SM-計算可能性と SM-計算可能関数のクラス

高 橋 信 行

もくじ

§ 1. はじめに
§ 2. 記号と準備
2.1 有限オートマトンと正規集合
2.2 計算可能性
§ 3. SM-計算可能性
§ 4. むすび

§1. はじめに

有限オートマトンの言語受理能力については，McCulloch & Pitts〔4〕以来，様々な研究がなされ，今日では Hopcroft & Ullman〔2〕，Salomaa〔7〕のような成書の形で広くその研究成果が紹介されている。本稿は，そのような広く認められた有限オートマトンの言語受理能力に対して，更に別の機能を定め，そうして得られた諸性質について考察を試みたものであって，主題であるところのSM-計算可能性は，最終状態集合がfだけから成るような有限オートマトンに出力記号のアルファベットΔを加え，6文字組で定義される列機械による計算可能性である。

計算可能性に関しては，1930年代後半より K. Gödel, A. Church, E. L. Post, A. M. Turing, S. C. Kleene らによる研究があり，今日ではこれらの計算可能性の概念は，『計算の理論』とか "Theory of Recursive Functions" 等の名前の付された数多くの文献によって紹介されている。また1960年代の N. Chomsky, S. Ginsburg らによる形式言語理論の分野においても，典型的な4つの言語のクラスに対応して，それぞれ Turing 機械，線形有界オートマトン，プッシュダウン・オートマトンおよび有限オートマトンが提案され，見事なハイアラーキーを構成している。更に今日では，各種の複雑な能力を持ち合わせた計算機械が考案され，受理される言語のクラスのハイアラーキーも一層，多様性に富んだものになっている。しかしながら本稿の列機械は，これらの機械に比して，極めて単純な能力しか持ち合わせておらず，従っておのずから計算される関数のクラスの偏狭性が予想される。本稿の列機械による計算可能性は，有限オートマトンの受理能力を保存し，更に出力能力も付加した列機械（Sequential

Machine）による計算可能性であって，通常の列機械との相違点はその停止性の定義にある。本稿の出力能力を持った有限オートマトンは，唯一の最終状態fに対しては，どんな入力文字についても次の動作は定義されず，従って機械が停止するとすれば，常に状態fで停止することになる。

さて，Ritchie〔6〕は Turing 機械の計算能力に制限を加える形で有限オートマトンを定義し，その有限オートマトンによって計算される関数のクラスの閉包性を論じている。すなわち

（i） 後者関数，定数関数，恒等関数および加法は Ritchie 式の有限オートマトンで計算されること，

（ii） Ritchie 式の有限オートマトンによって計算される関数のクラスが Composition, Explicit transformation の下で閉じていること，および

(iii) すべての Ritchie 式の有限オートマトンで計算可能な関数は，一定の Linear function $K \cdot max\,(x_1, \cdots, x_n, 1)$ で評価されることを示し

(iv) このクラスが掛け算さえも含まぬことをその導入部分で述べている。

本稿 § 3 では

(1) 上記の（i）（ii）(iii) の性質が，本稿の列機械によっても保存されることを示し，(iii) に関連して

(2) 任意の linear bounded function は SM-計算可能であるか，という問いに対する否定的な部分的解答を与え，

(3) 引き算は，部分的に SM-計算可能であることを示し

(4) 計算結果のテープ上に占める大きさが，その機械の状態数に依存して評価できることを用いて (iv) の証明を与え，

(5) 集合 A が正規集合であるとき，A の特徴関数は SM-計算可能

であること等を論じている。尚，Ritchie 式の有限オートマトンは，テープ上右から左へ移動してゆくが，本稿の機械は通常通り左から右へ移動するよう改められている。

§2. 記号と準備

本節では，次節で必要となる基本的な概念とそのための記法を導入し，併せて計算可能性の理論の歴史的発展経緯を概観する。

2.1 有限オートマトンと正規集合

空でない有限集合 Σ を任意に固定し，これをアルファベットと呼ぶ。Σ の元の有限列を語（word）という。すなわち x が語であれば，x は $a_0a_1\cdots a_{n-1}$ と書けて a_0，a_1，…，a_{n-1} はすべて Σ の元である。このとき n を x の長さといい $|x|=n$ とあらわす。空の語，すなわち長さ0の語も考えることにして，これを ε であらわす。ε も含めたすべての語の集合を Σ^* であらわし，ε を除いた語の全体を Σ^+ であらわす。Σ^* の部分集合を言語（language）という。すなわち L が言語であるとは，L が次のように書けることである。

$$L=\{x\in\Sigma^*\mid P(x)\}$$

ここに $P(x)$ は，x についての述語である。P が恒真のとき $L=\Sigma^*$ であり，P が恒偽のとき $L=\phi$ である。

x，$y\in\Sigma^*$ で $x=x_0x_1\cdots x_{n-1}$，$y=y_0y_1\cdots y_{m-1}$ であったとする。このとき，x と y を連接してできる有限列 $x_0x_1\cdots x_{n-1}y_0y_1\cdots y_{m-1}$ を xy であらわすことにする。明らかに $xy\in\Sigma^*$ であって $|xy|=n+m$ である。

A，B を言語とする。以下に言語についての演算を定める。

$$AB=\{xy \mid x\in A \ \& \ y\in B\}$$

これを連接（concatenation）という。

$$A\cup B=\{x \mid x\in A \ \vee \ x\in B\}$$

これを和（union）という。

冪乗を次のように定める。

$$\begin{cases} A^0=\{\varepsilon\} \\ A^{n+1}=A^nA \end{cases}$$

$$A^*=\bigcup_{n\in\omega} A^n$$

これを Kleene 閉包（Kleene closure）という。

定義 2.1 $A\subseteq\Sigma^*$ とする。A が Σ^* のいくつかの有限部分集合から連接，和，Kleene 閉包の 3 つの演算を有限回，適用して得られるとき，A を正規集合（regular set）という。

前述のように言語とは，ある特徴 $P(x)$ を持った語 x の集まりであった。かかる内延的な言語の特徴付けに対して，以下では外延的なアプローチを試みる。すなわち任意に与えられた語，$x\in\Sigma^*$ に対して，x は自身の元であるのか（YES），否か（NO）を判定し得るようなシステムを構築するのである。そのようなシステムが "YES" とした語 x だけを集めれば，Σ^* の部分集合，すなわち言語が定められよう。

定義 2.2 次に定めるシステムを有限オートマトン（finite automaton）という。

$M=(K, \Sigma, \delta, q_0, F)$ ここに

（i） K は状態の空でない有限集合，

（ii） Σ はアルファベットである。

（iii） δ は $K\times\Sigma$ から K への関数，

（iv） $q_0\in K$ を初期状態といい，

（v） $F\subseteq K$ を最終状態集合という。

δ は $K\times\Sigma$ から K への関数であるが，次のようにして $K\times\Sigma^*$ から K への関数に拡張できる。

$$\begin{cases}\hat{\delta}(q,\ \varepsilon)=q\\ \text{すべての } x\in\Sigma^*,\ a\in\Sigma \text{ に対して}\\ \hat{\delta}(q,\ xa)=\delta(\hat{\delta}(q,\ x),\ a)\end{cases}$$

このようにしてできた $\hat{\delta}$ をもって，新しく δ とする。

語 x は，$\delta(q_0,\ x)=p$ が F に属すとき，有限オートマトン M によって受理されるという。M によって受理される語の全体を $T(M)$ であらわす。すなわち

$$T(M)=\{x\in\Sigma^*\mid\delta(q_0,\ x)\in F\}$$

有限オートマトンによって受理される語の集合を有限状態言語（f. s. l）という。

定義2.2のシステムは決定性であった。すなわち，次の時刻の状態が唯一つ決まるのであるが，これを次のように非決定性に拡張してみよう。

「δ は $K\times\Sigma^*$ から 2^K への関数とする。」

このようにして，有限オートマトンの次の時刻にとり得る状態は K の部分集合として与えられる。従って，M は0個以上の状態の中から，任意の1つの状態をとって次期状態とすることができ，$x\in\Sigma^*$ 上を M が走ったとき初期状態 q_0 を根とする有限木が構築されることになる。$\delta(q_0,\ x)$ は，K の部分集合であるから非決定性の有限オートマトン M によって受理される語の集合は，次のように定義される。

$$T(M)=\{x\in\Sigma^*\mid\delta(q_0,\ x)\cap F\neq\phi\}$$

定理 2.1 (Rabin & Scott〔5〕)

