フォン・ノイマンと「自己」の問題
――セルラー・モデルと自己複製――

大西琢朗*

はじめに

19 世紀終わりから 20 世紀初め。数学基礎論論争は、数学者・論理学者たちの「自己」を巡る悪戦苦闘であった。ラッセルの「自分自身を要素として含まない集合全体の集合」は、矛盾を引き起こし、論理主義を標榜するフレーゲの野望を打ち砕いた。ある形式系で表現された「この文は証明できない」は、真だけれども、その形式系内部では証明できない。数学を数学自身で基礎づけようとしたヒルベルトの夢は、ゲーデルの不完全性定理により、文字通り夢に終わったのである。

もちろんこれだけではただのお話。彼らの成果とそれを支える議論を、きめ細かく検討することなしに、あの時代から本当に興味ある洞察を引き出すことはできない。とはいえ、これらのパラドクスや定理たちが、同じ構造をしていることもまた確か。本稿の主人公であり、数学基礎論論争の終盤に活躍したフォン・ノイマンも「自己」を巡る問いを立てている。ただし、まさに彼らしい仕方で。すなわち、

> 機械は自分自身を複製することができるか？

答えは「できる」。彼は自己複製するオートマトンを実際に構成してみせている。そして、彼が提示した「セルラー・モデル」は、生命現象を計算機上でシミュレートするという、人工生命研究における重要な手法の「元祖」である。

彼のオートマトン理論の構想は、彼自身の興味と才能の幅広さを反映して、数理論理学に留まらず広範な学問領域をカバーする、壮大なものだった。ただ、自己複製オートマトンを構成し、そこで起こる（現代の言葉で言えば）「創発」現象を理解し、説明しようとするときのフォン・ノイマンは、間違いなく、数理論理学の天才としてのフォン・ノイマンである。本稿は、彼の「自己複製オートマトンの理論」を、主に数理論理学の観点から読み解くことを目指す。この試みが、今号のテーマ、人工生命研究の理解にどれだけ寄与するか、心もとないところではある。ともあれ、「元祖」フォン・ノイマンはこんなことを考えていた、ということである。

もうひとつ。本稿の後半は長大な「付録」として、その自己複製オートマトンの設計を比較的詳しく説明してある。本編自体は、それを見ずとも理解できるように書いてあるが、興味をもたれた方は、そちらもご覧ください。

1. オートマトン理論の構想

1.1 一般的オートマトン理論

フォン・ノイマンの目標は「**オートマトンの一般的・論理的理論**」の構築である。まず「一般的」。そこには人工のオートマトンだけでなく、「自然のオートマトン」も含まれる。生物、あるいはその一部を構成する器官、特に重要なのが、神経系や脳。これらを、計算機やさまざまな機械とひとしなみに「オートマトン」と見なし、ひとまとめに研究する。であるから、オートマトンの一般理論は、論理学・数学はもちろん、生物学、有機化学、計算機工学、情報理論、さらには物理学まで、多様な領域にまたがる構想であった。とはいえ、現実的な目的はやはり、より強力な計算機の開発。自然のオートマトン、特に、非常に高度な計算機と見なせる神経系や脳を参照しながら、人工のオートマトンの可能性を探っていく。そのための「一般性」である。

残念ながらこの壮大な構想は、彼が中途で病に倒れたために未完に終わる。われわれに残されているのは、一般理論の中でも、純粋論理的（形式的）な部分の成果だけ。自己複製オートマトンの理論もその一部である。この理論に関しては、講演録と未完成手稿が残されており、彼の同僚であったバークス氏の手による編纂、補足によって、完成した形で見ることができる。本稿の記述もバークス氏の業績に負うところが大きい。

ともあれ、本稿が見ていくのは「一般的オートマトン理論」のうち、純粋論理的な部分である。次は、この部門の性格について、概観する。

1.2 公理的手法

上の「一般的理論」には、一見して困難がある。「参照する」と言っても、人間の脳や神経系については、人工のオートマトンよりもはるかに、わからないことだらけなのである。そこからいかにして、計算機開発のためのアイディアを引き出せるというのか。

この困難をある程度回避するために、フォン・ノイマンは、オートマトンの「論理的」理論から始めたのである。それは、数理論理学で扱える形式的側面にまず集中するということではあるが、それ以上の意味も持つ。「論理的」とはこの場合、彼の言葉で言えば「**公理的手法**」(von Neumann, 1951, p.289) を用いる、ということでもあるのだ。もちろん、この「公理的 (axiomatic)」には、フォン・ノイマンの、ヒルベルト門下生としての主張が込められている。

公理化されるのは、オートマトンを構成する要素のふるまいである。**公理化は本質的にボトムアップ式**であることに注意しよう。数学理論の公理系は、自らがいかなる理論であるか、マクロ的に述べているわけではない。それが定めているのは、基本的な要素（原初表現ないし無定義語）の性質と、それらの適正な組み合わせ方。前者が公理、後者が推

論規則に当たるだろう。それが「よい」公理系であるのはもちろん、矛盾することなく、ターゲットの数学理論全体が導出できるときである。つまり、オートマトンの要素のふるまいをうまく公理化して、そこからの組み合わせにより、神経系や脳が見せるような高度な機能を実現する。そのための要素の組織方法について、何らかの洞察を得ること。これが目標である。「分析ではなく総合」という、現代のシミュレーションの精神は、彼の中に、というよりも、数学の公理化というアイディアの中に、すでに内蔵されているのである。

では、公理的手法にはどういう利点があるのだろうか。二点挙げる。

■ブラック・ボックス化　公理化は、フォン・ノイマンにとって、要素のふるまいだけを記述し、そのふるまいが何によって実現されているかは不問に付す、ということを意味する。つまり、オートマトンを構成する原子的な要素は、関数的なふるまいをする「ブラック・ボックス」として扱われる (von Neumann, 1951, pp.289-290)。

このアイディアの利点は、例えば「細胞あるいは機械の部品にあるふるまいをさせるためには、その細胞ないし部品はどのような材質でなければならないか」といった問題の、経験諸科学による解決を待つ必要がない、というところである。彼はそういった研究の重要性を否定しているのではない。しかし、公理的手法をとることで、それらとはある程度独立に、当面の「要素の組織化」の問題に取り組むことができる。

■数学の数学化　数学基礎論論争の背後にあって、そもそもそのような論争を可能にしたのは、「数学理論は形式化できる」という洞察であり、その洞察をもとに作り上げられた形式論理学。「形式化」とは、ここで言う「公理化」であり、それはすなわち「数学化」。数学の理論自身が、数学的手法で研究されることになったのである。

