

微分方程式と計算可能性

京都大学総合人間学部基礎科学科 高崎金久 (Kanehisa Takasaki)

概要

離散量に関する計算可能性の概念はすでに今世紀前半の Turing, Church, Kleene らの先駆的研究において基礎固めが終わっている。他方、連続量に関する計算可能性の概念は 50 年代に本格的な研究が始まり、今日までにいくつかの異なる枠組みが提案されている。この論説では、微分方程式への応用を中心に、連続量の計算可能性に対するいくつかの異なる考え方を紹介する。

1 まえおき

今世紀前半、デジタル計算機の出現に先行する形で、「計算とは何か」という問題が Turing, Church, Kleene らによって論じられた。(むしろ、そのような理論的な基礎があればこそ、プログラム内蔵方式のデジタル計算機をつくることが可能になった、というのが正しい。) Turing, Church, Kleene はそれぞれに異なる視点から「計算」というものの数学的な定式化を与えたが、それらは最終的にはすべて等価であることがわかった。計算の理論の基礎はこのようにして築かれて今日に至っている。(ただし、計算の概念そのものや「計算可能性」の問題はすでに過去のもので、今日ではもっぱら「計算量」に関する問題が研究の中心である。)

計算の理論(略して計算論)は数学基礎論と密接な関連を有する。計算というものには、我々が手で行うものから、計算機にさせるものまで、さまざまなレベルがある。それらの一般的な意味を考えるためには、いったんその外に出て対象化・抽象化する必要がある。数学に対してそれを行うのが数学基礎論であるが、数学基礎論(あるいはその基礎になる記号論理学)の方法や結果は計算の理論にも役に立つ。実際、Turing ら先達の研究は前世紀後半から今世紀前半にかけての数学基礎論の進展を背景に(あるいはそれと平行して)展開された。たとえば、計算論のもっとも基本的な事実として「計算不可能な函数の存在」があるが、これは数学基礎論における有名な Gödel の不完全性定理と対応する結果であり、技術的には Cantor 以来の「対角線論法」が証明の核心にある。

ところで、Turing 達が考えた計算論は基本的に離散量のみを扱う。たとえば Kleene がその基礎を築いた帰納的函数論では、もっぱら自然数の集合 $\mathbb{N}$ の上の函数を論じる。自然数以外のもの（たとえば文字列、グラフ、など）は自然数にコード化する。この「自然数へのコード化」という考え方は本質的に Gödel のアイディアなのだが、今日ではデジタル計算機がすべてのデータやプログラムを2進数（ビット列）や16進数（バイトの並び）のコードとして表現していることがよく知られているので、Gödel の時代よりはるかになじみやすいものになっている。（もっとも、Gödel の用いたコード化は素数を用いるもので、もともとあまりわかりやすくはない。）Turing は Turing 機械という機械モデルを、また Church は λ 計算という抽象的な計算モデルをそれぞれ提唱したが、結局これらは等価な計算能力 — 今日のデジタル計算機に ∞ バイトのメモリーを積めば実現される — をもつことがわかる。そして、帰納的函数とはこのような理想化されたデジタル計算機でプログラム可能な函数のことに他ならない。

以下で扱うのは、このような標準的な計算論とは対照的な、連続量（正確には実数）の計算論である。なかでも、微分方程式に関わる問題に焦点を絞る。機械モデルでいえば、これは「アナログ計算機」のことを指すと思えばよい。¹ 函数モデルでいえば、実数の集合 $\mathbb{R}$ 上で帰納的函数論に相当する理論を展開することになる。

もっとも、微分方程式を解くアナログ計算機と、単に連続変数の函数を計算するだけの計算機とでは、かなりの差があるように思われる。実際、後者を使ってできるのは、せいぜい、離散時間の連続力学系のシミュレーション — Runge-Kutta 法などによる近似計算もアルゴリズム自体を眺めればそういうものである — であり、直接に微分方程式を解くことは望めない。微分方程式を正確に解く² ことのできる計算機はもう一段上の階層に属するように思われる。³ このように、連続量の計算論には、離散量の場合とは違って、本質的に異なるものがあるいろいろな有り得ることが想像できる。

¹ かつて von Neumann が「風洞実験装置は大がかりなアナログ計算機のようなものだ」といったそうだが、この場合の「アナログ」という言葉には本来の「実物と相似である」という意味合いも濃厚である。我々はもっぱら今では普通に使われる「アナログ＝連続量」という意味で「アナログ計算機」という言葉を使っている。

² これは数式処理で厳密解を求めるという話ではない。数式処理は明らかに古典的な計算論の範疇内にある。また、ここでは微分方程式の積分可能性はまったく仮定していないから、数式処理による解法はもともと期待できない。

³ そのような計算機があるのか、と問われるかもしれない。アナログ計算機といえども、普通は本当に解きたい問題の類似物 — 実近似 — でしかない。その近似度が自由に制御し難いため、アナログ計算機は結局廃れてしまった。しかしながら、ここでいう「アナログ計算機」は計算の対象 — 典型的には連続時間の連続力学系 — を正確に真似ることのできる理想化された計算機である。そのようなものは対象と区別できないかも知れない。そうならば発想を逆転させて、対象自体が「計算している」と考えたらどうか。たとえば風洞はそれ自体が流体力学的計算をしているし、磁性体はそれ自体が量子力学を解いている、と考えるのである。このように物理的過程などを「計算」と捉える考え方は以前からある。

実際、文献を探ると、そのような互いに異なる枠組み（あるいはそれに基づく結果）として、次のようなものを見つけることができた：

- $\mathbb{R}$ 上の函数の計算論に関する L. Blum, M. Shub, S. Smale の論文 [1]. $\mathbb{R}$ 上の帰納的函数（計算可能函数）や計算量クラスの理論を $\mathbb{N}$ とほぼ平行する形で展開し、その応用として離散時間の連続力学系の問題、特に複素力学系の Julia 集合の計算不可能性を扱っている.
- 微分方程式の解の計算可能性に関する M.B. Pour-El, I. Richards の論文 [2, 3]. Gregorczyk による $\mathbb{R}$ 上の計算可能函数論の枠組の中で、計算可能な函数で構成される微分方程式が計算可能でない解をもつ例を与えている.
- 辻下徹の北大における超準数学の講義の資料 [4]. 超準解析の応用として無限小 Turing 機械の概念を導入し、それに基づいて函数の計算可能性、微分方程式の解の計算可能性、などを論じている.

これらはそれぞれ実数（およびその上の函数）の計算可能性に関する異なる見方に基づいている. 最初 1 つと後の 2 つは扱っている対象が上に述べた意味で異なっている（離散時間系と連続時間系）.

これらの文献を簡単に紹介するのがこの論説の目的である. 内容的にはこれらの文献や関連文献からの切り抜きを集めた「スクラップブック」に過ぎないが、何か新しいことを始めるためにはこのようなスクラップブック作りが欠かせないのである.

2 $\mathbb{N}$ 上の計算論

この節の内容は計算論のどのような教科書にも出ている [5].

2.1 計算可能函数

ここで考えるのは自然数の集合 $\mathbb{N}$ から $\mathbb{N}$ への函数である. 普通の数学では $\mathbb{N}$ から $\mathbb{N}$ への函数といえば、 $\mathbb{N}$ 全体で定義されているものを指すが、計算論では $\mathbb{N}$ の部分集合 $D \subset \mathbb{N}$ で定義されるもの $f: D \rightarrow \mathbb{N}$ も含めて $\mathbb{N}$ 上の函数（正確には「部分函数」）ということが多い. 極端な場合は $D = \emptyset$ で、これは未定義函数ということになる. ($\mathbb{N}$ 全体で定義された函数を「部分函数」と区別するため「全域函数」という.)