L が非決定性の有限オートマトンによって受理される。*iff* L は決定性の有限オートマトンによって受理される。

従って，有限状態言語を定める上ではシステムは，決定性でも非決定性

でも，どちらでも良く，これらはまったく同等な能力を持ち合わせていると言える。従って以後，有限状態言語を定めるシステムを考えるときは，特に決定性か非決定性かは必要がない限り言明しないことにする。

定理 2.2（Kleene〔3〕）

L が有限状態言語 iff L は正規集合

すなわち，有限オートマトンによって受理される言語は，Σ^* の有限部分集合から演算の順序を指定するかっこと和，連接，Kleene 閉包とを有限回，用いて組み立てられるある式で表現され，逆にそのような式は有限オートマトンによって受理されるような或る言語を表現していると言えるのである。この事実は §3，定理 3.8 の動機付けを与えている。

2.2 計算可能性

「計算」ということが，数学史上，初めて厳密な分析の対象になったのは，1930年代になってからである。応用科学の立場から見た「計算」は，数値計算を指す場合が多いが，数学，特に情報科学の基礎理論において，考察の対象となる「計算」は数値計算や代数計算も含むが，他にも命題計算や述語計算，集合計算も含んでいる(1)。現に今日のプログラミング言語，例えば PASCAL などでは命題計算や集合計算も 1 つの手続きとして記述できるような形になっている。すなわち我々が考察の対象とする「計算」とは，プログラムの命令の実行なども含めた有限回の具現的な手続きのことである。

今日，計算機科学の発達によって，コンピュータの様々な分野への貢献が著しいが，このような有限回の具現的な手続きを実行する機械に人間は理論上，何を行なわせることができて，又それはどこまで可能であるのか，つまりコンピュータの理論的可能性の限界は果たして存在するのか，

もし存在するとすればそれはどのような議論で論断されるのであろうか。我々が，ア・プリオリに持っているこのような漠然とした「計算可能性」の概念に対する数学的定式化は，既に1930年代後半に数学基礎論の世界で得られており，今日での帰納的関数論あるいは広義に計算の理論は，単に情報科学基礎論としての教養に留まらず，プログラムの検証やプログラムそのものに対する意味論的基礎付けを行なったり，形式言語理論における決定問題の解を与える上での強力な武器となっているのである。もちろん，当時は今日のような電子計算機（electronic computer）は出現していなかったが，1930年代から特に1950年代の帰納的関数論の研究成果には目を見はるものがあり，それらの一部は今日での計算機科学の強固な理論的基礎付けを形成しているのである。

それでは，当時コンピュータという金物の無かった時代になぜ，「計算」あるいは「有限回の具現的手続き」について問題にしなければならなかったのであろうか。この問いに答えるためには，前世紀末から今世紀初頭にかけての，数学界の状況を探ってみる必要がありそうである。

1871年に始まる G. Cantor によるフーリエ級数の単一性の研究は，1872年の実数論の展開へと引きつがれ，彼の集合論研究の布石となった(2)。Cantor は1874年から1898年にかけての11編の論文で**集合論**，特に点集合論を提起している。その前半では，実数全体が非可算集合であることや，座標空間の直線との対等性等を論じ，更には点集合の幾何学的な構造，すなわち集積点の定義に始まって開集合，閉集合，完全集合，連結集合等の概念を得，位相数学の根幹的な部分を形成した(3)。後半での彼の集合論研究は，より抽象的な集合の研究に移り超限計数，超限順序数の概念を得て，「整列集合の計数についての極めて自然な理論を得た(4)。」しかし連続体問題の出現や超限順序数全体に対する超限順序数を定義しようとするときに起こ

る逆理，その他様々な逆理が，Cantor の集合論で起こることが指摘されて，Cantor をはじめ，この新しい創造的な理論に好意的であった人々に衝撃を与えた。しかしながら Cantor の集合論は，Dedekind の実数論とも結び付いて，解析学の基礎付けを行なう上での有用な武器となることが明らかになっていった。このような素朴集合論の中での逆理の発見と，一方それが極めて有用な理論であることの認識という，二律背反的状況が，20世紀初頭の数学界の状況であった。かかる**数学の危機**を乗り越えるために生まれたのが数学基礎論であって，これには Russell の論理主義，Brouwer の直観主義および Hilbert の形式主義の 3 つの流れがあったが，論理主義と直観主義は数学の内部では自然淘汰的にその効力を失ない（直観主義は，最近になって再び脚光を浴びてきている），Hilbert の形式主義が，逆理の排除を指向する公理的集合論に組み入れられていった。このような公理的接近方法は，あらゆる数学に有効であり，その真髄は記号の無内容性にあるとする Hilbert の立場を特に，自然数論に具体化するとすれば，まず有限個の記号を準備して，それらから論理式を構成し，そのうちのいくつかを自然数論の公理として選び，更に推論規則を与えて 1 つの理論が，構築されることになる。このような，有限個の記号を土台として展開する公理論の立場を**有限主義**という[5]。

Hilbert は有限主義の立場から，すべての数学を公理化して，各々の公理系の無矛盾性を証明するという壮大な計画を立てた。これがヒルベルト計画といわれるものであるが，この計画は1931年の K. Gödel による不完全性定理の発表で，無残にも打ち破られることになるのである。この不完全性定理を端的に述べると，次のようである。

不完全性定理：公理的自然数論 Z が無矛盾であるとき，この公理系における閉論理式 P で，P とその否定 ¬P が，共に有限的に証明され得ないも

のが存在する。

すなわち，ヒルベルト計画に則って形式化した自然数論が，もし無矛盾であれば，それは不完全であるというのである。この定理の証明の中でGödel は「有限回の手続き」で得られもの，すなわち論理式とか証明とかをゲーデル数化して，次のような述語Aを考えた。

「A(p, q)：pは，ある論理式のゲーデル数。

〔qは，ゲーデル数がpである論理式の，証明のゲーデル数[6]。」

Gödel は，J. Herbrand の示唆により一般帰納的という概念を導入して「具現性」を定式化し，A (p, q) は一般帰納的であるが，$\exists q$ A(p, q) は一般帰納的でないことを証明したのである。すなわち，ゲーデル数pを持つ論理式があって，その論理式の証明を与えるゲーデル数qが，存在するか，どうかということは，有限回の具現的な手続きでは得られないということである。更に直観的に述べれば，この事実は有限主義で得られた結果を算術化してみたら，有限主義内部に，有限主義的方法で手の施しようの無いものがあることがわかった，ということにもなろう。このようにして，「有限回の具現的な手続き」に対する研究の動機付けと方法論が与えられ，1933年のA. Church による λ-定義可能性，1936年の A. M. Turing の Turing 機械による計算可能性，および Turing 機械とは独立で，しかもほとんど同様な E. L. Post の機械によって，本来直観的な概念である「計算可能性」が，数学的研究対象として精確に定義されたのであった。

一方，このような機械による計算可能性に対して，J. Herbrand, K. Gödel によって導入された帰納的関数の概念は，数論的関数のあるクラスを定めるためのものであったが，A. Church は λ-定義可能性を用いて，この帰納的関数のクラスを論じ，Turing, Kleene らによる具現性の様々な定

式化が，すべて一定の概念（＝帰納性）に到達することをふまえて

Church's Thesis：帰納的関数のクラスこそが，我々が，ア・プリオリに持つ「計算可能」関数のクラスである，

ことを提唱した。同じ年（1936年）S. C. Kleene は，部分帰納的関数についても，Church's Thesis が提唱されることを示し，今日ではこの拡張された場合も含めて，Church's Thesis と呼ぶようになっている(7)。