その結果わかったのは、数や図形など、お馴染みの数学的存在者たちではなく、他でもない数学の理論自身が、興味深い数学的構造をもつ、ということ。そして、フォン・ノイマンの自己複製オートマトンの理論もやはり、この洞察の線上にあるものとして理解できるのである。

2. 複雑性仮説

2.1 複雑さとその逆説性

では、なぜ「自己複製」が中心問題となるのだろうか。言い換えれば、それに答えることが、オートマトン理論の構築に対してどのような寄与をなすのだろうか。

フォン・ノイマンが一貫して注目するのは、「複雑さ」という概念である。求めているのは、「強力な計算機を作りたい」。強力さは、計算の速さもさることながら、やれる仕事

の複雑さとも密接に関わる。つまり、計算機にさまざまな複雑な仕事をやらせたい。さてこのとき、計算機自体も、実行する仕事と同じくらい複雑でなければならないのだろうか？あるいはどれくらい複雑でなければならないのだろうか？もちろんこの問いとそれに対する答えは、「複雑さ」の意味に依存する。上の問いをしっかりと定式化すること自体が、オートマトン理論にとって重要な課題である。

彼のアプローチは、天下り式の定義ではなく、あくまで事例ベース。さまざまなオートマトンとその仕事の関係を考え、そこから正確・厳密な「複雑さ」概念を抽象することを目指す。ただ、そこで取り上げる事例は「重大な事例、すなわち、複雑さというものの重大かつ逆説的な性質を示す」(von Neumann, 1966, p.80 邦訳, 97 頁[1]) ものでなければならない。「重大な」はよいだろう。しかし「逆説的」とはどういうことか。フォン・ノイマンは、複雑さはある種の逆説性を持っており、それが複雑さの重要な性質（のひとつ）だと考えていた。そして、ほかならぬ自己複製オートマトンこそ、その「重大かつ逆説的」な性質を示す事例なのである。

2.2 複製のディレンマと複雑性仮説

次のような「ディレンマ」を考える (von Neumann, 1966, pp.78-79, 邦訳, 94-96 頁)。

(a) 自然のオートマトン、つまり生物は自己複製できる。さらに、進化を通じて、自分より複雑なものを生み出している。［(親の複雑さ) ≤ (子の複雑さ)］
(b) 人工のオートマトン、つまり機械は、自分より単純なものしか作れないように見える。［(親の複雑さ) > (子の複雑さ)］

(a) はよいだろう。(b) が問題である。複雑精巧にできた工作機械が、ある単純な物を作る、これは当たり前。では、より複雑なものを作れるだろうか。これは一見したところ、不可能である。なぜか。機械 A が他の機械 B を作る能力を持っているとする。そのとき A は、B の設計図を内に含んでおり、かつ、それをもとに実際に組み立てを行うための機構をも持っているはずだ。ここで、B の設計図の複雑さと B そのものの複雑さを同一視するなら、

A の複雑さ = B の複雑さ＋組み立て機構の複雑さ

となり、必然的に親 A は子 B よりも複雑であるはずなのである。

このように、オートマトン一般から見れば、相反する二つの主張が成り立っている。しかし、自然・人工のオートマトンをひとまとめにして理論化しようとするなら、この (a) と (b) のギャップは埋められるべきである。

自己複製オートマトンは、この (b) に対する有無を言わさぬ反例になる。つまり、機械

は自分と同じだけの複雑さを持つ機械を作ることができる。ここでのポイントは、この反例は、複雑さの定義に依存しないということだ。コピーなのだから、どう考えても複雑さは同じ、というわけである。さらに、自己複製オートマトンは、万能組み立てオートマトン、すなわち、設計図が与えられれば、どんなオートマトンでも組み立てることができる。そのようにして、(b) の不等号は「>」から「=」、そして「≤」へ書き換えることができるのである。

つまり、一見正しそうに見えた (b) は実は正しくない。ここに、フォン・ノイマンは複雑さの「逆説性」を見てとる。もちろんこれは、矛盾でも何でもなく、フォン・ノイマンが引き出すのは、次のような考えである。

> こうして複雑さというものには、完全に決定的 (decisive) な性質があることになります。すなわち、ある臨界的 (critical) なサイズ、それ以下のサイズのところでは、合成 (synthesis) プロセスは退化的 (degenerative) ですが、それを超えたところでは、うまくアレンジしてやれば、合成の現象は爆発的になる、そういったサイズが存在するのです。(von Neumann, 1966, p.80, 邦訳, 97 頁)

ある程度単純なものは、それよりも単純なことしかできない（退化的）、しかし、**複雑さがある一定の程度を超えると、機械は自分よりも複雑なことをやれるようになる**。これを本稿では「**複雑性仮説**」と呼ぶことにしよう。もちろんこれは仮説ないし予想であって、確証や証明がなされたわけでも、複雑さの厳密な定義が得られたわけでもない。ただ、自己複製という例が、複雑さの「逆説的」な性質を際立たせる事例である、ということはお分かりいただけるはずである。

ところで、(b) に対しては明確な反例を提出しうるわけだが、それでは、(b) を構成する議論のどこが間違っていたのだろうか。これは言い換えれば、自己複製オートマトンの事例をどのように理解すれば、複雑性仮説が出てくるのか、ということでもある。ポイントは、オートマトンの「万能性」とそれを可能にする「設計図」。予想できるように、この発想は、万能チューリング・マシンに由来しており、複雑性仮説も大部分、チューリング・マシンの観点から説明できる。自己複製オートマトンの存在は、その議論の最後の部分を補強する役割を果たす。

このようなわけで、次節では、まずチューリング・マシン、次に自己複製オートマトンを使って、複雑性仮説を理解することを目指す。そして最後に、この仮説を背後で支えているのは、前節で見た「公理的手法」であることを示したい。

3. 万能性から自己複製へ

3.1 チューリング・マシン：万能性と停止問題

複雑性仮説の言う「複雑さの爆発」を、フォン・ノイマンは、ゲーデルとチューリングを持ち出して説明しようとする。（第一）不完全性定理と、万能チューリング・マシンである。ただ、彼の説明は示唆的なものに留まっているので、それをなるべく丁寧に、私なりに再構成してみたい (cf. von Neumann, 1966, pp.49-56, 邦訳, 60-62 頁[2])。またここでは、不完全性定理の代わりに、その兄弟分とも言える、これまたチューリングの「停止問題」を使うことにする。