1 変数函数に話を限定していることに注意されたい. 多変数函数は変数の組の適当なコード化により 1 変数函数に直せる. また、すでに触れたように、文字列やグラフなどのデー

タを変数とする関数も結局は1変数関数にコード化できる。このように、扱う対象を1変数関数に限定しても、一般性は失われない。

このような（部分）関数の「計算可能性」は次のようにさまざまな形で特徴づけられる：

- Turing 機械で実現できる計算
- λ 計算で記述できる計算
- 帰納的関数
- while プログラムで書ける関数
- ...

最初の3つが Turing, Church, Kleene によって提案されたものに他ならない。「while プログラム」というのは正式な用語ではないが、通常の手続き型プログラミング言語から余計な要素を取り去り、代入文、if 文、while 文という基本的な部分のみを残したプログラムを意味する。

これらの詳細は参考文献 [5] に譲るが、計算機プログラミングに慣れた人には while プログラムによる特徴づけがもっともわかりやすいだろう。この特徴づけにおいては、（部分）関数 f が計算可能であるとは n を引数、 $f(n)$ を返戻値とする普通の意味の計算機プログラムが存在する、ということである。ただし、入力 n に対してプログラムが無限ループに陥る（while 文が含まれていればそういうことが起こり得る）場合には、 $f(n)$ は n に対して定義されない、と解釈する。一般に部分関数を許す形で理論を展開しなければならないのは一つにはこのためであるが、計算可能性の理論では無限ループは単なる「バグ」ではなく、もっと積極的な意味をもっている。

さて、上の説明からすぐに次のことがわかる：

【事実】 $\mathbb{N}$ から $\mathbb{N}$ への計算可能な関数は可算個しかない。つまり、そのような関数に番号をつけて $\phi_0, \phi_1, \dots$ というように枚挙できる。

なぜなら、while プログラムは長さの順に枚挙可能だからである。

他方、 $\mathbb{N}$ から $\mathbb{N}$ への関数の集合は非可算集合である。実際、その中で、全域関数で値が $\{0, 1\}$ だけのもの（つまり Boole 値関数）は $\mathbb{N}$ の部分集合全体のなす集合（ $\mathbb{N}$ のべき集合） $P(\mathbb{N})$ と同濃度の集合をなすが、よく知られているように、 $P(\mathbb{N})$ は非可算集合である。（このことは対角線論法によって証明される。）こうして次のことがわかる：

【事実】 $\mathbb{N}$ から $\mathbb{N}$ への計算可能な関数の中には計算可能でないものが（沢山）存在する。

しかも、単に存在が言えるだけではなく、重要な意味をもつさまざまな計算不可能函数の例を作ることができる。次の例は其中でも最も基本的なものである。

【例】停止性判定函数 $HALT(p, x)$: 2 個の自然数 p, x を受け取り、 p がある while プログラムの自然数コードで、かつそのプログラムが入力 x に対して停止する（無限ループに陥らない）ときに値 1、それ以外のとき値 0 を返す函数を $HALT(p, x)$ と書く。この函数は計算可能でない。

【注釈】これは要するに（自然数コードの形で）与えられたプログラムが与えられた入力に対して停止するかどうかを判定する函数である。（与えられた自然数がプログラムコードであるか否かを判定するのは単なる前処理で、今の場合は本質的ではない。実際、この前処理は字句・構文解析に相当し、計算可能である。） $HALT(p, x)$ の計算不可能性の証明ではこの函数の対角線 $p = x$ に沿う値 $HALT(p, p)$ に注目する。その意味で対角線論法の変形と見ることができる。実は Gödel の不完全性定理の証明も一種の対角線論法である。もっとも、Gödel の定理の方ははるかに複雑で沢山の準備を必要とするのだが。

2.2 補足：帰納的函数

Turing 機械・ λ 計算・帰納的函数のうち、帰納的函数の概念は準備なしですぐに説明できるのでここで触れておこう。（Turing 機械については無限小 Turing 機械について考えるときに、併せて説明する。）帰納的函数は一般に多変数函数として定義されるが、計算可能性の観点からは 1 変数函数となんら変わらないことはすでに注意した通りである。

【定義】帰納的函数とは、次の 3 種類の基本的函数（初期函数）の集合から出発し、後述のいくつかの操作を繰り返し適用することによって得られるものをいう。

【初期函数】次の 3 種類の函数から出発する。

1. ゼロ函数 $f(x_1, \dots, x_n) = 0$.
2. 後者函数 $f(x) = x + 1$.
3. 射影函数 $p_i(x_1, \dots, x_n) = x_i$.

【操作】次の操作を繰り返し適用する。

1. 合成：すでに得られた函数 $f(y_1, \dots, y_m), g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)$ から合成函数

$$h(x_1, \dots, x_n) = f(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n))$$

をつくること.

2. 原始帰納: すでに得られた函数 $f(x_2, \dots, x_n)$ と $g(y, x_1, \dots, x_n)$ から次の関係式で函数 $h(x_1, \dots, x_n)$ を定めること.

$$h(0, x_2, \dots, x_n) = f(x_2, \dots, x_n),$$

$$h(x_1 + 1, x_2, \dots, x_n) = g(h(x_1, x_2, \dots, x_n), x_1, x_2, \dots, x_n).$$

3. 最小化: すでに得られた函数 $f(y, x_1, \dots, x_n)$ から次のような函数 $g(x_1, \dots, x_n)$ を定めること.

$$g(x_1, \dots, x_n) = \min\{y \mid f(y, x_1, \dots, x_n) = 0\}.$$

ただし, 右辺のような y が存在しないときには $g(x_1, \dots, x_n)$ の値は未定義とする.

上記の操作から最小化を除くことで得られる函数のクラスを原始帰納的函数という. これは計算機プログラムで言えば for ループのみを許すプログラムに相当する. それに対して最小化は while ループまで許すことに当たる. たとえば, 上の $g(x_1, \dots, x_n)$ を計算するプログラムは Pascal 風に行けば

```
function g(x1, ..., xn)
begin
  y := 0;
  while f(y, x1, ..., xn) ≠ 0 do
    y := y + 1;
  return y;
end
```

となる.

2.3 IN の帰納的集合と帰納的枚挙可能集合

計算可能性の概念は IN の部分集合にも導入される.

【定義】 IN の部分集合 A の特性函数

$$P_A(n) = \begin{cases} 1 & (n \in A) \\ 0 & (n \notin A) \end{cases}$$

が計算可能 (帰納的) 函数であるとき, A は帰納的であるという.

【定義】 IN の部分集合 A が次の等価な条件のいずれかを満たすとき, A は帰納的枚挙可能であるという.

1. 計算可能な全域函数 $a: \mathbb{N} \rightarrow \mathbb{N}$ が存在して, A はその値域である.
2. 計算可能な部分函数 $p: \mathbb{N} \rightarrow \mathbb{N}$ が存在して, A はその定義域である.

【注釈】

(1) プログラムの言葉で表現すれば, 帰納的枚挙可能集合を特徴づける上の2つの条件はそれぞれ次のように言い換えられる.

1. $a(n)$ を計算するプログラムが存在して, そのプログラムを $n=0$ から出発して n を1ずつ増やす無限ループに組み込めば, A の要素を次々に (ただし一般には重複を許して) 出力するプログラムができる. (a を A の「枚挙函数」という.)
2. 引数 n をもつ while プログラムが存在して, $n \in A$ のとき, かつそのときに限り, このプログラムは停止する. (A をこのプログラムの「停止集合」という.)