Church's Thesis は，あくまで提唱であって，人智の豊かさ，例えばコンピュータの計算能力を数学の degree で推し測ろうとすれば，それは高々，帰納的関数のクラスであろうということである。ここで人智の豊かさとか，コンピュータの計算能力とかいう曖昧な言い方をしたが，Turing 機械で計算可能な関数のクラス＝帰納的関数のクラスという A. M. Turing の結果と Chuch's Thesis とにより，我々の直観的な「計算可能性」たとえば未来のコンピュータの計算能力さえも，数学的に定式化された Turing 計算可能性で置き換えられてしまうことになる。更に，Gödel の不完全性定理により帰納的でない述語の存在が言えているから，その述語 $\exists q\ A(p, q)$ の真偽を決める Turing 機械は，存在しないことになる。換言すれば，$\exists q\ A(p, q)$ によって定められる集合の特徴関数の値を得るようなコンピュータ・プログラムは存在しないということである。これは，コンピュータの理論的能力の限界をはっきり示すものである。数論的関数の中には，このように帰納的でない，すなわち計算可能でないものがあるのである。そのようなアルゴリズムのない問題（非可解問題という）は，その後，続々と発見され帰納的関数の理論は，非可解性の度合いの強弱を調べる方向に進んでいったのである。

しかしながら，1960年代に入ってコンピュータのプログラムを作るという実用的な要請から，アルゴリズムのない問題よりは，アルゴリズムのあ

る問題に目を向ける工学的研究，すなわち計算量の理論の研究が盛んになって来た。そのころからプログラマーの間で，ORの問題の中には，確かに計算可能ではあるけれども膨大な計算ステップや記憶領域を必要として，実際上は手に負えない，やっかいな問題が存在することが知られていた。「確かに，実際に計算可能な問題とは，オーダが n の多項式を越えない程度の問題であり，それ以上の計算量をもつ問題，例えばナップザック問題などは理論的には計算可能ではあるが，実際上は計算不能と考えて良いのである[8]。」このように計算可能性の理論は，計算不能関数の理論も含めて，真に**計算の理論**と呼ぶにふさわしい内容となって発展を続けているのである。

§ 3. SM-計算可能性

定義 3.1 アルファベットとは，空記号Bを常に含む記号の有限集合である。

定義 3.2 アルファベット Σ 上の列機械 SM は，6-tuple $(S, \Sigma, \Delta, s_0, \delta, f)$ で表わされる。ここに

（i） S は状態の空でない有限集合.

（ii） Σ は入力記号のアルファベット.

（iii） Δ は出力記号のアルファベットである。

（iv） $s_0 \epsilon S$ は初期状態と呼ばれる。

（v） δ は $(S-\{f\}) \times \Sigma$ から $S \times \Delta$ への写像で yield mapping と呼ばれる。

（vi） $f \epsilon S$ は最終状態である。

一般の列機械はこのように定義されないが，本稿では上記の機械をもっ

て列機械と呼ぶことにする。

定義 3.3 列機械の時点表示 (instantaneous description) は, 3-tuple (t, s, p) であらわされる。ここに

(i) t は, xy なる形をしたテープ, ただし $x\in\Delta^*$, $y\in\Sigma^*$.

(ii) s は現時点での機械の状態.

(iii) p は状態 s で見られているテープの駒の番号である。

テープの個々の駒には, 最左端の駒から順に自然数 1, 2, $\cdots$, $|t|$ が割り当てられており, 与えられたテープに対し制御部が $|t|$ 番目の駒を飛び出しそうになったときは, 必要に応じて空記号Bを付け加えることができるものとする。

定義 3.4 列機械 SM の時点表示 X, Y について $X \to Y\ (SM)$ と書けて, これが SM の yield operation と呼ばれるのは, 次の条件が満足されるときである。すなわち

適当な自然数 p, f でない状態 s, および $s'\in S$, $a_i\in\Sigma$, $b_i\in\Delta$ が存在して

$\delta(s, a_i) = (s', b_i)$ が成り立ち

$i < p$ のとき $X = (b_1 \cdots b_{i-1}\ a_i\ a_{i+1} \cdots a_p,\ s,\ i)$

$Y = (b_1 \cdots b_{i-1}\ b_i\ a_{i+1} \cdots a_p,\ s',\ i+1)$

$i \geq p$ のとき $X = (b_1 \cdots b_{i-1}\ a_i,\ s,\ i)$

$Y = (b_1 \cdots b_{i-1}\ b_i\ \mathrm{B},\ s',\ i+1)$

定義 3.5 SM の計算 (computation) とは, 次の (i) (ii) を満足するような時点表示の有限列 $X_1 \cdots X_p$ のことである。

(i) すべての $i < p$ に対して $X_i \to X_{i+1}\ (SM)$

(ii) 適当な $t\in\Sigma^+$, $t'\in\Delta^+$ に対して

$X_1 = (t, s_0, 1)$ & $X_p = (t', f, |t'|)$

X_1 を SM の initial configuration といい, X_p を SM の resultant

configuration という。また t を input tape, t' を t の resultant tape といい，$t' = Res(t)$ と書きあらわす。あきらかに $|Res(t)| = p$ である。

定義 3.6 Σ^+ の部分集合 D に対し SM が関数 $\varphi : D \to \Delta^+$ を計算するとは，D の任意の列 t に対して SM の計算 $X_1 \cdots X_p$ が存在して $Res(t) = \varphi(t)$ かつ $p > |t|$ となることである。

本稿の議論は，数論的関数に限定される。1変数の数論的関数を考える上では，アルファベットは記号0，1，Bを含めば十分である。

2つの記号 a, b の vertical concatenation

$$a \circ b \equiv \begin{matrix} a \\ b \end{matrix}$$

を考える。同一記号 a の n 個の vertical concatenation は $\overset{\circ}{a}{}^n$ で表わす。ただし $\overset{\circ}{\mathrm{B}}{}^n$ については B と同一視することがある。集合についても

$$\Sigma \circ \Delta = \{a \circ b \mid a \epsilon \Sigma,\ b \epsilon \Delta\}$$

が，定義され 3^n $(n \geq 1)$ 個の記号から成るアルファベット Σ_n を次のように定める。

(i) $\Sigma_1 = \{0,\ 1,\ \mathrm{B}\}$

(ii) $\Sigma_{n+1} = \Sigma_n \circ \Sigma_1,\ n \geq 1$

これは input tape を並列に読み込ませるためのアルファベットであり，更にオートマトンがテープの左端から右方向に移動することができて，しかも桁上がりもうまく行なわれるようにするため次の表記法を導入する。すなわち任意の自然数 $n \epsilon \omega$ は，適当な $m \geq 1$，$a_i \epsilon \Sigma_1 - \{\mathrm{B}\}$ に対して

$$n = a_1 2^0 + a_2 2^1 + \cdots + a_m 2^{m-1},\ (n \geq 1 \to a_m \neq 0)$$

と書けることから $(n)_2 = a_1 a_2 \cdots a_m$ であらわす。

定義 3.7 テープ t が，自然数の n-tuple $(x_1, \cdots, x_n) \epsilon \omega^n$ の vertical tape であるとは

$$t = \bar{x}_1 \circ \cdots \circ \bar{x}_n \in \Sigma_n^+$$

なる形に書けることである。ここに個々の $i \leq n$ に対して $\bar{x}_i$ は次のような Σ_1 上の列である。

$$\bar{x}_i = (x_i)_2 \, \mathrm{B}^{k_i}$$

ここに $k_i = max\{|(x_1)_2|, \cdots, |(x_n)_2|\} - |(x_i)_2|$.