■TM とそのふるまい　まず、チューリング・マシン（TM と略す）そのものと、TM のふるまいを分けて考えよう。TM は、抽象的な計算機モデルだが、その基本動作はヘッド（あるいはテープ）を動かし、テープ上の記号を読み書きし、停止する、ということに尽きる。時点 $t+1$ でどの動作をするかは、時点 t での内部状態と、テープで読み取った記号から決定される。それゆえ、個々の TM は、その動作の段取り、すなわちプログラム（まずヘッドを右に、次に記号が 1 ならそれを消せ、等々）によって記述することができる。プログラムは、命令の**有限列**である。その命令の種類ももちろん有限。プログラムは TM の記述ではあるが、TM そのものと見なしてもよい。つまり、TM とは有限の命令列である。

次に、TM の「ふるまい」を考えてみよう。TM が行うのは、動作開始時のテープ上の記号列、すなわち入力を読み取り、テープを書き換えつつ、うまく行けば最後に停止する、という動作。この一連の動作を、プログラムに従って行うのである。停止したときにテープ上に残された記号列が出力。しかし、場合によってはループに陥って、いつまでも停止しないこともある。そこで、入力と「結果」の対を全部集めよう。その中には、$\langle a, b\rangle$（これは出力された場合）という形のものもあれば、$\langle c,$ 停止しない$\rangle$ というものもあるだろう。ともかく、TM M に対して、この対応の総体を「M のふるまい」と考えることにしよう。

可能な入力は可算無限個考えられるので、M のふるまいも無限集合である。すると、有限の命令列から、無限集合が生まれたことになる。しかし、これを「複雑さの爆発」とは言えない。というのは、M のふるまいの記述が考えられるからだ。極端な例を挙げよう。すべての入力に対して、その記号列を全部消して、停止する TM N を考える。ふるまいは無限集合だが、その記述は 1 行で済むだろう。ところで、TM の記述（プログラム）を TM そのものと見なしているのだった。N のプログラムはおそらく 1 行ではない。もちろん、二つの「1 行」は簡単に比較可能なものではないかもしれない。しかしこのことは、記述レベルで考えるなら、一般に TM そのものより TM のふるまいが（爆発的に）複雑

だとは言い切れない、ということを示すのに十分だろう。「臨界点」はここではないのである[3]。

■万能 TM 「臨界点」は万能チューリング・マシン（**UTM** と略す）である[4]。最重要ポイント：**任意の TM は、自然数でコード化できる**。TM は命令の有限列であり、命令の種類は有限であるから、可能である。その方法は省略するが、ともあれ、各 TM に対し、一意的な番号を割り当てることができる。そして自然数なら、テープ上に表現可能である。任意の TM M に対し、その自然数コードのテープ上の表現を、$\#M$ と書くことにしよう。

事実 1：次のような TM **UTM** が存在する。すなわち、任意の入力 (m, n) に対し、

(1) m がある TM M のコードであるとき：
- M が入力 n に対し、k を出力するとき：**UTM** は k を出力する

(2) それ以外のとき：**UTM** は停止しない ⊠

UTM に、ある M のコード $\#M$ を与えてやれば、**UTM** はそれを**解釈**し、その後、もうひとつの任意の入力に対し、M と同じふるまいを生み出す。M が出力するときは、その出力を出し、M が停止しないときは停止しない。結局、**UTM** は任意の TM のふるまいを生み出すことができる。この事実は、次のようなことを教えてくれる。

重要なのは、コードを解釈し、そこから元の TM の動作をシミュレートできることである。そのような機能を実現するために、**UTM** はある程度の複雑さを持たねばならない。しかし、**UTM** もあくまで有限の命令列である。それゆえ、命令列の複雑さの基準として何を採用しようと、おそらく **UTM** より複雑な TM が考えられる。しかし、**UTM** は万能。自分より複雑な TM のふるまいでも生み出すことができる。するとここでも、ふるまいの複雑さの基準に依存せず、**UTM** のふるまいは、どんな TM のふるまいよりも複雑だ、と言えそうである。

教えてくれないこともある。確かに **UTM** はすごいのだが、目標は「ある程度の複雑さをもつ機械は、自分よりも複雑な仕事ができる」であった。今わかっているのは、

- **UTM** より複雑な TM が（おそらく）存在する
- **UTM** のふるまいは、どんな TM のふるまいよりも複雑である

という、TM 同士、ふるまい同士の複雑さの関係だけなのである。**UTM** のふるまいが、**UTM** 自身よりも複雑だ、とはまだ言えない。それを言うためには、つまり **UTM** が複雑さの臨界点だと言うためには、停止問題を持ち出す必要がある。

■停止問題 「ふるまいの記述」について考えよう。**UTM** は、TM すべてのふるまいの総体を生み出す。この仕事はいったいどういう仕事だったのだろうか？単純化して言えば、

任意の TM とそれに対する入力に応じて、TM が停止するときは停止する、停止しないときは停止しない、という仕事だ。これを TM のプログラム言語で記述できるだろうか。できないのである。

事実 2：任意の TM M とそれに対する入力 z に対し、

- M が z に対し、有限回の動作の後停止するとき：0 を出力
- M が z に対し、停止しないとき：1 を出力

するような TM は存在しない。 ⊠

UTM そのものは、有限の命令列によって記述できる。しかし、そのふるまい、その仕事の内容は、有限の（TM に対するプログラム言語の）命令列によっては記述できない。「できることなら何でもできるような器官は作れるが、できるかどうかを答えてくれる器官は作れない」(von Neumann, 1966, p.51, 邦訳, 62 頁) のである。この言語による記述可能性を、ひとつの複雑さの基準と見なせば、**UTM** は自分よりも複雑な仕事をしていると言えるのである。

命令をひとつの部品と見なすなら、こうして、それらの部品をある程度の複雑さで組み合わせることにより、複雑さは臨界点に達し、機械は「自分よりも複雑な」仕事をするようになる。「実行する方が記述するよりも早い」(von Neumann, 1966, p.47, 邦訳, 57 頁) 状況を作り出せるのである。

3.2 なぜ自己複製？

ここまで、TM の観点から「複雑さの爆発」を見てきた。結構うまく説明できているはずである。しかしではなぜ、フォン・ノイマンはそれに加えて、自己複製オートマトンを作ったのか？もちろん、自己複製という自然現象を、セルラー・モデル上でシミュレートするというのは、それ自身興味深いことではある。だが、自己複製オートマトンは、そこから彼が引き出したかった「複雑性仮説」に対してもやはり、一定の役割を果たしているのである。

まず、TM の複雑さに関する議論に対して、次のような疑念が生じうる。確かに **UTM** は万能である。しかしその能力も、何らかの TM のコードが入力されてはじめて発揮されるもの。だとすれば、**UTM** のふるまいの複雑さは、**UTM** 自身の複雑さではなく、コード＝元の TM の複雑さに由来するのではないか。つまり、

