19) 詳しい証明は次の通り。最初の関数$S^1_n$はR６によって与えられるが，その明示的定義を，代入の図式によって，

$S^1_n(e,y)=\langle 4,(e)_2-1,e,\langle 0,(e)_2-1,y\rangle,\langle 1,(e)_2-1,1\rangle,\cdots,\langle 1,(e)_2-1,n-1\rangle\rangle$

と書き下す。このとき，

$$|S^1_n(e,y)|(\vec{x})=z \Leftrightarrow \exists q_1\cdots q_n\,[|\langle 0,(e)_2-1,y\rangle|(\vec{x})=q_1\wedge|\langle 1,(e)_2-1,1\rangle|(\vec{x})=q_2\wedge\cdots\wedge|\langle 1,(e)_2-1,n-1\rangle|(\vec{x})=q_n\wedge|e|(q_1,\cdots,q_n)=z]。$$

節R１，R２によって，$q_1=y$，$q_{i+1}=x_i$。よって，$|S^1_n(e,y)|(\vec{x})=|e|(y,\vec{x})$を得る。eが固定されたインデックスであるから，$S^1_n$は原始帰納的である。$S^m_n$は$S^1_n$をm回適用することで得られる。（Q．E．D．）この証明はVan Dalen［1983］による。

(20) Van Dalen［1983］p.440。

(21) Van Dalen［1983］pp.473-4　参照。

(22) われわれの戦略は，$f(y,\vec{x})=0$　であるような，最初のyを見出すまでyのすべての値を連続的にチェックすることである。もし$f(y,\vec{x})=0$ならば，$\psi(y,\vec{x})$が0を生み出し，もし$f(y,\vec{x})\neq 0$ならば，ステップ数を数えながら次のyへと動くような関数$\psi$が求められている。この関数$\psi$がインデックスeを持つとしよう。われわれは，$\psi_1(e,y,\vec{x})=0$　および　$\psi_2(e,y,\vec{x})=\psi(y+1,\vec{x})+1=|e|(y+1,\vec{x})+1$　であるような，インデックスbとcを各々持つ補助関数$\psi_1,\psi_2$を導入する。もし$f(y,\vec{x})=0$ならば，$\psi_1$を考え，もし$f(y,\vec{x})\neq 0$ならば，$\psi_2$を考える。よって，われわれは，R４によって，新しい関数$\chi_0$：

$$\chi_0(e,y,\vec{x})=\begin{cases} b, & \text{もし} f(y,\vec{x})=0 \text{のとき；} \\ c, & \text{もし} f(y,\vec{x})\neq 0 \text{のとき，} \end{cases}$$

を導入し，$\chi(e, y, \vec{x}) = \{\chi_0(e, y, \vec{x})\}(e, y, \vec{x})$　とおく。帰納定理によって，

$\chi(e_0, y, \vec{x}) = \{e_0\}(y, \vec{x})$

であるようなインデックス$e_0$が与えられる。もしそれが存在すれば，$\{e_0\}(0, \vec{x})$は望ましい値を生み出す。すなわち，$e_0$がわれわれが目指していた関数$\psi$のインデックスである。というのは，もし$f(y, \vec{x}) \neq 0$ならば，$\chi(e_0, y, \vec{x}) = \{c\}(e_0, y, \vec{x}) = \psi_z(e_0, y, \vec{x}) = \psi(y+1, \vec{x}) + 1$，もし$f(y, \vec{x}) = 0$ならば，$\chi(e_0, y, \vec{x}) = \{b\}(e_0, y, \vec{x}) = 0$。よって，$y_0$が$f(y, \vec{x}) = 0$となる最初の値yと仮定すると，$\psi(0, \vec{x}) = \psi(1, \vec{x}) + 1 = \psi(2, \vec{x}) + 2 = \cdots = \psi(y_0, \vec{x}) + y_0 = y_0$ (Q. E. D.)。

(23) チャーチの提唱が論文の形で明示的になされたのはChurch [1936] においてである。

(24) Kleene [1952] の§§60, 62参照。

(25) Rogers [1967] 参照。

(26) Gandy [1980] 参照。

(27) 例えば，Gödel-Herbrand, Church, Curry, Turing, Markov, Post, Minski, Shepherdson-Sturgis などによって提案されている。

(28) Gödel [1946] P. 84（ただし，この頁づけはDavis [1965] による。Gödel [1990] では150頁)。尚，ゲーデルはこの論文をDavis [1965] に収録するに際して，引用した部分に対して次のような脚註を与えている：「正確には：整数の関数が算術を含む任意の形式的体系で計算可能であるのは，それが算術で計算可能であるとき，かつそのときにかぎる。ここで，関数fが［体系］Sで計算可能であるのは，fを表現する計算可能な項がSに存在する場合である。」

(29) 例えば，$\neg(x < \mathbf{0})$，$x <_S y \equiv x < y \vee x = y$，$x < y \vee x = y \vee y < x$，$x < y \wedge y < z \supset x < z$　などが証明できる。

(30) Tarski, Mostowski, Robinson [1953] 参照。

(31) Shoenfield [1967] p. 22.