このとき $|t| = max\{|(x_1)_2|, \cdots, |(x_n)_2|\}$ であることに注意しよう。たとえば，3-tuple (8, 0, 3) の vertical tape は

$$\mathrm{t} = \begin{matrix} 0 & 0 & 0 & 1 \\ 0 & \mathrm{B} & \mathrm{B} & \mathrm{B} \\ 1 & 1 & \mathrm{B} & \mathrm{B} \end{matrix}$$

である。

このような vertical encoding を一般に $\bar{\ }$ であらわし，テープの自然数への decoding が $(x_1, \cdots, x_n)$ であることを $N(t) = (x_1, \cdots, x_n)$ であらわす。また自然数の n-tuple $(x_1, \cdots, x_n) \in \omega^n$ を $\mathfrak{x}^{(n)}$ であらわすこともある。n が明らかなときは単に $\mathfrak{x}$ と書く。

定義 3.8 Σ_n 上の列機械 SM が自然数の n-tuple から自然数への total function f を計算するとは，SM が次の条件 (i) (ii) を満足する，或る関数 $\varphi : D \to \Sigma_1^+$ を計算することである。

(i) $D = \{t \in \Sigma_n^+ \mid$ 適当な $\mathfrak{x}^{(n)} \in \omega^n$ に対して $t = \bar{\mathfrak{x}}^{(n)}\}$

(ii) $N(\varphi(t)) = f(\mathfrak{x}^{(n)}) \ \& \ |\bar{f}(\mathfrak{x}^{(n)})| < |\varphi(t)|$

total function f が，適当な列機械によって計算されるとき，f は SM-計算可能であるといわれる。f が partial function のとき，適当な列機械が存在して $\mathfrak{x} \in Dom\ f$ に対しては停止し，$\mathfrak{x} \notin Dom\ f$ に対しては停止しないとき，f は部分的に SM-計算可能であるといわれる。

SM-計算可能なすべての全域的数論関数の全体を $\mathscr{F}$ であらわす。

以上で必要な準備は整った。まず列機械の計算能力を示唆する1つの結

果を示そう。

定理 3.1 関数 φ を計算する $k \geq 2$ 個の状態を持つ列機械 SM を考える。このとき，すべてのテープ t に対して $t \in Dom\,\varphi$ であれば $|t| < |\varphi(t)| \leq k + |t|$ である。

〔**証明**〕SM は φ を計算するのであるから，すべての $t \in Dom\,\varphi$ に対し，最終状態 f で $|t|$ 以下の自然数が割り当てられた駒を見ることはない。もしそうであるとすると SM による φ の計算が定義されない（*cf.* 定義 3.6)。よって $\varphi(t)$ の長さは $|t|$ よりもほんとうに大である。

更に列機械 SM が，右方向に付け加えられた空記号 B を読み続ける間に f でない状態に 2 度落ち入れば，同じ B の列に対し機械はループに入って計算は停止しない。よって $|t|$ 番目の駒を読み込んだ後，高々 k 回のステップの後に機械は最終状態 f になって φ の計算が定義される。よって $|\varphi(t)| \leq k + |t|$ である。 ☒

このように計算結果 $Res\,(t)$ のテープ上に占める大きさに上限が存在することは，SM-計算可能な関数を特徴付ける上で本質的な役割を演ずる。

この結果をもとに，クラス $\mathscr{F}$ の種々の性質を考えて行くことにしよう。

定理 3.2 クラス $\mathscr{F}$ は，つぎの（Ⅰ），（Ⅱ），（Ⅲ），（Ⅳ）のリストをもとに（Ⅴ）Composition，（Ⅵ）Explicit transformation の各演算の下で閉じている。

（Ⅰ） 後者関数

（Ⅱ） 定数関数

（Ⅲ） 恒等関数

（Ⅳ） 加　法

証明は，個々の演算に対する補題をかかげて完遂される。

(Ⅰ) **後者関数** (The Successor Function)

$$S(x) \equiv x + 1$$

補題 3.1 後者関数は SM-計算可能である。

〔**証明**〕 $S(x)$ は次なる列機械 SM によって計算される。

$SM = (\{s_0, s_1, s_2, f\}, \Sigma_1, \Sigma_1, s_0, \delta, f)$

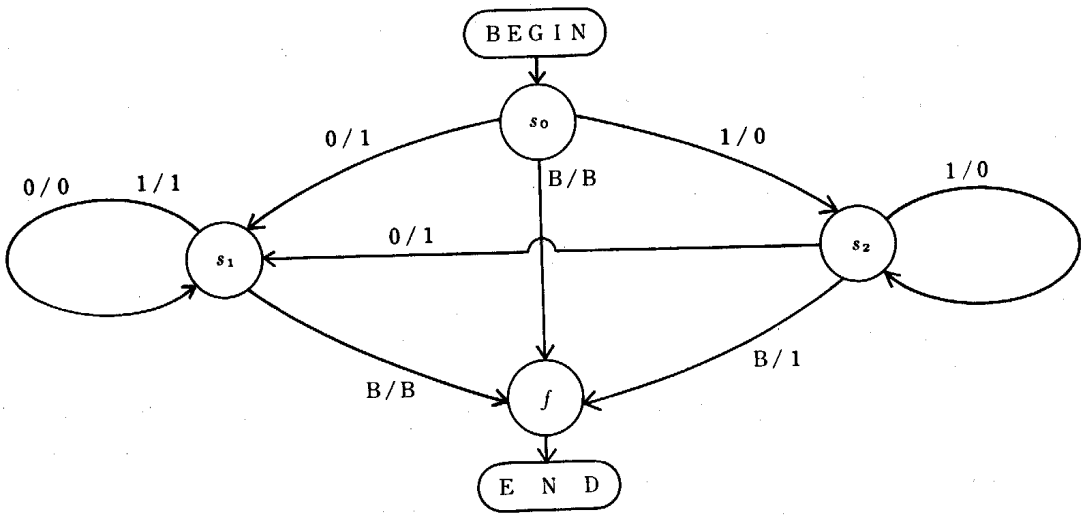
ここに

$\delta(s_0, 0) = (s_1, 1)$ $\delta(s_0, 1) = (s_2, 0)$ $\delta(s_0, \mathrm{B}) = (f, \mathrm{B})$

$\delta(s_1, 0) = (s_1, 0)$ $\delta(s_1, 1) = (s_1, 1)$ $\delta(s_1, \mathrm{B}) = (f, \mathrm{B})$

$\delta(s_2, 0) = (s_1, 1)$ $\delta(s_2, 1) = (s_2, 0)$ $\delta(s_2, \mathrm{B}) = (f, 1)$

実際に SM が関数 S を計算することは次の Flow Graph から明らかである。



□

(Ⅱ) **定数関数** (The Constant Function)

$$C_q^n(x_1, \cdots, x_n) \equiv q$$

補題 3.2 定数関数は SM-計算可能である。

〔**証明**〕 $(q)_2 = a_1 a_2 \cdots a_m$ に対して

$SM = (\{s_i \mid 0 \leq i \leq m\} \cup \{f\}, \Sigma_n, \Sigma_1, s_0, \delta, f)$

を考えればよい。ただし

$\delta : \{s_i \mid 0 \leq i \leq m\} \times \Sigma_n \rightarrow (\{s_i \mid 0 \leq i \leq m\} \cup \{f\}) \times \Sigma_1$ は次のように定義されている。

（i） すべての $\alpha \in \Sigma_n$ に対して

$\delta(s_i, \alpha) = (s_{i+1}, a_{i+1})$; $i = 0, 1, \cdots, m\text{-}1$

（ii） すべての $\beta \in \Sigma_n - \{\overset{\circ}{\mathrm{B}}{}^n\}$ について

$\delta(s_m, \beta) = (s_m, \mathrm{B})$

（iii） $\delta(s_m, \overset{\circ}{\mathrm{B}}{}^n) = (f, \mathrm{B})$

（Ⅲ） **恒等関数** (The Identity Function)

$U_i^n(x_1, \cdots, x_n) \equiv x_i$; $1 \leq i \leq n$

補題 3.3 関数U_i^nは SM-計算可能である。

〔**証明**〕 一般性を失なうことなく input n-tuple $(x_1, \cdots, x_n)$ の vertical tape は，次のように展開されているとすることができる。

$$t = \begin{matrix} \overline{x_1} \\ \vdots \\ \overline{x_i} \\ \vdots \\ \overline{x_n} \end{matrix} \begin{matrix} a_{11} \cdots a_{1j} \cdots a_{1m} \\ \vdots \quad \vdots \quad \vdots \\ a_{i1} \cdots a_{ij} \cdots a_{im} \\ \vdots \quad \vdots \quad \vdots \\ a_{n1} \cdots a_{nj} \cdots a_{nm} \end{matrix}$$