(2) 次のことが容易に示せる:

A は帰納的 $\Leftrightarrow A$ と A の補集合 $\bar{A}$ はともに帰納的枚挙可能

次に示すのは基本的な事実だが, Pour-El, Richards の結果の証明においてもこのような事実が重要な役割を演じる.

【事実】 帰納的枚挙可能だが帰納的でない集合が存在する.

【例】 $\phi_0(n), \phi_1(n), \dots$ を (1変数の) 計算可能函数全体, $p_0, p_1, \dots$ をそれらの (適当なコード化における) 自然数コードとする. このとき

$$A = \{n \mid \phi_n(p_n) \text{ は停止する} \}$$

(つまり函数 $n \mapsto \phi_n(p_n)$ の定義域) はそのような集合の例になっている.

この例の検証は $HALT(p, x)$ の計算不可能性の証明と同様の論法によって行われる.

3 Blum-Shub-Smale による $\mathbb{R}$ 上の計算論

Blum, Shub, Smale [1] (以下 B-B-S) の立場では実数は atomic なものとして扱われる. その加算・減算・乗算 (設定によってはさらに除算) は1ステップで実行される基本的演算とみなされる. プログラムは一般にループ辺のある有向グラフ (流れ図) として表現される. 流れ図の接点は次の4種類からなる:

1. 入力節点: これは出て行く辺のみもつ節点で, プログラムの中にただ一つある. 文字通り, ここで入力を行ってプログラムを開始する.

2. 出力節点：これは入ってくる辺のみもつ節点で、プログラムの中に複数あってよい。プログラムはこの節点で終了する。
3. 計算節点：入ってくる辺以外に出て行く辺をただ一つもつ。この節点では多項式（除算を基本的演算として扱うときには有理式）を計算して変数に代入する命令を実行する。
4. 分岐節点：この節点は出て行く辺を2つもつ。この節点ではある多項式の正負に応じてあらかじめ定められた方の辺へ進む。

分岐節点が

if $h(x) < 0$ then goto L1 else goto L2

という条件分岐を表わすことは明らかだろう。これとループ辺を組み合わせれば、while文に相当する制御構造もつくれる。（要するに、Fortran77でプログラムを書く要領である。）

このようなプログラム（あるいは機械）では変数は実数値の連続変数であるが、時間発展はループを回ることによって実現されるので、本質的に離散時間の連続力学系を扱っていることになる。実際、B-S-Sが応用としておもに論じているのは複素力学系の Julia 集合である。次の例（B-S-Sの論文で最初に出てくる）のように、このような場合の Julia 集合はプログラムの停止集合の補集合として特徴づけられる。

【例】1変数多項式 $g(z)$ が Gauss 平面上に生成する離散時間の力学系を考える。正定数 C_g を選べば

$$|z| \geq C_g \Rightarrow \lim_{n \rightarrow \infty} |g^n(z)| = \infty$$

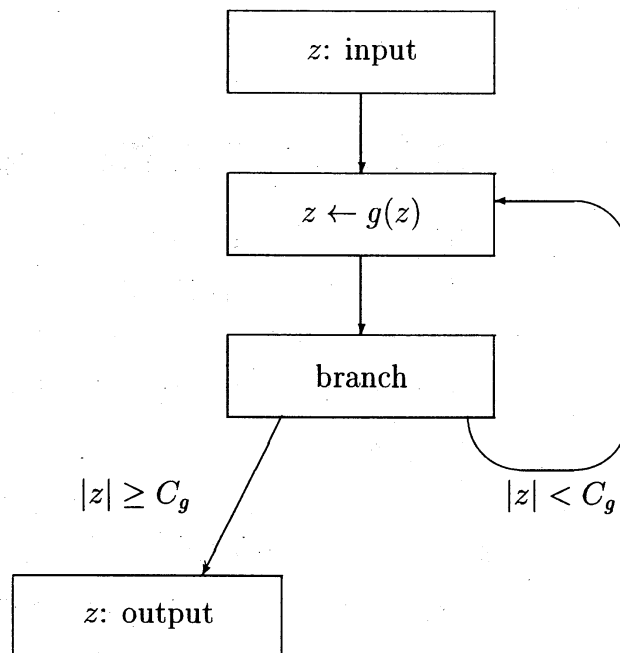
となる。そこで複素数 z を入力として次の図のような流れ図のプログラムを考える。Pascal 風に行けば（ C_g は定数として）

```

procedure p(z)
begin
    while  $|z| < C_g$  do
         $z = g(z);$ 
    return z;
end

```

となる。



$\mathbb{N}$ 上の計算論とのアナロジーから、B-S-S は一般にこのようなプログラム（あるいは機械）の停止集合

$$\Omega = \{z \mid p(z) \text{ は入力 } z \text{ に対して停止する} \}$$

を帰納的枚挙可能集合と呼んでいる。今の場合、それは $\mathbb{C}$ の部分集合で、その補集合が Julia 集合であるが、こちらは一般には（同じ意味で）帰納的枚挙可能ではない — つまり、 $\mathbb{R}$ 上の計算可能集合ではない — ことがわかる。特に 2 次式 $g(z) = z^2 + c$, $|c| > 4$ の場合が本来の Julia 集合であり、その場合の $0 \notin \Omega$ というパラメータ値 c の集合が有名な Mandelbrot 集合である。

以上が機械モデルの簡単な説明であるが、これに引き続いて B-S-S は機械モデルに対する計算量、多項式クラス、NP クラス、NP 完全問題、帰納的函数、というように議論を展開して行く。帰納的函数はここでも、初期函数（ $\mathbb{N}$ の場合とはかなり異なる）から出発して、合成、原始帰納、最小化、というような操作で生成される函数族として定義される。インタープリタに相当する万能機械も示される。

このように、B-S-S の $\mathbb{R}$ 上の計算論は $\mathbb{N}$ 上の計算論とほとんど同じ形で展開される。当然、出てくる結果もよく似ている。離散時間の力学系やフラクタル集合を扱うにはきわめて自然な枠組といえるだろう。しかも、逆に、 $\mathbb{N}$ 上の計算論 — それは数理論理学とも密接な関係をもつ — に対して力学系やフラクタルの視点を持ち込んだらどう見えるか？

という興味深い問題を提起している。しかしながら、微分方程式を扱うにはこれでは不十分であるように思われる。

4 Pour-El, Richards の微分方程式に関する結果

Pour-El と Richards [2, 3] (以下 PL-R) の仕事は Kreisel の掲げた「現存する物理の理論 — たとえば古典力学あるいは量子力学 — は計算可能でない物理定数の存在を理論的に予見できるか」という問題 [6] に対する一つの答を与えるものである。PL-R は常微分方程式の場合 [2] と偏微分方程式 [3] の場合について、計算可能な微分方程式や初期値から計算不可能な解が生じることがあることを示した。

PL-R の実数のとらえ方は B-S-S とは異なる。PL-R が採用している枠組では実数は atomic ではなく、bit や digit に展開されるもの（浮動小数点数）として扱われる。もちろん普通の計算機と違って、無限個の桁をもつとする。そして、計算可能な実数、計算可能な実数列、計算可能な実変数関数、という概念が導入される。この概念に基づいて微分方程式や解の計算（不）可能性を論じる。議論は全体として実函数論・函数解析的である。⁴