(32) 証明のポイントは適切な形に各述語を書き下すことである：

(a) $\mathrm{Vble}(n) \Leftrightarrow \exists x \leqq n\ (n = \langle 2, x\rangle)$

(b) $\mathrm{Term}(n) \Leftrightarrow [\mathrm{Vble}(n) \vee n = \langle 1\rangle \vee \exists x < n\,(n = \langle 9, x\rangle \wedge \mathrm{Term}(x)) \vee \exists x, y < n\ (n = \langle 11, x, y\rangle \wedge \mathrm{Term}(x) \wedge \mathrm{Term}(y)) \vee \exists x, y < n\,(n = \langle 13, x, y\rangle \wedge \mathrm{Term}(x) \wedge \mathrm{Term}(y))]$

(c) $\mathrm{Form}(n) \Leftrightarrow \exists x, y < n\,(n = \langle 15, x, y\rangle \wedge \mathrm{Term}(x) \wedge \mathrm{Term}(y))$
$\vee \exists x, y < n\,(n = \langle 3, x, y\rangle \wedge \mathrm{Form}(x) \wedge \mathrm{Form}(y))$
$\vee \exists x < n\,(n = \langle 5, x\rangle \wedge \mathrm{Form}(x))$
$\vee \exists x, y < n\,(n = \langle 7, x, y\rangle \wedge \mathrm{Vble}(x) \wedge \mathrm{Form}(y))$

(33) Kleene [1952] §49, Shoenfield [1967] pp. 128-130参照。

(34) Shoenfield [1967] p. 211参照。

(35) Shoenfield [1967] pp. 55-57参照。

(36) Shoenfield [1967] pp. 57-61参照。

(37) Gödel [1931] 参照。

(38) 証明：まず代入関数Subを用いて，$s(m, n) :=\mathrm{Sub}(m, \ulcorner x_0\urcorner, \ulcorner \mathrm{Num}(n)\urcorner)$とおき，$k := \ulcorner \phi(s(x_0, x_0))\urcorner$とする。そのとき，$\psi := \phi(s(\overline{k}, \overline{k}))$によりトリックが完成する。というのは，$s(k, k) = s(\ulcorner \phi(s(x_0, x_0))\urcorner, k) = \ulcorner \phi(s(\overline{k}, \overline{k}))\urcorner$。それゆえ，sの表現可能性定理により，$PA \vdash s(\overline{k}, \overline{k}) = \overline{\ulcorner \phi(s(\overline{k}, \overline{k}))\urcorner}$。よって，$PA \vdash \psi(s(\overline{k}, \overline{k})) \equiv \phi(\overline{\ulcorner \phi(s(\overline{k}, \overline{k}))\urcorner})$。ここで，$\psi$として$\phi(s(\overline{k}, \overline{k}))$を取ると，$PA \vdash \psi \equiv \phi(\overline{\ulcorner \psi \urcorner})$。(Q. E. D.)

(39) Rosser [1936] 参照。

(40) 証明：真理定義$\tau$が存在すると仮定する。すると，嘘つきのパラドクスを定式化できる。すなわち，不動点定理により，$T \vdash \lambda \equiv \neg \tau(\overline{\ulcorner \lambda \urcorner})$であるような文$\lambda$が存在する。$\tau$は真理定義であるから，$T \vdash \lambda \equiv \tau(\overline{\ulcorner \lambda \urcorner})$，よって$T \vdash \lambda \equiv \neg \lambda$。これはTの無矛盾性に反する。

(41) 次のような理論が決定不可能であることが知られている（Van Dalen [1983] p. 465参照)。①ペアノの算術，②環の理論（Tarski：1949)，③順序体（J. Robinson：1949)，④束の理論（Tarski：1949)，⑤二項述語の理論（Kalmár：1936)，⑥対称的二項述語の理論（Church, Quine：1952)，⑦半順序の理論（Tarski：1949)，⑧二つの同値関係の理論（Rogers：1956)，⑨群の理論，⑩半群の理論，⑪整域（integral domain）の理論。