ただし $m = max\{|(x_1)_2|, \cdots, |(x_n)_2|\}$ で，すべての $i \leq n$, $j \leq m$ に対して $a_{ij} \in \Sigma_1$ である。

このとき，関数U_i^n $(x_1, \cdots, x_n)$ は次なる SM によって計算される。

$SM = (\{s_0, f\}, \Sigma_n, \Sigma_1, s_0, \delta, f)$

ただし，$j = 1, \cdots, m$ に対して

（i） $\beta \in \Sigma_n - \{a_{1j} \circ \cdots \circ a_{nj}\}$ であれば

$\delta(s_0, \beta) = (f, \mathrm{B})$

（ii） $\delta(s_0, a_{1j} \circ \cdots \circ a_{ij} \circ \cdots \circ a_{nj}) = (s_0, a_{ij})$ ☒

(Ⅳ) **加法** (Addition)

$$\begin{cases}\varphi(x_1, x_2) \equiv x_1 + x_2 \\ \varphi(x_1, \cdots, x_{n-1}, x_n) \equiv \varphi(\varphi(x_1, \cdots, x_{n-1}), x_n)\ ; n \geq 3\end{cases}$$

補題 3.4 足し算は SM-計算可能である。

〔**証明**〕 補題 3.5 により 2 変数の足し算について言えば，十分である。

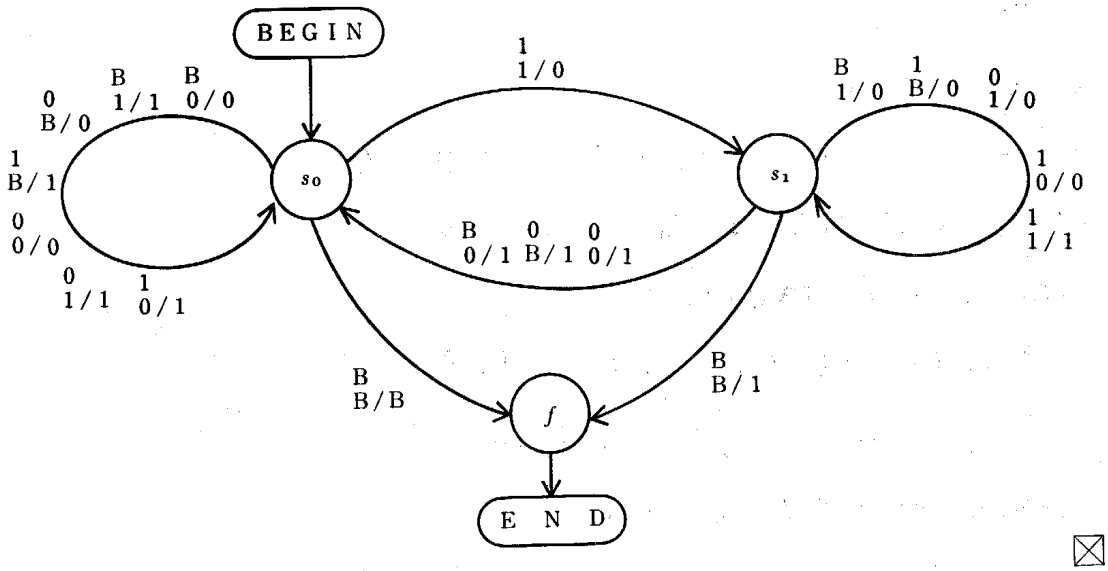
次なる SM を考えれば良い。

$$SM = (\{s_0, s_1, f\}, \Sigma_2, \Sigma_1, s_0, \delta, f)$$

ただし δ は次のように定められている。

δ	B	B	B	0	0	0	1	1	1
	B	0	1	B	0	1	B	0	1
s_0	fB	$s_0$0	$s_0$1	$s_0$0	$s_0$0	$s_0$1	$s_0$1	$s_0$1	$s_1$0
s_1	f1	$s_0$1	$s_1$0	$s_0$1	$s_0$1	$s_1$0	$s_1$0	$s_1$0	$s_1$1

このとき，SM は Σ_2 上で次に示すように足し算を実行する。



□

これで initial functions のすべてのリストが，クラス $\mathscr{F}$ に属すことが示された。次の 2 つの補題によってクラス $\mathscr{F}$ が composition, explicit transformation のもとで閉じていることを確かめる。

（V） **合成**（Composition）

$$h(x_1, \cdots, x_n) \equiv f(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n))$$

補題 3.5 クラス $\mathscr{F}$ は合成の演算の下で閉じている。

〔**証明**〕 $h(x_1, \cdots, x_n) = f(g(x_1, \cdots, x_n))$ について確立する。f が m 変数関数のときも同様である。

関数 f, g は，それぞれ列機械

$$F = (S, \Sigma_1, \Sigma_1, s_0, \delta, f)$$

$$G = (T, \Sigma_n, \Sigma_1, t_0, \eta, g)$$

によって計算されるものとする。

F, G から Σ_n 上の列機械 SM を次のように構成しよう。

$$SM = (U, \Sigma_n, \Sigma_1, u_0, \xi, h)$$

ここに

$$U = \left\{ \begin{bmatrix} s \\ t \end{bmatrix} \middle| \, s \in S, \ t \in T \right\}$$

$$u_0 = \begin{bmatrix} s_0 \\ t_0 \end{bmatrix}$$

$$h = \begin{bmatrix} f \\ g \end{bmatrix}$$

ξ : $(U - \{h\}) \times \Sigma_n \to U \times \Sigma_1$ は次の通り。

任意の $s \in S$, $t \in T$, $a \in \Sigma_n$, $b \in \Sigma_1$ について

case 1 $s \neq f$ & $t \neq g$ のとき

$\delta(s, b) = (s', c)$, $\eta(t, a) = (t', b)$ に対し

$\xi\left(\begin{bmatrix} s \\ t \end{bmatrix}, a\right) = \left(\begin{bmatrix} s' \\ t' \end{bmatrix}, c\right)$ とおく。

case 2 $s \neq f$ & $t = g$ のとき

$\delta(s, b) = (s', c)$, $\eta(g, a)$: undef に対し

$\xi\left(\begin{bmatrix} s \\ g \end{bmatrix}, \overset{\circ}{\mathrm{B}}{}^{n-1} b\right) = \left(\begin{bmatrix} s' \\ g \end{bmatrix}, c\right)$ とおく。

かかる列機械 SM が，実際に関数 $h(\mathfrak{x}^{(n)}) = f \circ g(\mathfrak{x}^{(n)})$ を計算する

ことを見てみよう。

今，列機械 G, F による関数 g, f の計算

$X_1, \cdots, X_p, Y_1, \cdots, Y_q$

が存在して

すべての $i<p$ に対し $X_i \to X_{i+1}$ (G)

すべての $i<q$ に対し $Y_i \to Y_{i+1}$ (F)

で，$X_1 = (\bar{\mathfrak{x}}, t_0, 1)$, $X_p = (Res\ (\bar{\mathfrak{x}}), g, p)$

$Y_1 = (\overline{g}\,(\mathfrak{x}), s_0, 1)$, $Y_q = (Res\ (\overline{g}\,(\mathfrak{x})), f, q)$

であったとする。

ただし

$N\ (Res\ (\bar{\mathfrak{x}})) = g\,(\mathfrak{x})$, $|\overline{g}\,(\mathfrak{x})| < p$

$N\ (Res\ (\overline{g}\,(\mathfrak{x}))) = f \circ g\,(\mathfrak{x})$, $|\overline{f \circ g}\,(\mathfrak{x})| < q$ である。

G, F の計算の長さについては，$q<p$ ということはあり得ないから次の2つの場合を考えれば良い。

case 1 $p = q$ のとき

$i<p$ を任意に固定して

$X_i \to X_{i+1}$ (G), $Y_i \to Y_{i+1}$ (F)