TM M のふるまいの複雑さ $\leq$ **UTM** のふるまいの複雑さ

という不等式は単純には成り立たないのではないか。

これは、先の (b) の場合と同様の議論である。自己複製オートマトン（**SRA** と略す）は、この議論が成り立たないこと、すなわち、万能マシンのふるまいの複雑さは、そこで利用される情報の複雑さには依存していないことを示してくれる。「複雑さの爆発」を起こす本質的な要因はそこではない、と教えてくれるのである。

3.3 自己複製オートマトンの概要

というわけで、自己複製オートマトンを見ることにしよう。詳しいことは付録を見ていただくとして、ここでは、ごく簡単な発想、図式だけ説明する (cf. von Neumann, 1966, pp.84-87, 邦訳, pp.102-106)。

自己複製オートマトン **SRA** は、抽象的・形式的・数学的な工作機械のモデルである。その構造、工作作業の様子、作業の結果できたもの、これらはすべて**セルラー・モデル**と呼ばれる無限の広さの平面上で表現される。その平面をセル平面と呼ぼう。

セル平面上で表現可能な、任意のオートマトン X を考える。TM のときと事情は同じ。ひとつのプログラム（命令列）で記述することもできる。それゆえ、コード化が可能。その設計情報を $\#X$ で表わそう。そしてそのコードを、これまたセル平面上で表現される「テープ」の上に記載することもできるのである。

先に述べたように、**SRA** は万能組み立てオートマトンでもある。というよりも、それを自分の部品として含んでいる。その万能組み立てオートマトンを A で表わそう。A は、テープを読み取る機構を備えており、任意のオートマトン X に対して、そのコード $\#X$ をテープ上に記載された入力として受けとると、それをもとに、セル平面上のどこかに、X を組み立てることができる。このことを、

$$A + \#X \longrightarrow X$$

と表わすことができるだろう。ここで、テープ $\#X$ は、A の部分ではなく、A に与えられる付加的な情報であることに注意しよう。それゆえ、単純に X に代入すると、確かに

$$A + \#A \longrightarrow A$$

となり、A は出来上がるが、これでは自己複製とは言いにくいかもしれない。A という複雑なオートマトンを組み立てられたのは、$\#A$ という、まさに A と同じ複雑さをもった情報のおかげではないのか。上の式を見れば、先の (b) とまさに同じ状況になっているのである。

しかし心配は要らない。まずコピー・オートマトン B を考えよう。これはテープを見て、それをそのまま好きな場所に複製できる機械である。さらに B を A と接続して、そ

れらを制御する機械 C を考えよう。万能組み立てオートマトンが構成できるくらいなのだから、B や C も構成できることは、明らかだろう。このオートマトン $A+B+C$ 全体は、次のように動作する。

1. A のテープ差込口に、X のテープ $\#X$ を差し込む。
2. X を組み立てるのは後にして、C はまず B にそのテープの複製を作らせる。
3. 次に C は、A に最初のテープを元に、実際に X を組み立てさせる。
4. 最後に、完成した X と、複製されたテープ $\#X$ をうまく結びつける。

この手順の結果できるのは、$X+\#X$ である。この X に、$A+B+C$ を代入しよう。

$$A+B+C+\#(A+B+C) \longrightarrow A+B+C+\#(A+B+C)$$

が成り立つはずである。どう見ても自己複製である。しかも、テープ $\#(A+B+C)$ まで複製できているから、この作業は、$\#(A+B+C)$ の複雑さに依存していない、というわけである。このようなわけで、$A+B+C$ こそが、何でも作れる、そしてそれゆえに自分も作れる、自己複製オートマトン **SRA** の正体である。

3.4 自己複製が教えてくれること

では、この **SRA** が、複雑性仮説の議論にどう役立つか、考えてみよう。まず話を、セルラー・モデル上で構成される組み立てオートマトン CA [5] に限ろう。CA たちも TM たちと同じく、ある種のプログラムで記述できるから、プログラムと CA それ自身を同視しよう。もちろん、プログラムはコード化して、セル平面上のテープに表現できる。CA のふるまいは、入力とそれに従い組み立てられたオートマトンの対の集まり、と考えることができる。

さて、**SRA** は万能組み立てオートマトンでもあったから、そのふるまいは、どんな CA のそれよりも複雑だと言えるだろう。しかも、**SRA** は組み立ての過程で、自らに差し込まれたテープも複製する。このとき、**SRA** のふるまいの一部、すなわち組み立てられたものの複雑さに、たとえ上の疑念の通り、テープの複雑さが寄与していたとしても、その複雑さは丸ごと、作られたものに含まれている。これはつまり、言い換えれば、テープの複雑さは、**SRA** のふるまいの複雑さに本質的な寄与はしていない、ということである。

このようにセルラー・モデルは、親オートマトン自身とそのふるまい（子オートマトン）が、同じ平面上に並ぶため、両者の複雑さを比較するのに便利である。自己複製が生じたと見なせるのも、このオートマトンとそのふるまいの同質性による。

しかし、もっと重要なことがある。この議論が教えてくれるのは、**UTM** や **SRA** の万能性、すなわちそれらのふるまいの（高度な）複雑さは、コードの複雑さではなく、**UTM**

や **SRA** それ自身の複雑さに由来するということ。コード自身は、万能性の観点からすれば、容易に複製できる程度の複雑さしかもっていない。コードではなく、コードを解釈し実行する、その能力を実現するための複雑さが決定的なのである。それだけの複雑さがなければ退化的になり、オートマトンは貧弱な仕事しかできないだろう。しかし、その複雑さにまで達すると、爆発的な効果、すなわち万能性を得ることができるのである。

さらにもうひとつ遡ろう。コードの解釈と実行による「複雑さの爆発」を可能にしたのは、「オートマトンのコード化可能性」。コードは、ふるまいの複雑さを内に含んではいるが、それ自体は、言わば手軽に扱うことができる。つまり、機械が扱うテープに載せることができる。その手軽さを利用するのが万能 TM であり、また、自己複製の議論にとって決定的な、テープの複製という作業だった。

そしてコード化が可能なのは、オートマトンは有限の命令列で記述できるから、つまり、公理的手法をとったからに他ならない。TM のなす仕事は、TM を記述する言語では記述できず、別のより強力な言語を必要とする。しかし、強力な言語を用意せずとも、TM 自身を記述してやりさえすれば、記述に必要な言語だけで、それをある意味で超える仕事を、すべて行うことができる。これが公理的手法の利点なのである[6]。