4.1 Gregorczyk による実数上の計算論

ここで用いられている実数上の計算可能性の概念は 50 年代から Gregorczyk により研究されていたもので [9], B-S-S よりも歴史が長い。PL-R はそれらの要点を簡潔にまとめている。以下、それを抜粋しておこう：

【定義】有理数列 $\{r_n\}$ が計算可能であるとは、帰納的（つまり $\mathbb{N}$ 上の計算可能）函数 $a, b, c: \mathbb{N} \rightarrow \mathbb{N}$ が存在して

$$r_n = (-1)^{s(n)} \frac{a(n)}{b(n)}$$

と書けることをいう。

【定義】実数 x が計算可能であるとは、 x に「effective に収束する」計算可能な有理数列 $\{r_n\}$ が存在することをいう。ここで「effective に収束する」とは、帰納的函数 $e: \mathbb{N} \rightarrow \mathbb{N}$ が存在して

$$k \geq e(n) \Rightarrow |x - r_k| \leq 10^{-n}$$

という不等式が満たされることをいう。⁵

⁴ 実際、PL-R はその後、函数解析的な方向に研究を展開している [7]。これは「構成的数学」という数理論理学の分野と関連がある [8]。

⁵ 10 のべきを使うことには特に意味はない。2 のべきでも構わない。

【注意】 r_n を $r'_n = r_{e(n)}$ で置き換えることにより、上の不等式を

$$|x - r_n| \leq 10^{-n}$$

としても一般性は失われないことがわかる。

【例】 $\mathbb{N}$ の帰納的でない帰納的枚举可能部分集合 A を一つとる。(このような集合の例はすでに示した。) その枚举函数 $a: \mathbb{N} \rightarrow \mathbb{N}$ は単射, つまり $m \neq n \Rightarrow a(m) \neq a(n)$, であるとしてよい。⁶ このとき

$$x = \sum_{n=0}^{\infty} 10^{-a(n)}$$

により実数が定まるが、これは上の意味で計算可能でない。理由はこうである：この数は 10 進小数で書けば

$$0.\cdots a(n_0)\cdots a(n_1)\cdots a(n_2)\cdots$$

(「 $\cdots$ 」には 0 個以上の 0 が並ぶ) となる。仮に、この数を上の注意のように近似する計算可能な有理数列が存在すれば、 $a(n)$ のみならず 0 の出現位置 (それは A の補集合に他ならない) も帰納的函数で枚举できることになり、 A が帰納的でない ($\Leftrightarrow A$ の補集合が帰納的枚举可能でない) ということに矛盾する。従って、 x は計算可能でない。

【定義】 実数の組 (ベクトル) $(x_1, \cdots, x_p) \in \mathbb{R}^p$ が計算可能であるとは、各成分 $x_1, \cdots, x_p$ が計算可能な実数であることをいう。

【定義】 $\mathbb{R}^p$ の計算可能な有界閉区間とは $\{(x_1, \cdots, x_p) \mid a_1 \leq x_1 \leq b_1, \cdots, a_p \leq x_p \leq b_p\}$ という形の閉区間で端点の値 a_i, b_i がいずれも計算可能実数であるものである。

【定義】 実数列 $\{x_n\}$ が計算可能であるとは、計算可能な 2 重有理数列 $\{r_{kn}\}$ が存在して、すべての k, n に対して

$$|x_k - r_{kn}| \leq 10^{-n}$$

という不等式が満たされることをいう。

【定義】 計算可能な有界閉区間 $I \subset \mathbb{R}^p$ 上の実数値函数 $f: I \rightarrow \mathbb{R}$ が計算可能であるとは次の 2 つの条件が満たされるである：

(a) f は点列的に計算可能、すなわち、 I 内の任意の計算可能点列 $\{x_k\}$ に対して $\{f(x_k)\}$ が計算可能数列である。

⁶ $a(n)$ を $a(0), \cdots, a(n-1)$ と比較して、重複があれば $a(n)$ を出力しないようなプログラムが書けるから。しかし、 $a(n)$ は例えば決して単調増加にはならない。単調増加ならば、 A の補集合を枚举する函数のプログラムが a を使ってつくれる。

(b) f は「effective に一様連続」、すなわち、帰納的函数 $d: \mathbb{N} \rightarrow \mathbb{N}$ が存在して、すべての $x, y \in I$ に対して

$$|x - y| \leq 1/d(n) \Rightarrow |f(x) - f(y)| \leq 10^{-n}.$$

— 同様にして、計算可能な函数列、その「effective で一様な収束」、などの概念が定義される。その定義のもとで、『計算可能な函数列の effective で一様な極限はやはり計算可能な函数になる』などの基本的結果が従う。

4.2 Pour-El, Richards の得た結果

PL-R は以上のような計算可能性の概念のもとで、微分方程式と解の計算（不）可能性を考察している。

1. 常微分方程式の場合 [2]

考察の対象は正規形の方程式

$$\frac{dy}{dx} = F(x, y)$$

で、右辺の函数は計算可能連続函数である。PL-R はこのような微分方程式で、ある値 $x = a$ の近房でどの解も計算不可能函数になるようなものが存在することを示した。この場合、Lipschitz 条件を仮定しないのがミソで、解が一意に決まらないような微分方程式を次々に貼り合わせてゆく巧妙な（しかしかなり人工的な）やり方で目的の微分方程式をつくっている。また、Lipschitz 条件を仮定するとどの解も計算可能函数になる、という結果も示している。

2. 偏微分方程式の場合 [3]

考察の対象は波動方程式

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} - \frac{\partial^2 u}{\partial t^2} = 0$$

である。こちらは $t = 0$ における初期値問題

$$u(x, y, z, 0) = f(x, y, z), \quad u_t(x, y, z, 0) = 0$$

の初期データ $f(x, y, z)$ を計算可能函数として与えて、解 $u(x, y, z, t)$ の計算可能性を問題にする。この場合には、解に対する積分公式（Kirchhoff の公式）

$$u(\mathbf{x}, t) = \int_{|\mathbf{n}|=1} [f(\mathbf{x} + t\mathbf{n}) + t(\nabla f)(\mathbf{x} + t\mathbf{n}) \cdot \mathbf{n}] d\sigma(\mathbf{n})$$

を用いて解析を行うことができる。これを用いて、

- ある時空点における解の値が計算不可能数になる例
- ある時刻における解が $\mathbb{R}^3$ 上の計算不可能関数になる例

などを示している。前者の結果は証明のアイデアが比較的わかりやすいので、以下に紹介する。

【定理】 [3][Theorem 1] 計算可能な初期データ $f(x, y, z, t)$ で、 $u(0, 0, 0, 1)$ が計算不可能数になるようなものがある。

【証明の概略】 f を球対称関数 $f = f(r)$, $r = (x^2 + y^2 + z^2)^{1/2}$, に選ぶ。このとき Kirchhoff の公式から

$$u(0, 0, 0, t) = f(t) + tf'(t)$$

となることがわかる。そこで、 $f(t)$ として $r \geq 0$ 上のコンパクトな台をもつ 1 回連続微分可能関数で次のような条件を満たすものを構成すれば証明が終わる：

- $f(r)$ は計算可能関数、特に $f(1)$ は計算可能数。
- $f'(1)$ は計算不可能数、特に $f'(r)$ は計算不可能関数。

そのために、まず、 $[-1/2, 1/2]$ に台をもつ C^∞ 関数 $\phi(x)$ で、 $\phi(x) \geq 0$ かつ $\phi'(0) = 1$ となるものを一つ選ぶ。PL-R は

$$\phi(x) = \begin{cases} (1+x) \exp[-x^2/(1-4x^2)] & (|x| < 1/2) \\ 0 & (|x| \geq 1/2) \end{cases}$$