を考えると yield operation の定義により，適当な $s \in S - \{f\}$, $t \in T - \{g\}$, $s' \in S$, $t' \in T$, $a \in \Sigma_n$, $b, c \in \Sigma_1$ が存在して

$\eta\,(t, a) = (t', b)$

$\delta\,(s, b) = (s', c)$

で，$X_i = (\beta a \alpha, t, i)$, $X_{i+1} = (\beta b \alpha, t', i+1)$

$Y_i = (\gamma b \beta', s, i)$, $Y_{i+1} = (\gamma c \beta', s', i+1)$

である。ここに $\alpha \in \Sigma_n^*$, β, β', $\gamma \in \Sigma_1^*$ である。

このとき，ξ の定義 (case 1) から SM の yield operation

$Z_i \to Z_{i+1}$ (SM) が得られて

$$\xi\left(\left[\begin{smallmatrix}s\\t\end{smallmatrix}\right], a\right) = \left(\left[\begin{smallmatrix}s'\\t'\end{smallmatrix}\right], c\right)$$

がなりたち

$$Z_i = (\gamma a \alpha, \left[\begin{smallmatrix}s\\t\end{smallmatrix}\right], i),\ Z_{i+1} = (\gamma c \alpha, \left[\begin{smallmatrix}s'\\t'\end{smallmatrix}\right], i+1)$$

である。

i は任意であったから，すべての $i<p$ に対して，$Z_i \to Z_{i+1}$ (SM) であって

$$Z_1 = (\bar{\mathfrak{x}}, \left[\begin{smallmatrix}s_0\\t_0\end{smallmatrix}\right], 1),\ Z_p = (Res\ (\bar{g}(\mathfrak{x})), \left[\begin{smallmatrix}f\\g\end{smallmatrix}\right], p)$$

であるような SM の計算 $Z_1 \cdots Z_p$ が得られる。

case 2　$q>p$ のとき

$i<p$ に対しては，$Z_i \to Z_{i+1}$ (SM) で

$Z_1 = (\bar{\mathfrak{x}}, \left[\begin{smallmatrix}s_0\\t_0\end{smallmatrix}\right], 1)$，$Z_p = (\gamma\alpha, \left[\begin{smallmatrix}s\\g\end{smallmatrix}\right], p)$ $(s\neq f)$ であるようなSMの時点表示の有限列 $Z_1 \cdots Z_p$ が存在する。

$p\leq i<q$ に対しては，$Y_i \to Y_{i+1}$ (F) が定義されていて

適当な $s\in S-\{f\}$，$s'\in S$，$b,c\in \Sigma_1$ について

$$\delta(s, b) = (s', c)$$ がなりたち

$Y_i = (\gamma b\beta, s, i)$，$Y_{i+1} = (\gamma c\beta, s', i+1)$ である。

このとき ξ の定義 (case 2) により SM の yield operation

$Z_i \to Z_{i+1}$ (SM) が存在して

$$\xi\left(\left[\begin{smallmatrix}s\\g\end{smallmatrix}\right], \mathring{B}^{n-1}b\right) = \left(\left[\begin{smallmatrix}s'\\g\end{smallmatrix}\right], c\right)$$ がなりたち

$Z_i = (\gamma \mathring{B}^{n-1} b\alpha, \left[\begin{smallmatrix}s\\g\end{smallmatrix}\right], i)$，$Z_{i+1} = (\gamma c\alpha, \left[\begin{smallmatrix}s'\\g\end{smallmatrix}\right], i+1)$ である。

従って任意の $i<q$ について，$Z_i \to Z_{i+1}$ (SM) で

$$Z_1 = (\bar{\mathfrak{x}}, \left[\begin{smallmatrix}s_0\\t_0\end{smallmatrix}\right], 1)$$

$$Z_p = (\gamma\alpha, \left[\begin{smallmatrix}s\\g\end{smallmatrix}\right], p)$$

$$Z_q = (Res\,(\bar{g}(\mathfrak{x})),\ [\frac{f}{g}],\ q)$$

であるような SM の計算 $Z_1 \cdots Z_p \cdots Z_q$ が得られる。

これで列機械 SM が関数 $f \circ g$ を計算することが確かめられ，よって補題は成り立つ。 ☒

(VI) Explicit transformation

$h\,(x_1, \cdots, x_n) \equiv f\,(y_1, \cdots, y_m)$

ただし任意の $i \leq m$ に対して $y_i = x_j$ であるような $j \leq n$ が存在するか，または y_i は定数.

補題 3.6 クラス $\mathscr{F}$ は，explicit transformation の下で閉じている。

〔**証明**〕 関数 $g_i\,(x_1, \cdots, x_n)$, $i \leq m$ を次のように定義する。

$$g_i\,(x_1, \cdots, x_n) = \begin{cases} U_j^n\,(x_1, \cdots, x_n)\ ; \text{ある } j \leq n \text{ について } y_i = x_j \text{ のとき} \\ C_q^n\,(x_1, \cdots, x_n)\ ;\ y_i = q \text{ (定数) のとき} \end{cases}$$

このとき，すべての $\mathfrak{x}$ に対して

$h\,(\mathfrak{x}) = f\,(g_1(\mathfrak{x}), \cdots, g_m(\mathfrak{x}))$

と書けて補題は，補題 3.2, 3.3 および 3.5 よりあきらかである。 ☒

これで定理 3.2 の証明がすべて完了した。

次に議論は linear functions との関係について展開される。定理 3.3 はクラス $\mathscr{F}$ のおおよその大きさを示すものである。

定理 3.3 $\mathscr{F}$ はすべての linear functions のクラス $\mathscr{A}$ を含んでいる。

〔**証明**〕 $\mathscr{A}$ の元は，適当な $k_1, \cdots, k_n > 0$ に対して

$f\,(\mathfrak{x}^{(n)}) = k_1 x_1 + \cdots + k_n x_n + k_0$ と書けて，更に

$$= \varphi_{n+1}\,(\varphi_{k1}\,(U_1^n(\mathfrak{x}), \cdots, U_1^n(\mathfrak{x})), \cdots,$$
$$\varphi_{kn}\,(U_n^n(\mathfrak{x}), \cdots, U_n^n(\mathfrak{x})),\ C_{k0}^n(\mathfrak{x}))$$

と書ける。

ここに $\varphi_n(\mathfrak{x}) = x_1 + \cdots + x_n$ である。 (Grzegorczyk [1])

定理は補題 3.2, 3.3, 3.4 および 3.5 より明らかである。 ☒

次の定理は，$\mathscr{F}$ の関数が linear bounded であることを主張している。これは定理 3.1 の結果である。

定理 3.4 関数 $f(\mathfrak{x}^{(n)})$ が $\mathscr{F}$ に属すならば，適当な $K \in \omega$ が存在して，すべての $\mathfrak{x}^{(n)}$ に対して

$$f(\mathfrak{x}^{(n)}) < K \cdot max(\mathfrak{x}^{(n)}, 1)$$

である。

〔**証明**〕 任意の自然数 $x \in \omega$ に対して

$$x < 2^{|\overline{x}|} \leq 2\, max(x, 1)$$

が成り立つ。ここに $|\overline{x}|$ は x の vertical tape の長さである。

n-tuple $\mathfrak{x}^{(n)} \in \omega^{(n)}$ を考えたとき，$\mathfrak{x}$ の vertical tape の長さは $max\{|(x_1)_2|, \cdots, |(x_n)_2|\} = |\overline{max}(\mathfrak{x}, 1)|$ で与えられる。よって関数 $f(\mathfrak{x}^{(n)})$ が SM-計算可能であれば，定義 3.8，定理 3.1 により，適当な $k \geq 2$ が存在して，任意の $\mathfrak{x}$ に対して