機械を公理化することでコード化を可能にし、その機械自身を付加的情報として、別の機械の作用する場所に埋め込んでしまう。この手法は、数学の理論自体の数学的構造を研究する、つまり数学を数学自身で研究する、数理論理学の手法の典型的な適用例である。フォン・ノイマンならではの、視野の広い「自己」問題。これもやはり、「自己」を巡る数理論理学の問題として見ることができる。冒頭に書いた、冗談のようなお話は、あながち冗談とも言えないのである。

おわりに

セルラー・モデルといえば、計算機による試行錯誤。さまざまな初期配置・条件を考え、それを計算機上で実行してみることで、分析（解析）的にはメカニズムがわからないふるまいを、総合的に作り上げることを目指す。失敗すれば、また別の条件。大量のルーティンを高速でこなしてくれる、計算機の進歩が、その試行錯誤を可能にした。

だが面白いことに、フォン・ノイマンは分析的に自己複製オートマトンを作ってしまったのである (Burks, 1970b, p.61)。使ったのは、彼自身による推論といくつかの場合の手計算だけ。「さすが」である。しかしその後、例えばコッドは、計算機を使った試行錯誤（効率のよい戦略を用いて）により、計算的にも組み立ての観点からも万能な、8 状態のセルラー・モデルを作ることに成功している。フォン・ノイマンの場合は 29 状態だったから、かなりの節減である。計算機も「さすが」である（もちろんコッド氏も）。とはい

え、計算機の開発に大きく貢献し、計算機を用いたこの手法を提案したのも、やっぱりフォン・ノイマン。彼の「万能」ぶりをよく物語るエピソードではある。

付録：セルラー・モデルと自己複製オートマトン

セルラー・モデルとは、後で見るようにチューリング・マシンと同じく、抽象的な計算モデルである。その中で、自己複製オートマトンが構成される。セルラー・モデルを定義した後、そこで自己複製がどのように行われるかを見ていくことにする。なお、このモデルについての基本文献は、von Neumann(1966) および Burks(1970b) だが、以下の説明は主に Thatcher(1964) に基づく。

I. セルラー・モデル

無限に広い将棋盤を考える。方眼紙でもよい。ともかくマス目が無限に並んでいる。そしてすべてのマスの上に、ある条件を満たす有限オートマトンを置いてやる。加算無限個の有限オートマトンが将棋盤の上にずらっと並べられたもの、これが**セルラー・モデル**である。つまり、ひとつひとつの有限オートマトンが**セル**である。どこかのマスを原点 $(0,0)$ と定めてやれば、それと相対的に、すべてのマスの位置は整数の対で表わすことができ、位置を指定してやることで、そこにあるセルを参照できる。この平面を**セル平面**と呼ぶことにしよう。

有限オートマトンとは、入力に従って自身の状態を変化させていく機械（のモデル）である。ある時点 $t+1$ における有限オートマトンの状態は、時点 t における入力と自身の状態から、遷移関数により決定される。つまり、ひとつの有限オートマトン M は、$\langle Q, \Sigma, \delta \rangle$ という三つ組、ただし、Q は**状態**の有限集合、Σ は**入力**の有限集合、δ は**遷移関数**であり、その型は $Q \times \Sigma \to Q$ であるが、これらを決めてやることで確定される。

セルラー・モデルを構成する有限オートマトン、セルの満たすべき条件は、もちろんそのモデルの作者によって違うのだが、ここではフォン・ノイマンのそれに従おう。

まず、各セル a に対し、上下左右の隣接セルを、a の**近傍**と呼ぼう。セルラー・モデルの特徴は、**セル（有限オートマトン）への入力として、その近傍のセルの状態をとる**というところにある。すなわち、上の Σ は、ここでは Q^4 と表わすことができ、遷移関数 δ の型は、$Q \times Q^4 \to Q$ となる。セル a の時点 $t+1$ における状態は、時点 t における a とその 4 近傍の状態から、遷移関数に従って確定する。ただし、状態集合 Q の中に、少なく

ともひとつの特別な状態 q_0 が指定されねばならず、この q_0 に対し、遷移関数 δ は、

$$\delta(q_0, (q_0, q_0, q_0, q_0)) = q_0$$

という条件を満たさねばならない。このような q_0 を**静穏状態**と呼ぶ。この条件が意味しているのは、自分も周りも静穏なら次の時点でも静穏、つまり、セルラー・モデルの中の「静かな一帯」は、外からの働きかけがない限り、静かなままで留まる、ということである。

このような一定の条件を満たす有限オートマトンをひとつ定めてやることで、セルラー・モデルがひとつ定まる。しかし、セル平面は言わば画面（セルはドット）のようなものであって、セルラー・モデルを定めただけでは、そこに何が映し出されるかは決まらない。何かを映し出し、それを動かす（アニメーションを描く）ためには、個々のセルの初期状態を決めてやる必要がある。それを決めることで、遷移関数の性質から明らかなように、セル同士が相互作用し、状態を変えていき、セル平面全体の状況も変わっていく。

いま、セル $\mathbf{M} = \langle Q, \delta, q_0 \rangle$ により定められる、あるセルラー・モデル **CM** を考える。各セルがいかなる状態 $(\in Q)$ にあるかを **CM** の**配置**と呼ぶ。**CM** のひとつの配置 s とは、

$$s : \mathbb{Z} \times \mathbb{Z} \to Q, \quad \text{ただし, } s^{-1}[Q \setminus \{q_0\}] \subset_{\text{fin}} \mathbb{Z} \times \mathbb{Z}.$$

なる関数である。ひとつひとつのマスにひとつの状態を割り当てるが、静穏状態ではないマス（セル）の数は、常に有限であるものとする、ということである。ひとつ配置を決めれば、セルの遷移関数の性質から、次の時点の配置が決まる。

さて、配置 s は、真っ白な（静穏状態ばかりの）無限の広さの画面の中に描かれた（状態を色と考えるなら、色つきの）有限の図形と考えてよい。あるいは、ひとつだけではなく、画面上にいくつかの図形が、互いに距離をとって映っていることもありうる。このような図形について考えるには、その図形を含む領域 $U \subset \mathbb{Z} \times \mathbb{Z}$ を適当にとり、配置 s（これは関数である）の制限、$s|U = \{\langle \alpha, s(\alpha) \rangle \mid \alpha \in U\}$ における図形を考えればよい。

これらの図形は時間の経過に従って形を変えるかもしれないし、さまざまな運動をするかもしれない。消え去ってしまうかもしれない。そこで、図形をひとつのオートマトン、機械と考えてよいだろう。図形の変化により、セルラー・モデルは、画面上に図形、すなわちオートマトンを描画し、その変化によって計算その他、オートマトンがやる仕事を表現するわけである。