という例を与えている。さらに（ここが PL-R のアイデアの核心なのだが） $\mathbb{N}$ の帰納的でない帰納的枚挙可能部分集合 A とその単射な枚挙可能関数 $a: \mathbb{N} \rightarrow \mathbb{N}$ を一つ選ぶ。そして

$$f(r) = \sum_{n=0}^{\infty} w_n(r), \quad w_n(r) = 10^{-n-a(n)} \phi(10^n(r-1)),$$

というように $f(r)$ を定義する。 w_n は計算可能関数の列で、 f を定義する上の級数は各項の絶対値が 10^{-n} で抑えられるので、effective かつ一様に収束する。特に、 $f(r)$ は $\mathbb{R}$ 上の計算可能関数である。しかしその導関数はそうではない。実際、導関数は

$$f'(r) = \sum_{n=0}^{\infty} w'_n(r) = \sum_{n=0}^{\infty} 10^{-a(n)} w'_n(10^n(r-1))$$

と書けて（ 10^{-n} という因子が消えた！）、 $\sum 10^{-a(n)}$ を優級数にもつので一様収束するが、もはや effective に収束することが保証されない。（すでに例として注意したように、 $\sum 10^{-a(n)}$

という級数は effective な収束をしない．その結果，値は計算不可能数となる．）計算不可能性をもっと正確に確かめるために， $r = 1$ での値を調べると，

$$f'(1) = \sum_{n=0}^{\infty} 10^{-a(n)}$$

となり，確かにこれは計算不可能数である． $f(1)$ の方は上の議論から計算可能数なので，結局 $u(0, 0, 0, 1) = f(1) + f'(1)$ は計算不可能数であることがわかった．*Q.E.D.*

— 上の証明が示すように，PL-R は $\mathbb{N}$ 上の計算論を解析的な考察ときわめて巧妙に組み合わせている．波動方程式についての結果や常微分方程式の結果を導くときには，次のような（ $\mathbb{N}$ 上の計算論ではやはり基本的な）次の事実を利用している：『 $\mathbb{N}$ の帰納的枚举可能な部分集合 A, B で， $\mathbb{N}$ のいかなる帰納的部分集合 C に対しても $A \subset C$ と $B \subset C$ が同時には成り立たないようなものが存在する』．

4.3 物理学との関連？

数理論理学の奇才 Roger Penrose がその著書 "The Emperor's New Mind" [10]⁷ の中で PL-R の仕事を引用している．Penrose がこの著書（発表された後おおいに議論を巻き起こした由）とその続編 [11] で主張しているのは，いまのコンピュータはいくらがんばっても人間のように考えることはできない，逆に言えば，人間の思考能力は原理的に Turing 機械やそれと同等の計算原理を超えたものである，ということである．その意味で（古典力学に関する話ではあるが）微分方程式の計算能力が Turing 機械を超えることを示唆する PL-R の結果は気になるところなのだろう．もっとも，Penrose は Turing 機械を超える脳の思考（もはや計算とは言わないのかも知れない）の能力の原理を最終的には量子論に求めている．⁸

ちなみに，素因数分解をきわめて高速で行えることがわかって話題になった「量子コンピュータ」[12] は通常のビットを「量子ビット」に置き換えて並列性を高めているが，計算可能性のみを考えれば計算能力は Turing 機械と等価である．しかし，連続空間に広がった Schrödinger の波動関数を計算媒体に使えるならば，計算能力は質的に違ってくるだろう．

⁷ 訳者はたぶん仕方なく「皇帝の新しい心」と訳しているが，原題は "The Emperor's New Clothes" つまり「裸の王様」のもじりである．"The Emperor's New Clothes" が日本では「裸の王様」と訳されているので，原題に忠実な訳がピンと来ないものになっているのは気の毒である．敢えて対応する邦題を考えるならば「無脳の王様」とでもなるのだろうか？

⁸ もっとも，「皇帝の新しい心」における Penrose の議論は量子力学をはるかに超えて量子重力にまで及んでいる．最後の章では，量子重力と脳が関係しているかも知れない，という衝撃的な見解が飛び出す．

同様のことは PL-R の扱った波動方程式でも言えそうだ。偏向などを無視して大ざっぱに考えれば、この方程式が記述するのは電磁波、特に光の古典論である。光コンピュータが「光ビット」に基づいて構築されるならば、量子ビット同様、重ね合せによる並列演算が可能であろうが、その計算能力はやはり Turing 機械を超えない。しかし連続空間に広がった場を計算媒体とする機械は $\Sigma 10^{-a(n)}$ のような計算不可能数を出力する能力をもっていることになる。

もっとも、Schrödinger 方程式と波動方程式では偏微分方程式の性質がまったく違うので、いまのところ Schrödinger 方程式の計算能力については何とも言えない。そもそも、PL-R の構成した解は $t=0$ では滑らかだが $t=1$ で focusing を起こして滑らかさが破れる（弱解になる）のが特徴で、そのことが Gregoryczyk の意味での計算可能性が破れる原因である。線形の Schrödinger 方程式にはそのような現象はなさそうであるが、量子カオスが何か面白い現象をもたらす可能性はある。（もっとも、そのためには、差分方程式が必要かも知れない。）

非線形方程式はどうだろうか。特に、パターン形成を伴う反応拡散系、時空カオスを生じる蔵本-Sivashinsky 方程式、ソリトンをもつ分散性非線形波動方程式、などはいかにも「アナログ計算機」の候補としてふさわしい。ただし、これらは本来「巨視的スケール」で現象をモデル化した方程式である。これらの計算能力を PL-R のいわば「微視的」な計算可能性の枠組で扱うことにどれほど意味があるのか俄かには判断できない。微視的スケールの法則をいろいろ考えることはできるが、任意性があって、巨視的スケールの法則からは一意に決めることはできない。しかし、これらの巨視的方程式の妥当性が破れるところではミクロの構造が見えてくることもある。そういうところが一種の計算可能性の破れを引き起こすのかも知れない。

5 辻下による超準解析的アプローチ

辻下は「形式的方法」に基づく複雑系研究 (FCS = Formal Methods for Complex System Research) [13] の一環として、微分方程式を超準解析的枠組で記述することを試みている [4]。超準解析はすでに確率微分方程式にさえ応用されているので（たとえば竹内外史の本 [14] を見よ）、普通の微分方程式を超準解析で扱うことはそれほど不自然ではない。しかし微分方程式の計算可能性に結び付けるにはもう一段の飛躍が必要だろう。そこで辻下は無限小 Turing 機械の概念を導入する [4]。これも実数上の計算可能性に対する「微視的」アプローチといえるが、PL-R とはまったく異なる原理である。超準解析では微分方程式を無限小変位についての差分方程式とみなすことができる（一種の離散化！）。この「ミク

口の離散構造」を記述するのが無限小（あるいは超準）Turing 機械であり，そこから「マクロの連続構造＝微分方程式の解」が生まれてくる，というのが複雑系論において辻下の描く微分方程式像のようである [15].