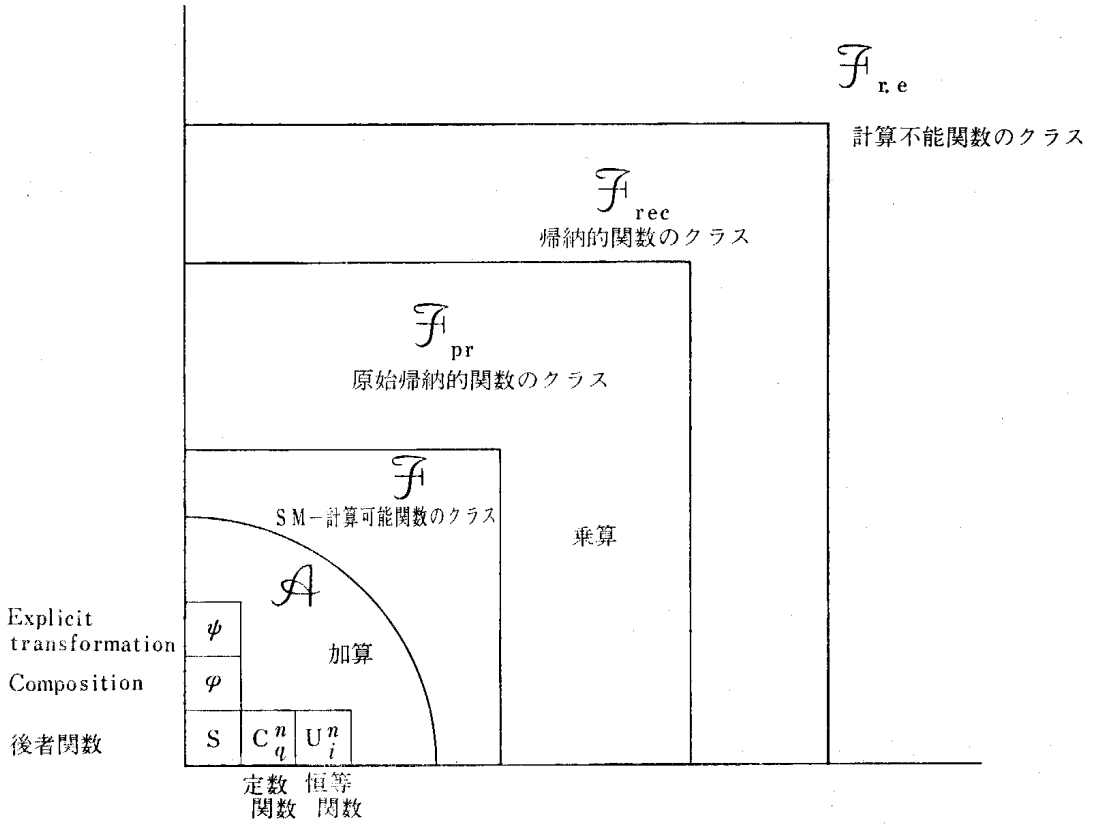
$$|\overline{f}(\mathfrak{x})| < k + |\overline{max}(x, 1)|$$

と書ける。

従って

$$\begin{aligned} f(\mathfrak{x}) < 2^{|\overline{f}(\mathfrak{x})|} &< 2^k \cdot 2^{|\overline{max}(\mathfrak{x}, 1)|} \\ &\leq 2^k \cdot 2\, max(max(\mathfrak{x}, 1), 1) \\ &= K \cdot max(\mathfrak{x}, 1) \end{aligned}$$ ☒

定理 3.2, 3.3 および 3.4 から次図のようなハイアラーキーを得る。



ここで定理 3.4 に関して次のような問題が生ずる。

『ある linear function $g \in \mathcal{A}$ が存在して，任意の f に対して $f < g$ であるならば，$f \in \mathcal{F}$ であるか』

この問題に対する否定的な部分的解答を与えよう。

定理 3.5 $\mathcal{F}$ に属さず，すべての x に対して $f(x) < 2$ であるような関数 f が存在する。

〔**証明**〕 f として集合 $A \subseteq \omega$ の特徴関数 C_A を考える。自然数の部分集合全体の濃度は連続の濃度であるから，個々の A に対応して，その特徴関数 C_A の全体の濃度も連続の濃度である。ところがクラス $\mathcal{F}$ の濃度は，可算濃度であるから，$\mathcal{F}$ に入らない C_A が存在する。 ☒

クラス $\mathcal{F}$ に関して更に 2,3 の結果を挙げよう。

定義 3.9 特殊減算 $\mathbb{F} : S \to \omega$ を次のように定める。ただし $S = \{(x, y) \in \omega^2 \mid x \geq y\}$ である。

$$\Psi\ (x,\ y)\ =\ \begin{cases} x-y & ;\ x \geq y\ \text{のとき} \\ \text{undef} & ;\ \text{otherwise} \end{cases}$$

定理 3.6 特殊減算は，部分的に SM-計算可能である。

〔**証明**〕 次なる列機械 SM は，$(x,\ y) \in S$ に対しては値 $x-y$ を出して停止し，$(x,\ y) \notin S$ に対しては停止しない。

$$SM\ =\ (\ \{s_0,\ s_1,\ s_2,\ f\},\ \Sigma_2,\ \Sigma_1,\ s_0,\ \delta,\ f)$$

δ	B B	B 0	B 1	0 B	0 0	0 1	1 B	1 0	1 1
s_0	f B	s_2 B	s_2 B	s_0 0	s_0 0	s_1 1	s_0 1	s_0 1	s_0 0
s_1	s_2 B	s_2 1	s_2 0	s_1 1	s_1 1	s_1 0	s_0 0	s_0 0	s_1 1
s_2	s_2 B	s_2 B	s_2 B	s_2 B	s_2 B	s_2 B	s_2 B	s_2 B	s_2 B

⊠

定理 3.7 掛け算はクラス $\mathscr{F}$ に含まれない。

〔**証明**〕 k 個の状態を持つ列機械 SM が，つぎの条件（i）（ii）を満足するような関数 $\chi : D \to \Sigma_1^+$ を計算すると仮定する。

（i） $D\ =\ \{t \in \Sigma_2^+ \mid t$ は $(x,\ y) \in \omega^2$ の vertical tape$\}$

（ii） $N\ (\chi\ (t))\ = x \cdot y\ \&\ |\overline{xy}| < |\chi\ (t)|$

このとき定理3.1により，任意の $x,\ y$ に対し

$$\begin{aligned} k &\geq |\chi\ (t)| - max\ (|\bar{x}|,\ |\bar{y}|) \\ &> |\overline{x\ y}| - max\ (|\bar{x}|,\ |\bar{y}|) \\ &\geq |\bar{x}| + |\bar{y}| - 1 - max\ (|\bar{x}|,\ |\bar{y}|) \\ &= min\ (|\bar{x}|,\ |\bar{y}|) - 1 \end{aligned}$$

すなわち，任意の $x,\ y \in \omega$ に対して

$$min\ (|\bar{x}|,\ |\bar{y}|) \leq k$$

これは不合理である。 ⊠

このように列機械で計算可能な関数は，後者関数，定数関数からせいぜ

い足し算，引き算までで掛け算や割り算に関しては，これらは機械の能力を越える関数であるといえる。これらの関数を計算するには，スタック記憶部が必要であろう。

つぎに適当な算術化の下で，正規集合が $\mathscr{F}$ のある関数で特徴付けられることを示そう。ここでは，算術化の一方法としてつぎのような m-adic notation を導入する。

すなわち，与えられたアルファベット Σ から空記号 B をとり除いた空でない有限集合 $\Gamma = \{a_1, \cdots, a_m\}$，$m \geq 2$ を考え，Γ^* から ω への全単射 (bijection) σ を次のように定義する。

$$\sigma(\varepsilon) = 0$$

$$\sigma(a_{ik} a_{ik-1} \cdots a_{i0}) = \sum_{j=0}^{k} i_j \cdot m^j$$

ここに $k \geq 0$ で，すべての $j \leq k$ に対し $1 \leq i_j \leq m$ である。更に $A \subseteq \Gamma^*$ に対し $\sigma(A) = \{i \in \omega \mid \exists\, x \in A\ \sigma(x) = i\}$ とする。