さて、本稿のテーマである自己複製はどのように表現されるか。厳密な定義も可能だが、ここでは、非形式的な形で考え方だけを説明すれば十分だろう。後で見るフォン・ノイマンのオートマトンは、どう考えても自己複製をしているように見えると思われるからである。

まずは組み立てから。あるセルラー・モデル **CM** と **CM** に対するある配置 s を考える。最初はひとつの図形だけと考えよう。つまり画面上に、ひとつのオートマトン A が映し出されている、と考える。このとき、A が別のオートマトン B を組み立てた、と言えるためには、

(i) 配置 s から出発して、ある時点 t において、
(ii) A からある程度の広さの「静穏領域」で隔てられた領域に、図形 B が現れ、
(iii) 元の図形 A はそのまま、ないしそれに近い形で残っている、

という条件がだいたい必要十分だろう。(ii) の「静穏領域、つまりセルがすべて静穏状態であるような領域が必要なのは、A と B が図形として繋がっていると、それは A が成長しただけ、となる恐れがあるからだ。同様に (iii) は、A が移動しただけと見なされないための条件である。

これが組み立てだとすると、自己複製は、B として A をとった場合ということになる。もちろん、図形をその位置（座標）込みで考えるなら、A が自己複製したと言えるのは、A とは離れた領域に図形 A' が現れ、A' と A が「同じ形」をしている場合、ということになろう。

II. フォン・ノイマンのセルラー・モデル

では、フォン・ノイマンのセルラー・モデルと、その上で構成される自己複製オートマトンについてみていくことにしよう。

■セル　まずセルラー・モデルを定める。そのためには、セルとなる有限オートマトンを定めればよい。フォン・ノイマンの考えたセルは、状態数 29。それらは大きく言えば 5 種類に分かれる。表 1 を見ていただきたい。これら各状態の機能が遷移関数により決定される。いくつかのポイントだけ説明する。

- **U** が、平面上の無限の部分を占める静穏状態である。
- 意味のある仕事をするオートマトンを構成するには、主に、通常・特殊伝達状態と合流状態でひとつの配置を作ることになる。
  - そこで表現されるのは、矢印記号が示唆している通り、デジタル信号がオートマトンのある部分から他の部分へ伝達される様子である。
  - 伝達状態の矢印に点が付いていないときは、信号が伝わっていない。つまり、信号の通り道だけがある状態。点が付いているのは、その通り道をその時点に信号が伝わっている状態である。合流状態は、分岐を作る。

- 潜像状態については、次の項で見ることにしよう。

興奮不能	$\mathbf{U}$			
通常伝達　（非興奮）	$\rightarrow$	$\uparrow$	$\leftarrow$	$\downarrow$
（興奮）	$\dot{\rightarrow}$	$\cdot\uparrow$	$\dot{\leftarrow}$	$\cdot\downarrow$
特殊伝達　（非興奮）	$\Rightarrow$	$\Uparrow$	$\Leftarrow$	$\Downarrow$
（興奮）	$\dot{\Rightarrow}$	$\cdot\Uparrow$	$\dot{\Leftarrow}$	$\cdot\Downarrow$
合流	$\mathbf{C}_{00}$	$\mathbf{C}_{01}$	$\mathbf{C}_{10}$	$\mathbf{C}_{11}$
潜像	$\mathbf{s}_{\theta}$	$\mathbf{s}_{0}$	$\mathbf{s}_{1}$	$\mathbf{s}_{00}$
	$\mathbf{s}_{01}$	$\mathbf{s}_{10}$	$\mathbf{s}_{11}$	$\mathbf{s}_{000}$

表 1　29 状態 [(Burks1970b, p.9, Figure 2) をもとに筆者が作成]

■構成と破壊　セルラー・モデルの定義どおり、各セルは、上下左右 4 近傍のセルの状態により、状態を変化させていく。例えば、ある時点 t において、セル α が状態 $\rightarrow$ にあるとし、さらにその左近傍セル β の状態が $\dot{\rightarrow}$ であったとする。他の条件がうまく整えば、次の時点 $t+1$ での α は、状態 $\dot{\rightarrow}$ になる。信号が伝わったわけだ。

さて、自己複製に向けて、重要なのは「構成と破壊」である。静穏状態 **U** を、他の状態に変化させることで、「白紙」の上に「図形」を描かねばならない。また、オートマトンの「腕」（自己複製オートマトンは腕を伸ばすのである）が自由自在に「動く」ためには、図形の「伸び縮み」が可能でなければならない。そして、もちろん構成と破壊の条件も遷移関数の定める規則に従う。

まず破壊条件の例。セル α における通常伝達状態が「破壊」されて、**U** になるのは、少なくともひとつの α の近傍セルが存在し、それが α の方を向いた、興奮した特殊伝達状態であるとき、そのときに限る。例えば、$\dot{\Rightarrow}\uparrow$ となっていれば、次の時点では右側のセルは **U** になってしまう。つまり、破壊されてしまう。

次に構成である。ここで 8 つの潜像状態が役に立つ。例えば、時点 t におけるセル α の状態が **U** だとする。そのとき、少なくともひとつの近傍セルが、α の方を向いた、興奮した通常または特殊伝達状態であるとき、$t+1$ における α の状態は $\mathbf{s}_{\theta}$ となる。つまり、$\dot{\rightarrow}$**U** なら、次の時点の右側のセルは $\mathbf{s}_{\theta}$ になる。新しく何かが構成されたというわけである。その後、α にどのような信号が送り込まれるかによって、$\mathbf{s}_{\theta}$ はさまざまな潜像状態に変化し、最終的に伝達状態や合流状態にまで変化できる。

このようにして、「破壊と構成」が可能になる。さまざまな状態を持つセルたちをうま

く組み合わせてやれば、あるひとつのセルについて、それがどんな状態であっても、そこから任意の状態へ変化させることが可能である。

■いろいろな部品たち　今度は、セルを組み合わせて、小さなオートマトンを作っていく。これらを「部品」として組み合わせて、複雑な自己複製オートマトンを作り上げる。

その部品たちは、基本的にインプットとアウトプット（に当たるもの）をもち、入力された信号（列）に対して、一定の信号（列）を出力する機械と見ることができる。例えば、下の**パルサー**。**a** をインプット、**b** をアウトプットと見る。

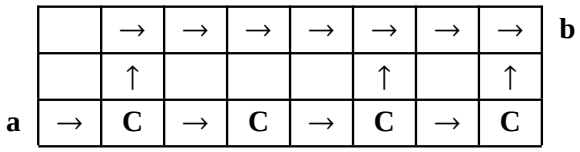

図 1　パルサー $P(1011)$ [(Thatcher, 1964, p147, Figure 2) をもとに筆者が作成]

a から入力がなされ、時点 t において左下隅の → が興奮したとする。その信号は 3 つの経路を通って伝わり、まず時点 $t+11$ にアウトプット **b** に信号が送られる。その後、時点 $t+13$ と $t+14$ にも信号が送られる。つまり、このパルサー $P(1011)$ は、11 単位時間のディレイの後、1011 という信号（列）を吐き出す機械なのである。