超準解析 (nonstandard analysis) は数学基礎論から生まれたもので，自然数や実数の公理系にはわれわれが普通に用いている標準モデル $\mathbb{N}$, $\mathbb{R}$ 以外に非標準（超準）モデル ${}^*\mathbb{N}$, ${}^*\mathbb{R}$ がある，という事実（これも Gödel の一連の仕事と関係がある）に基づいている．その意味で，計算論とはむしろ相性がよい．実際，自然数や各種の代数系の超準モデルを計算論の研究に持ち込むことはよくあるらしい．（証明論と計算量を扱った竹内外史の本 [16] にその例がある．）

非標準モデルは本来はこのように数学基礎論や数理論理学の研究対象であるが，Robinson [17] によって解析学に応用できることが示されて超準解析へと発展した．世間では超準解析と言えば微分積分学（無限小解析）への応用しか思い浮かべない向きが多いが，本来は解析学全体への応用をめざした構想である．事実，超準解析が始まって間もなく，函数解析学への応用が現れている [18]．その後，Keisler による初等的なアプローチ [19]，Loeb による測度論・確率論への応用 [20]（これが現在の超準解析的確率論の始まり），Nelson の内的集合論による再構築 [21]，などを経て現在に至っている．また，数学のみならず数理論理学にも輸出されている [22]．最近では，小沢正直・小嶋泉による量子力学の超準解析的再構成 [23] という興味深い試みもある．

以下では，超準解析について簡単な解説をした後で，辻下の講義資料 [4] から，無限小 Turing 機械と実数上の計算論を扱った部分を抜き書きする．残念ながら，講義資料のこの部分の記述がはかなり断片的なので，⁹ ここではきわめて不完全な紹介しかできない．

5.1 超準解析

すでに述べたように，超準解析の出発点は通常の実数や自然数の世界（標準モデル）を超準モデル ${}^*\mathbb{N}$, ${}^*\mathbb{R}$ に広げることである．もちろんこれは勝手な拡大でなく，次のような要請を満たすものである：

- 標準モデルの言葉だけからなる主張 (sentence) ϕ の真偽はそれを超準モデルの言葉に読み替えたもの ${}^*\phi$ の真偽と等価である．
- a を自由変数として，標準モデルの言葉だけからなる論理式 (formula) の集まり $\phi_i(x)$ ($i \in I$) があるとする． I から任意に有限個の $i_1, \dots, i_n$ をとりだすとき， $\phi_{i_1}(x), \dots, \phi_{i_n}(x)$

⁹ 1 年間の講義もそろそろ終わりで，時間が足りなくなっただけ

を同時に満たす x が標準モデルの中に存在するとする。このとき、超準モデルの中に ${}^*\phi_i(x)$ をすべて満たす x が存在する。

第一の要請はわかりやすい。たとえば、これによって $\mathbb{N}$, $\mathbb{R}$ における加・減・乗（・除）算はそのまま ${}^*\mathbb{N}$, ${}^*\mathbb{R}$ に拡張されるし、整数部分を取り出す関数 $[\cdot] : \mathbb{R}_+ \rightarrow \mathbb{N}$ も $[\cdot] : {}^*\mathbb{R}_+ \rightarrow {}^*\mathbb{N}$ に拡張される。 $\mathbb{N}$ における数学的帰納法も ${}^*\mathbb{N}$ で成立することになる。（これは帰納的関数の概念を ${}^*\mathbb{N}$ の中で考えるためには欠かせないことである。）

第二の要請はこれに比べて技術的でわかりにくいので、使用例を示す：

【例：無限大の要素の存在】 $I = \mathbb{N}$ で $\phi_i(x)$ が「 $i < x$ 」という関係をあらわす場合を考えてみよう。このとき、有限個の $i_1, \dots, i_n$ に対して $\phi_{i_1}(x), \dots, \phi_{i_n}(x)$ を満たす $\mathbb{N}$ の要素は確かに存在する。したがってすべての $i \in \mathbb{N}$ に対して $i < x$ となる x が ${}^*\mathbb{N}$ の中に存在する。この x はすべての自然数より大きいので、「無限大」の要素ということになる。

— こうして、 ${}^*\mathbb{N}$ は $\mathbb{N}$ より真に大きい集合で、その中には無限大の自然数というべきものの（超準自然数）が入っていることがわかる。さらに x に対して

$$\begin{aligned} \dots < x-2 < x-1 < x < x+1 < x+2 < \dots, \\ x < 2x < 3x < \dots < x^2 < \dots < x^3 < \dots, \end{aligned}$$

となるが、これらはいずれも超準自然数である。したがって、超準自然数もたくさんあることになる。同様の論法で ${}^*\mathbb{R}$ の中にも無限大の要素が入っていることもわかる。 ${}^*\mathbb{R}$ にはさらに無限小の要素も入っている。

【例：無限小の要素の存在】 a を任意の実数として、 $I = \mathbb{N}$, $\phi_i(x)$ が「 $0 < |x - a| < 1/i$ 」という関係をあらわす場合を考える。この場合もこのうちの任意の有限個に対しては同時に関係が成立するような x が必ず存在する。したがって、上の要請から、すべての $i \in \mathbb{N}$ に対して $0 < |x - a| < 1/i$ となる x が ${}^*\mathbb{R}$ の中に存在する。この x は a と「無限小」の差しかない要素ということになる。特に、 $a = 0$ とすれば、 x は無限小の要素である。

— このように、 ${}^*\mathbb{R}$ は $\mathbb{R}$ よりも真に大きい集合で、普通の実数のまわりに無限小の近房を従えている。そのうえに無限に大きい要素ももつという、途方もない代物であることがわかる。（ただし、反転写像 $x \mapsto 1/x$ により無限小と無限大は移りあうので、無限大要素の集合は原点の近房の反転像に過ぎない。）

一般に、 ${}^*\mathbb{R}$ の2つの要素 x, y の差が無限小であるときに「 $x \simeq y$ 」と書くことにする。また、 ${}^*\mathbb{R}$ の有界な要素 x には $x \simeq y$ となる $\mathbb{R}$ の要素 y がただ一つ存在することがわかる。これを「 x の標準部分」と呼んで「 $st(x)$ 」と書く。これらが無限小解析への応用にお

いて基本的な役割を演じる。例えば、 $\mathbb{R}$ 上の函数 $f(x)$ が $x = a$ において連続であるということは、「 $x \in {}^*\mathbb{R}$ かつ $x \simeq a$ ならば $f(x) \simeq f(a)$ 」というように簡潔に表現される。¹⁰ 連続性、一様連続性、微分可能性、連続微分可能性などの概念も同様に無限小を用いて直接的な表現ができる。たとえば微分係数（導函数）は

$$\frac{f(x) - f(a)}{x - a} \simeq f'(a) \quad (x \simeq a)$$

という表現ができる。

積分の方は逆に、無限小を無限に足し上げて定義する、という表現が可能になる：

$$\int_a^b f(x) dx = st \left(\epsilon \sum_{0 \leq i\epsilon < b-a} f(a + i\epsilon) \right).$$

ここで ϵ は適当に選んだ正の無限小数である。（例えば正の超準自然数 N を選んで $\epsilon = 1/N$ とおく。）ただし、このままでは右辺をどのように意味づけるか明らかではないが、それについては省略する。Keisler の本などを参考にして欲しい。