定義3.10 任意の $A \subseteq \Gamma^*$ に対し $\sigma(A) = A'$ とする。$A' \subseteq \omega$ の特徴関数 $C_{A'} : \omega \to \{0, 1\}$ とは，次なる関数である。

$$C_{A'}(x) = \begin{cases} 0\ ;\ x \in A' \text{ のとき} \\ 1\ ;\ x \in \omega - A' \text{のとき} \end{cases}$$

定理 3.8 $A \subseteq \Gamma^*$ が正規集合であれば，$\sigma(A)$ の特徴関数は $\mathscr{F}$ に属する。

〔**証明**〕 定理 2.2 により，A が正規集合ならば，ある有限オートマトン $M = (K, \Gamma, \eta, s_0, F)$ が存在して M は A を受理する。このとき

(i) $D = \{w \in \Sigma^* \mid \text{ある } x \in \omega \text{ に対して } \sigma^{-1}(x) = w\}$

(ii) $N(C_A(w)) = C_{A'}(x)\ \&\ |C_A(w)| > 1$

であるような関数 $C_A : D \to \Sigma_1^+$ を計算する列機械 SM をつぎのように構成する。

$$SM = (S, \Sigma \cup \{\varepsilon\}, \Sigma_1, s_0, \delta, f)$$

ここに

$S = K \cup \{f\}$，ただし f は K にない新しい記号である。

$\Sigma = \Gamma \cup \{\mathrm{B}\}$

yield mapping $\delta : (S-\{f\}) \times (\Sigma \cup \{\varepsilon\}) \to S \times \Sigma_1$

はつぎのように定義する。

任意の $a \in \Gamma$, s, $s' \in K$ について

$\eta(s, a) = s'$ のとき $\delta(s, a) = (s', \mathrm{B})$ とおき

$s \in F$ のとき　　$\delta(s, \mathrm{B}) = (f, 0)$

$s \notin F$ のとき　　$\delta(s, \mathrm{B}) = (f, 1)$ を付け加える。また

$s_0 \in F$ のとき　　$\delta(s_0, \varepsilon) = (f, 0)$

$s_0 \notin F$ のとき　　$\delta(s_0, \varepsilon) = (f, 1)$ と定める。

このように定義された SM が，実際に C_A を計算することを確かめよう。

そのためには，任意の $w \in \Gamma^*$ に対して $\eta(s_0, w)$ が F に属す *iff* $N(C_A(w)) = 0$ を示せばよい。

$\eta(s_0, \varepsilon) \in F$ *iff* $N(C_A(\varepsilon)) = 0$ は明らか。

$w = a_1 \cdots a_n$, $n \geq 1$ のとき

$\eta(s_0, a_1 \cdots a_n) \in F$ *iff*　適当な M の状態関数列 $\eta(s_i, a_{i+1}) = s_{i+1}$ $(i<n)$ が存在して $s_n \in F$.

iff　$\delta(s_i, a_{i+1}) = (s_{i+1}, \mathrm{B})$　$(i<n)$ かつ

$\delta(s_n, \mathrm{B}) = (f, 0)$

iff　すべての $i \leq n+1$ に対して $X_i \to X_{i+1}$ (SM) で

$X_1 = (w, s_0, 1)$, $X_{n+2} = (\mathrm{B}^n 0 \mathrm{B}, f, n+2)$

であるような SM の計算 $X_1 \cdots X_{n+2}$ が存在する。

$$iff \quad N(C_A(w)) = 0$$

すなわち，A が正規集合で $w \in A$ であれば $N(C_A(w)) = 0$ である。よって C_A は SM によって計算されて C_A' は $\mathscr{F}$ に属する。 ☒

§4. むすび

本稿では，言語の受理機械としての有限オートマトンに出力能力を付加し，計算機械としての列機械を新たに定め，この機械による計算可能性（SM-computability）を考察し，次のような結果を得た。すなわち

(1) 後者関数，定数関数，恒等関数および加法は，SM-計算可能である。

(2) SM-計算可能な関数のクラスは，Composition，Explicit transformation の演算の下で閉じている。

(3) SM-計算可能な関数は，linear function $K \cdot max(x, 1)$ で bound される。

(4) SM-計算可能でない linear bounded function が存在する。

(5) 減算は，部分的に SM-計算可能である。

(6) 乗算は，SM-計算可能でない。

(7) 正規集合の特徴関数は SM-計算可能である。

このうち (1)，(2)，(3)は Ritchie〔6〕の議論と本質的に変わるところはないが，(4)，(5)，(7)は文献〔6〕にない新しい結果である。(4)は，(3)の逆に対する否定的結果であり，SM-計算可能な関数のクラスの線形有界関数のクラスに対する偏狭性を示している。(5)は，加法が SM-計算可能であり（(1)），その逆演算も SM-計算可能であることを述べている。ただし，補数に対する加法演算をもって 2 進数の減算が，成就されることか

ら，この結果は当然の理とも言えよう。(6)は文献〔6〕の Introduction に結果的に述べられていたもので，本稿ではこれの証明を与えてみた。乗算は加算の繰り返し演算であるが，本稿の列機械にはこの繰り返し回数を記憶する部位が無く，そのことが乗算を計算不能たらしめている所以であろう。(7)は正規集合と有限オートマトンとの関係から予想される結果であるが，特徴関数の SM-計算可能性は，その集合の正規性を導くに十分でない。更に，SM-計算可能性に別の要因が必要であるが，それが何であるかは未解決である。

このように，本稿の SM-計算可能性は，初等的関数のごく一部の様相を示唆するにとどまっている。それは計算機械のベースに，有限オートマトンという“能力の低い”機械を置いているからであり，有界性や帰納性（＝再帰性）という“より高度な”関数の様相を調べるには，機械の能力が不十分なのである。この点に関して，より広範な関数の計算可能性を調べるとすれば，スタック機能を持った機械による計算可能性を考察したり，線形有界オートマトンに出力能力を付加して length-increasing 性と limited recursion との関係を調べたりすることが，今後の課題として残されていると言えよう。

日頃，御指導頂く，法政大学工学部田中尚夫教授に深謝します。

文　献

〔1〕 Grzegorczyk, A. (1953), “Some Classes of Recursive Functions,” Rozprawy Matematyczne 4. Instytut Mathematyczny Polskey, Akademe Nauk, Warsaw, Poland, 1-45.

〔2〕 Hopcroft, J. E. and Ullman, J. D. (1969) : “Formal Languages and Their Relation to Automata,” Addison-Wesley.

〔3〕 Kleene, S. C. (1956) : “Representation of Events in Nerve Nets and Finite Automata,” Automata Studies, Princeton Univ. Press,

Princeton, New Jersey, 3-42.

〔4〕 McCulloch, W. S. and W. Pitts. (1943), "A Logical Calculus of the Ideas Immanent in Nervous Activity," Bull. Math. Biophys, 5, 115-133.

〔5〕 Rabin, M. O. and Scott, D. (1959), "Finite Automata and Their Decision Problems," IBM. J. Res., 3 : 2, 115-125.

〔6〕 Ritchie, R. W. (1963), "Classes of Predictably Computable Functions," Trans. of the Amer. Math. Soc, 106, 139-173.

〔7〕 Salomaa, A. (1973) : "Formal Languages." Academic Press.

〔8〕 広瀬健編著『数学基礎論の応用』数学セミナー増刊，入門｜現代の数学〔12〕，日本評論社，昭和56年.

脱稿後，〔8〕が出版された。

註

1） 近藤基吉編著『情報科学の展開』　東海大学出版会，昭和48年．121～123頁参照。

2） 井関清志・近藤基吉共著『現代数学――成立と課題』　日本評論社，昭和52年，9頁参照。

3） 同上，10～11頁参照。

4） 同上，11頁より引用。

5） 同上，30頁参照。

6） 同上，275頁より引用。

7） 相沢輝昭著『計算理論の基礎』　文一総合出版，昭和45年，263～264頁参照。

8） 斎藤正彦・広瀬健・森毅編『数学と諸科学』　数学セミナー増刊，シンポジウム数学 2，日本評論社，昭和55年，55頁より引用。