その他、ある一定の種類の信号列を探知して、その有無を出力する**デコーダ**、ある特定の信号列を探知する**リコグナイザ**などなど、さまざまな機械を作ることができる。それらの機械は、上の例を見てもわかるように、その構造にある程度の規則性があるため、自分の望む信号列を探知したり、出力したりする複雑な機械の構成が比較的容易である。

■テープ　このような部品の中でも、自己複製のために重要なのが、オートマトンの設計情報を記載する**テープ**である。テープの構成にはその名の通り、セルの一直線の並び（無際限に延長可能）を使う。セルひとつがテープ上のマス目である。テープ上の「記号」としては、ゼロないし空白と 1 に当たる二つの状態を用意すればよい。空白は静穏状態 **U** で、1 は通常伝達状態のいずれかで表わす。

テープの読み書きは、「ヘッド」を動かしてテープに信号を送り反応を受け取る（読み取る）、あるいはある一定の信号列を送りテープ上の状態を変化させる（書き換える）ことにより実現される。ここでテープ自身はパッシヴ、すなわち他からの刺激がない限り同じ状態に留まり続けねばならない。そうでなければ、メモリとしての役割が果たせない。また、メモリとして役立つには、それを読み書きするオートマトンと切り離されていることも必要である。それぞれの部分が意図しない相互作用を起こして変化し始めると、メモリ

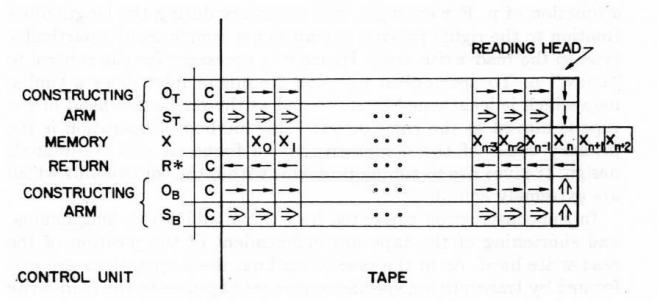

図 2　テープ部分 (Thatcher, 1964, p.170, Figure 20)

が台無しである。そのため、オートマトンの組み立てと同様、テープの読み書きもオートマトンが「腕」を伸ばすことで行う。図 2 は、テープを読み書きする部分の構造である。

III. 自己複製へ

ようやく、自己複製オートマトンが構成できそうである。先述したとおり、実際のところフォン・ノイマンが構成したのは、万能組み立てオートマトンである。作りたいオートマトン（これはセル平面上の図形である）の設計図さえ与えられれば、どんなものでも、つまり自分自身も、また、自分より複雑なオートマトンも組み立てることができる。

万能組み立てオートマトンができればあとは簡単。本編 3.3 節を見ていただきたい。というわけで、万能組み立てオートマトンの構成を見ていこう。

■設計図　万能性を持たせるためには、オートマトンの設計情報を、テープ上に表現せねばならない。その設計情報は、ある種のプログラム言語によって、命令の列として表現することができる。組み立ては、本体から伸びる**組み立てアーム**によってなされるので、命令は、そのアーム（のヘッド）に対するものとなる。本質的には次の 2 種類の命令で十分である。

- ヘッドを上（or 下・左・右）に 1 マス動かしなさい。
- ヘッドのあるマスの状態を、*v* にしなさい。

これらの命令を組み合わせれば、アームを好きな場所まで伸ばして、オートマトンを（例えば）右上隅から組み立てるための命令列ができる。このような命令列をゼロと 1 の列でコーディングするのは、チューリング・マシンなどの場合と同様であるから、難しくない。その 01 列をセル平面上のテープで表現し、組み立てオートマトンに与えられる設計図として用いよう。