常微分方程式も同様の考え方で扱うことができる。初期値問題

$$\frac{dx}{dt} = f(t, x), \quad x(a) = b,$$

の形で考える。 $f(t, x)$ に対しては (a, b) の有限近房で連続函数であるとする。このとき初期条件を満たす（一つの）解が次のように構成できる。（ f の連続性だけでは解の一意性は保証されないので、他にも解が有り得る。¹¹）以下、 $[a, a']$ という区間で解をつくることを考える（反対側も同様）。まず、正の無限小数 ϵ を一つ選ぶ。いま、求めるような解があれば

$$u(t + \epsilon) \simeq u(t) + f(t, u(t))\epsilon, \quad u(a) = b,$$

となる。そこで、 $[a, a']$ の代わりに「格子」 $L = \{a + i\epsilon \mid i \in {}^*\mathbb{N}, i < N\}$ （ N は超準自然数で、 $L \subset [a, a']$ となるもの）の上で「差分方程式」

$$v(t + \epsilon) = v(t) + f(t, v(t))\epsilon, \quad v(a) = b,$$

を考えれば、こちらはすぐに

$$v(t) = b + \sum_{a \leq s < t} f(s, v(s))\epsilon$$

¹⁰ f が ${}^*\mathbb{R}$ に拡張されるかどうかは気になるところだが、 f が完全に「論理的」に特徴づけられていれば、上の第一の要請により、それらをそのまま ${}^*\mathbb{R}$ の言葉に読み替えたものを満たす拡張が ${}^*\mathbb{R}$ 上に存在することになる。超準解析は論理（＝言葉）がすべてなので、万事この調子である。

¹¹ この非一意性をどのように扱うべきか、辻下の講義資料でははっきりしない。

と解ける。(ただし、右辺は実質的には ${}^*\mathbb{N}$ の要素を添え字とする和であることに注意。定積分の表示にも同じようなものが出てきた。これはそれほど自明なものでない。) そこで

$$u(t) = st \left(v \left(a + \left\lfloor \frac{t-a}{\epsilon} \right\rfloor \right) \right)$$

とおく。ここで $[\cdot]: {}^*\mathbb{R}_+ \rightarrow {}^*\mathbb{N}$ はすでに触れた整数部分である。あとはこれが求める解であることを確かめればよい(省略する)。

この解の構成法をみると、Cauchy-Peano の折れ線法による証明の複雑さはどこへ行ったのかと思う。もちろん、いろいろ端折った部分や超準解析の一般的事実の中に難しい部分が吸収されているわけである。それでも、物事の本質を浮き彫りにするという超準解析(あるいは超準数学一般)の特徴がここにもよくあらわれている。

5.2 Turing 機械

まず Turing 機械の定義と動作を説明する。

【Turing 機械の定義】次のようなものの三つ組 (Σ, Q, δ) を Turing 機械という：

- Σ はテープ記号の集合(有限)である。これは空白記号 B および区切り記号 $\#$ を含む。
- Q は状態集合(有限)である。これは初期状態 q_0 と停止状態 H を含む。
- δ は状態遷移関数 $\delta: \Sigma \times Q \rightarrow \Sigma \times Q \times \{L, R, 0\}$ であり、次の条件(停止状態の特徴)を満たす。

$$\delta(\sigma, H) = (\sigma, H, 0).$$

【状態】Turing 機械は直観的には、両側に無限に伸びたテープ(メモリーに相当する)、テープの読み取りヘッド、および内部状態を保持する制御部(CPUに相当する)からなる機械である。テープには Σ に属する記号が並んでいるが、それらは有限個を除いて空白文字 B である。機械全体の状態を $\Sigma \cup Q$ の要素からなる無限文字列

$$\cdots \sigma_{n-1} q \sigma_n \sigma_{n+1} \cdots$$

で表現する。 Q の要素(ここでの q) は一つだけ現れる。 q 以外の文字はテープ($\mathbb{Z}$ の番地が割り振ってある)の状態であり、 q は制御部の状態、 q の右隣り(つまりテープの n 番地)はヘッドの位置をあらわしている。この無限列の代わりに有限列を示すときには、そ

の両側は空白文字で埋められているものとする。(例: $q_0\omega$ は $q_0 \cdots BBq_0\omega BB \cdots$ を意味する.)

【動作】 Turing 機械は $q_0\omega$ (ω は Σ 上の有限文字列) という形の状態から出発し, 制御部の内部状態とヘッドの位置の記号を参照しながら, 次々状態を変えて行く. 現在の状態が $\cdots \tau q \sigma \cdots$ のときには次のように状態を変える:

- $\delta(q, \sigma) = (q', \sigma', 1)$ ならば $\cdots \tau q \sigma \cdots \mapsto \cdots \tau \sigma' q' \cdots$ (ヘッド位置の記号を σ' に, 内部状態を q' に変えて, ヘッドを右へ動かす),
- $\delta(q, \sigma) = (q', \sigma', -1)$ ならば $\cdots \tau q \sigma \cdots \mapsto \cdots q' \tau \sigma' \cdots$ (ヘッド位置の記号を σ' に, 内部状態を q' に変えて, ヘッドを左へ動かす),
- $\delta(q, \sigma) = (q', \sigma', 0)$ ならば $\cdots \tau q \sigma \cdots \mapsto \cdots \tau q' \sigma' \cdots$ (ヘッド位置の記号を σ' に, 内部状態を q' に変えて, ヘッドは動かさない).

一般に状態 ω から出発して上の動作を有限回繰り返したあと状態 ω' に達することを

$$\omega \vdash \omega'$$

と書く. Turing 機械は $H\omega'$ という状態に到達したら停止する.(この状態に達しないとき Turing 機械はループに落ち込んでいる.)

Turing 機械は現在の計算機に似ているようで, 実はあまり似ていない. 状態遷移関数は普通の計算機の CPU 命令のようなものであるが, Turing 機械ではこれも自由に設定することができる. これはプログラム内臓方式以前の計算機(配線を変えることでプログラムしていたという)を思わせるところがある. 他方, CPU レベルの加算命令や飛び越し命令などもなく, レジスターも使えない. そういうわけで, Turing 機械のプログラミングは独特のものになる.(ここではプログラミングの例には立ち入らない.) もちろん, 適当なインタプリタをつくっておけば, 現在の計算機のプログラミング環境に近づけることができるが.

Turing 機械と $\mathbb{N}$ 上の関数を関係づけるためには関数の引数と返戻値をテープ記号列にコーディングする必要がある. 標準的な方法は自然数 n の 1 進表現 1^n (1 の n 個の並び) である. これと区切り記号 $\sharp$ を使って, 引数の組 $(n_1, \dots, n_k)$ を $1^{n_1}\sharp \cdots \sharp 1^{n_k}$ というテープ文字列で表現する. このとき, 部分関数 $f: \mathbb{N}^k \rightarrow \mathbb{N}$ が Turing 機械 (Σ, Q, δ) で計算可能であるとは,

$$f(n_1, \dots, n_k) = n \Leftrightarrow q_0 1^{n_1} \sharp \cdots \sharp 1^{n_k} \vdash H 1^n$$

という意味である. これがその他の古典的計算可能性の概念と等価であることはすでに述べたとおりである.