(42) Gabbay [1981] 参照。

(43) 次のような還元クラスの例が研究されている (Van Dalen [1983] p. 466参照)：

$\exists\forall\exists\forall(0,3)$ : Büchi 1962,　　$\forall\exists\forall(0,\infty)$ : Kahr, Moore, Wang 1962,

$\forall^3\exists(0,\infty)$ : Gödel 1933,　　$\exists\forall\exists\forall(\infty,1)$ : Rödding 1969,

$\forall\exists\forall(\infty,1)$ : Kahr 1961,　　$\exists^\infty\forall^2\exists^2\forall^\infty(0,1)$ : Kalmár 1932

$\forall^\infty\exists(0,1)$ : Kalmár, Suranyi 1950　　$\forall\exists\forall^\infty(0,1)$ : Denton 1963,

$\forall\exists\forall\exists^\infty(0,1)$ : Gurevich 1966.

(44) 詳細については Lewis [1979] 参照。

(45) Tarski [1951] 参照。

(46) 決定可能な理論の研究については，Rabin [1977] 参照。

(47) Matijasevič [1973] 参照。

## 参考文献


Carnap, R. : 1937, *The Logical Syntax of Language*, Routledge and Kegan Paul.

Church, A. : 1936, "An unsolvable problem of elementary number theory", *The American Journal of Mathematics*, vol. 58 (1936), pp. 345-365, reprinted in Davis [1965].

Davis, M. (ed.) : 1965, *The Undecidables*, Raven Press.

Fraenkel, A. A., Bar-Hillel, Y., and Van Dalen, D. : 1973, *Foundations of Set Theory*, North-Holland.

Gabbay, D. M. : 1981, *Semantical Investigations in Heyting's Intuitionistic Logic*, D. Reidel.

Gandy, R. : 1980, "Church's thesis and principles for mechanisms", in J. Barwise, H. J. Keisler and K. Kunen (eds.), *Kleene Symposium*, North-Holland, 1980, pp. 123-148.

Gödel, K. : 1931, "Ueber formal unentscheidbare Sätze der *Principia mathematica* und verwandter Systeme I", *Monatshefte für Mathematik und Physik* vol. 38 (1931), pp. 173-198, also in Davis [1965] pp. 4-38, and in Gödel [1986] pp. 144-195.

Gödel, K. : 1946, "Remarks before the Princeton bicentennial conference on problems in mathematics" in Davis [1965] pp. 84-88, and in Gödel [1990] pp. 150-153.

Gödel, K. : 1986, *Collected Works* vol. I, Oxford U. P.

Gödel, K. : 1990, *Collected Works* vol. II, Oxford U. P.

Kleene, S. C. : 1952, *Introduction to Metamathematics*, North-Holland.

Lewis, H. R. : 1979, *Unsolvable Classes of Quantificational Formulas*, Addison-Wesley.

Matijasevič, Y. : 1973, "Hilbert's tenth problem", in P. Suppes, L. Henkin et al. (eds.), *Logic, Methodology and Philosophy of Science*, North-Holland, pp. 89-110.

Rabin, M. O. : 1977, "Decidable Theories", in J. Barwise (ed.), *Handbook of Mathematical Logic*, North-Holland, 1977.

Rogers, H. : 1967, *Theory of Recursive Functions and Effective Computability*, McGraw-Hill.

Rosser, J. B. : 1936, "Extensions of some theorems of Gödel and Church", *The Journal of Symbolic Logic* vol. 1 (1936), pp. 87-91, in Davis [1965].

Shoenfield, J. R. : 1967, *Mathematical Logic*, Addison-Wesley.

Tarski, A. : 1951, *A Decision Method for Elementary Algebra and Geometry*, 2nd rev. edn., Berkley.

Tarski, A., Mostowski, A., and Robinson, R. M. : 1953, *Undecidable Theories*, North-Holland.

Turing, A. : 1936, "On computable numbers, with an application to the Entscheidungsproblem", *Proceedings of London Mathematical Society*, ser. 2, vol. 42 (1936-7), pp. 230-265.

Van Dalen, D. : 1983, "Algorithms and Decision Problems : A Crash Course in Recursion Theory", in G. Gabbay and F. Guenthner (eds.), *Handbook of Philosophical Logic*, vol. I, Reidel, 1983, pp. 409-478.