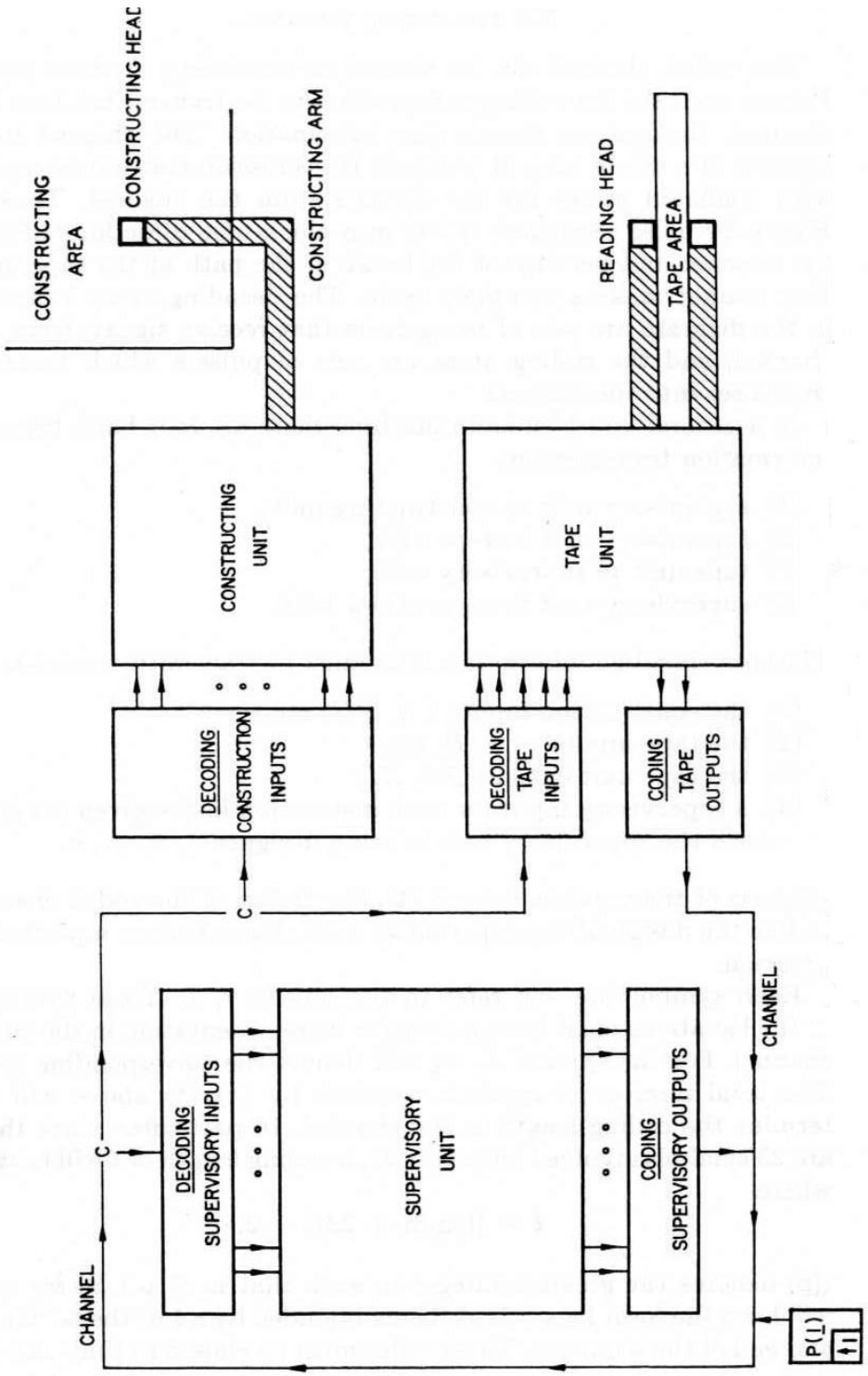

図 3　万能組み立てオートマトンの構造 (Thatcher, 1964, p.177, Figure 24)

この命令の実行、あるいは先ほどのテープの読み書きは、明らかにチューリング・マシンの動作と同様のものであり、実際その動作は、セル平面上のオートマトンで実現することができる。つまり、セルラー・モデル上での計算の定義を適切にしてやれば、万能チューリング・マシン（と同じ計算能力を持ったオートマトン）も構成できるのである。セルラー・モデルの場合、難しいのは、ヘッドの移動やテープの読み書きといった、チューリング・マシンでは前提されている機能をも実装する必要があるところ。そして、それは大変なのだが、可能なのである。

■注意　最後にひとつだけ注意。組み立てられるオートマトンはパッシヴでなければならない。つまり、組み立てていく中で、**U** 以外の状態のセルが増えていくのだが、それらが途中で動き出してしまうと、組み立てと見なすのは難しくなるかもしれないし、わけがわからなくなる。

そこで、組み立て途中のオートマトンは、→ や ⇒ などの非興奮の状態からなるものとする。とはいえ、組み立て終わったら動かしたくなるのが人情。完成したオートマトンをスタートさせるためのセルの構成も、命令の中に付け加えておくことにする。

■万能組み立てオートマトン　ようやく万能組み立てオートマトンにたどり着いた。その構造の概要は、図 3 で確認していただきたい。

大きく言えば、三つの部品からなる。ひとつは、テープを読み書きする部分。ここにオートマトンの設計情報を記載したテープが差し込まれる。もうひとつは、オートマトンを実際に組み立てる機構。この二つからはそれぞれ、読み書きアームと組み立てアームが伸び、必要な作業を行えるようになっている。そして最後に、テープの情報を解釈し、それを組み立て機構に伝える、それらの作業を制御する部分。また、左下隅にこの組み立てオートマトンをスタートさせるスイッチが付いている。新しくオートマトンを組み立てる際にも、このような部品をどこかに付けておき、組み立て後にスタートさせるわけである。

これらの部品がどのように作られているか、もはや説明する余裕はない。あとは、3.3 節で見たように、テープを複製する作業も加えて、疑問の余地のない自己複製を行えばよい。

註

*takuro.onishi@gmail.com

[1] von Neumann(1966) については、邦訳の頁数も合わせて示す。引用部分では邦訳を参考にさせていただいたが、文脈によって、筆者がかなり改変している場合もあることをお断りしておく。

[2] この箇所にはバークスの補足が付いている。バークスはチューリング・マシンなどいくつか基本的な事項を紹介したあと、この箇所の内容についてのゲーデルとの書簡のやりとりを公開している。以下の再構成には、バークス、ゲーデルの意見も参考にした。ただし、再構成は筆者自身の責任によるものである。

[3] 命題論理と一階述語論理の違いが参考になるだろう。「任意の文に対して、それがその公理系の定理であるか否か、機械的に決定できるか」という問題に対して、前者は「決定可能」、後者は「決定不能」。（述語論理の決定不能性は、後で見る停止問題の解決不可能性の系である。）一般に、公理系からは無限個の定理が導出できる。しかし、その無限集合の複雑さは一様ではなく，公理の複雑さに依存して変わりうるのである。

[4] 以下のチューリング・マシンについての諸結果は、高橋 (1991)、Thatcher(1965) を参考にした。

[5] 「組み立てオートマトン」という概念は、ここで話を簡単にするために導入しただけで、数学的に厳密に定義されたものを考えているわけではない。Thatcher(1964) は、セル平面上での「組み立て」を厳密に定義しようとしているが、「直観的に組み立てと見えるもの」とそうでないものを、うまくすくう定義は難しそうである。ただ、ここでの論旨に影響はないだろう。

[6] 興味深いのは、フォン・ノイマンは以上のことを「利点」と捉えていることである。停止問題や不完全性定理は、しばしば、機械や公理的手法（つまりヒルベルト・プログラム）の限界を示すもの、場合によっては数学や知性の限界を示すもの、とさえ言われる。このようなネガティヴな捉え方と比べると、オートマトンの設計という実際的な文脈に置かれたものとはいえ、フォン・ノイマンのポジティヴさはひときわ目を引く。本稿はもはやその余裕とてないが、これは探求に値する対比かと思われる。

文献

Burks, A. W. (Ed.). (1970a). *Essays on Cellular Automata*. University of Illinois Press.

Burks, A. W. (1970b). ‘Von Neumann’s self-reproducing automata’. In *Essays on Cellular Automata* (pp. 3–64).

Thatcher, J. W. (1964). ‘Universality in the von Neumann’s cellular model’. In *Essays on Cellular Automata* (pp. 132–186).

Thatcher, J. W. (1965). ‘Self-describing Turing machines and self-reproducing cellular automata’. In *Essays on Cellular Automata* (pp. 103–131).

von Neumann, J. (1951). ‘The general and logical theory of automata’. In A. H. Taub (Ed.), *John von Neumann Collected Works* (Vol. V, pp. 288–326). Pergamon Press.

von Neumann, J. (1966). *Theory of Self-Reproducing Automata* (A. W. Burks, Ed.). University of Illinois Press. (1975，高橋英俊監訳,『自己増殖オートマトンの理論』，岩波書店．)

高橋正子. (1991). 『**計算論：計算可能性とラムダ計算**』. 近代科学社.

〔哲学博士課程〕