5.3 無限小 Turing 機械と計算可能函数

辻下のいう無限小 Turing 機械とは最後に述べた自然数データが ${}^*\mathbb{N}$ に移行したものである。(Turing 機械の基本的な書き換え動作を無限回繰り返せるようにしたようなものだから、無限 Turing 機械という方が適切かも知れない。)したがって、機械そのものよりもそれが計算する函数 $f(n_1, \dots, n_k)$ で考える方が数学的にはわかりやすい。¹²

非標準的(つまり無限大の)自然数 η を一つ選び、その逆数(正の無限小実数となる)を ϵ とおく。さらに、任意の標準的(つまり普通の)実数に対して

$$N(x) = \left\lfloor \frac{x}{\epsilon} \right\rfloor$$

とおく。($\lfloor \cdot \rfloor$ はここでも整数部分をあらわす。) $x = 0$ でない限りこれは非標準整数(${}^*\mathbb{Z}$ の要素)となる。また

$$x \simeq N(x)\epsilon.$$

この $N(x)$ を通じて実数函数の計算可能性を ${}^*\mathbb{Z}$ 上の計算可能性として理解するのである:

【定義】部分函数 $f: \mathbb{R}^k \rightarrow \mathbb{R}$ が計算可能であるとは、古典的な意味での計算可能函数 $F: \mathbb{Z}^{k+1} \rightarrow \mathbb{Z}$ が存在して

$$f(x_1, \dots, x_n) \simeq F(N(x_1), \dots, N(x_n), \eta)\epsilon$$

となることをいう。(ただし、右辺の F は ${}^*\mathbb{Z}$ への拡張を意味する。)

【例】(すべて辻下の講義資料に出てくる。)

1. 実数の加算 $f(x_1, x_2) = x_1 + x_2$.

$$x_1 + x_2 \simeq (N(x_1) + N(x_2))\epsilon.$$

2. 実数の乗法 $f(x_1, x_2) = x_1 x_2$.

$$x_1 x_2 \simeq (N(x_1)N(x_2)/\eta)\epsilon.$$

3. 計算可能函数の原始函数 $\int_0^x f(x)dx$. 実際、 $\mathbb{Z}^2$ 上の計算可能函数 $F(n, m)$ によって $f(x) \simeq F(N(x), \eta)\epsilon$ となるとき、原始帰納

$$G(n+1, m) = G(n, m) + F(n, m), \quad G(0, m) = 0,$$

¹² しかし、ミクロの世界でこういう機械がせっせと働いているという描像 — 細胞内の代謝系を思い浮かべるとよいのだろうか? — も、いかにも複雑系らしくていい。

により $\mathbb{Z}^2$ 上の計算可能函数 $G(n, m)$ を定めると,

$$\int_0^x f(t) dt \simeq G(N(x), \eta) \epsilon.$$

4. 計算可能函数からなる常微分方程式の初期値問題

$$\frac{dx}{dt} = f(t, x), \quad x(a) = b,$$

の解.¹³ 実際, $f(t, x)$ が $\mathbb{Z}$ 上の計算可能函数 $F(n, m, \ell)$ と

$$f(t, x) \simeq F(N(t), N(x), \eta) \epsilon$$

という関係にあるとき, 微分方程式の解 $x(t)$ は

$$x(t) \simeq X(N(t), \eta) \epsilon$$

という関係を満たす. ただしここで, $X(n, \ell)$ は原始帰納

$$X(n+1, \ell) = X(n, \ell) + F(n, X(n, \ell), \ell), \quad X(0, \ell) = N(a),$$

で定まる $\mathbb{Z}$ 上の計算可能函数.

ここで定義された意味での計算可能な実数函数の概念と P-L-R が採用した計算可能な実数函数の関係はまだよくわからない. 両者の主張を見比べると, P-L-R の常微分方程式は辻下の意味では計算不可能な函数からできているのではないか, と思われる.

参考文献

- [1] L. Blum, M. Shub and S. Smale, On a theory of computation and computability over the real numbers: NP-completeness, recursive functions and universal machines, Bull. AMS **21** (1989), 1-46.
- [2] M.B. Pour-El and I. Richards, A computable ordinary differential equation which possesses no computable solution, Ann. Math. Logic **17** (1979), 61-90.
- [3] M.B. Pour-El and I. Richards, The wave equation with computable initial data such that its unique solution is not computable, Adv. Math. **39** (1981), 215-239.

¹³ ここで常微分方程式にたとえば Lipschitz 条件のように解の一意性を保証する条件をおくのかどうかは辻下の講義資料でははっきりしない. ここではほとんど原文のまま説明を書き写している.

- [4] 辻下徹, 超準数学入門, 北大での全学教育科目 1996/7 数学概論 B の講義資料.
- [5] いくつか教科書・入門書を掲げておく.
井田哲雄, 計算モデルの基礎理論 (岩波書店, ソフトウェア科学講座, 1991).
高橋正子, 計算論 — 計算可能性とラムダ計算 (近代科学社, 1991).
小林孝次郎, 計算可能性入門 (近代科学社, 1980).
渡辺治・米崎直樹, 計算論入門 (日本評論社, 1997).
- [6] G. Kreisel, A notion of mechanistic theory, *Synthese* **29** (1974), 11–16.
- [7] M. Pour-El and I. Richards, Noncomputability in analysis and physics: a complete determination of the class of noncomputable linear operators, *Adv. Math.* **48** (1983), 44–74; Three theorems on the computability of linear operators, their eigenvalues and eivenvectors, 数理解析研究所購究録 **588** (1986), 149–161.
- [8] S. Feferman, Between constructive and classical mathematics, in *Computation and Proof Theory*, Lect. Notes. Math. **1104** (Springer-Verlag, 1984).
- [9] A. Gregorczyk, Computable functions, *Fund. Math.* **42** (1955), 168–202; On the definitions of computable real continuous functions, *Fund. Math.* **44** (1957), 61–71.
- [10] R. Penrose, *The Emperor's New Mind* (Oxford University Press, 1989) (邦訳「皇帝の新しい心」, 林一訳, みすず書房, 1994)
- [11] R. Penrose, *Shadows of the Mind* (Oxford Universsity Press, 1994)
- [12] 西野哲朗, 量子コンピュータ入門 (東京電気大学出版局, 1997)
- [13] FCS ホームページ, URL <http://fcs.math.sci.hokudai.ac.jp/>.
- [14] 竹内外史, 無限小解析と物理学 (遊星社, 1985).
- [15] 辻下徹, 複雑系とは／複雑系私的序説, 数学セミナー 1994年8月号, 64–70.
- [16] 竹内外史, 証明論と計算量 (裳華房, 1996); P と NP (日本評論社, 1997).
- [17] A. Robinson, *Non-standard analysis* (North-Holland, 1966).
- [18] A.R. Bernstein and A. Robinson, Solution of an invariant subspace problem of K.T. Smith and P.R. Halmos, *Pacific J. Math.* **16** (1966), 421–431.

- [19] H.J. Keisler, *Foundations of Infinitesimal Calculus* (Prindle, Weber and Schmidt, Boston, 1976) (邦訳「無限小解析の基礎」, 斎藤正彦訳, 東京図書, 1979)
- [20] P.A. Loeb, Conversion from nonstandard to standard measure space and applications in probability theory, *Bull. AMS* **211** (1975), 113-122.
- [21] E. Nelson, Internal set theory: a new approach to nonstandard analysis, *Bull. AMS* **83** (1977), 1165-1198; Radically elementary probability theory, *Annals of Math. Stud.* **117** (Princeton University Press, 1987).
- [22] S. Albeverio, J.E. Femstad, R. Høegh-Krohn and T. Lindstrøm, *Nonstandard methods in stochastic analysis and mathematical physics* (Academic Press, 1986).
- [23] 小沢正直, 無限小解析と観測問題, *素粒子論研究* **83** (1991), F100-F117.
M. Ozawa, Operator algebras and nonstandard analysis, in *Current Topics in Operator Algebras* (ed. H. Araki et al) (World Scientific, Singapore, 1991), pp. 52-58.
I. Ojima and M. Ozawa, Unitary representations of the hyperfinite Heisenberg group and the logical extension methods in physics, *Open Systems and Information Dynamics* **2** (1993), 107-128.