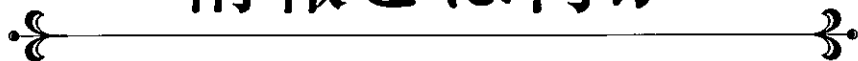


情報とは何か





森 洋久 著

国際日本文化研究センター准教授

総合研究大学院大学・学融合推進センター研究費助成プロジェクト

「日本における諸科学の編成と基礎概念の検討
— 分離統合の有効性をさぐる」

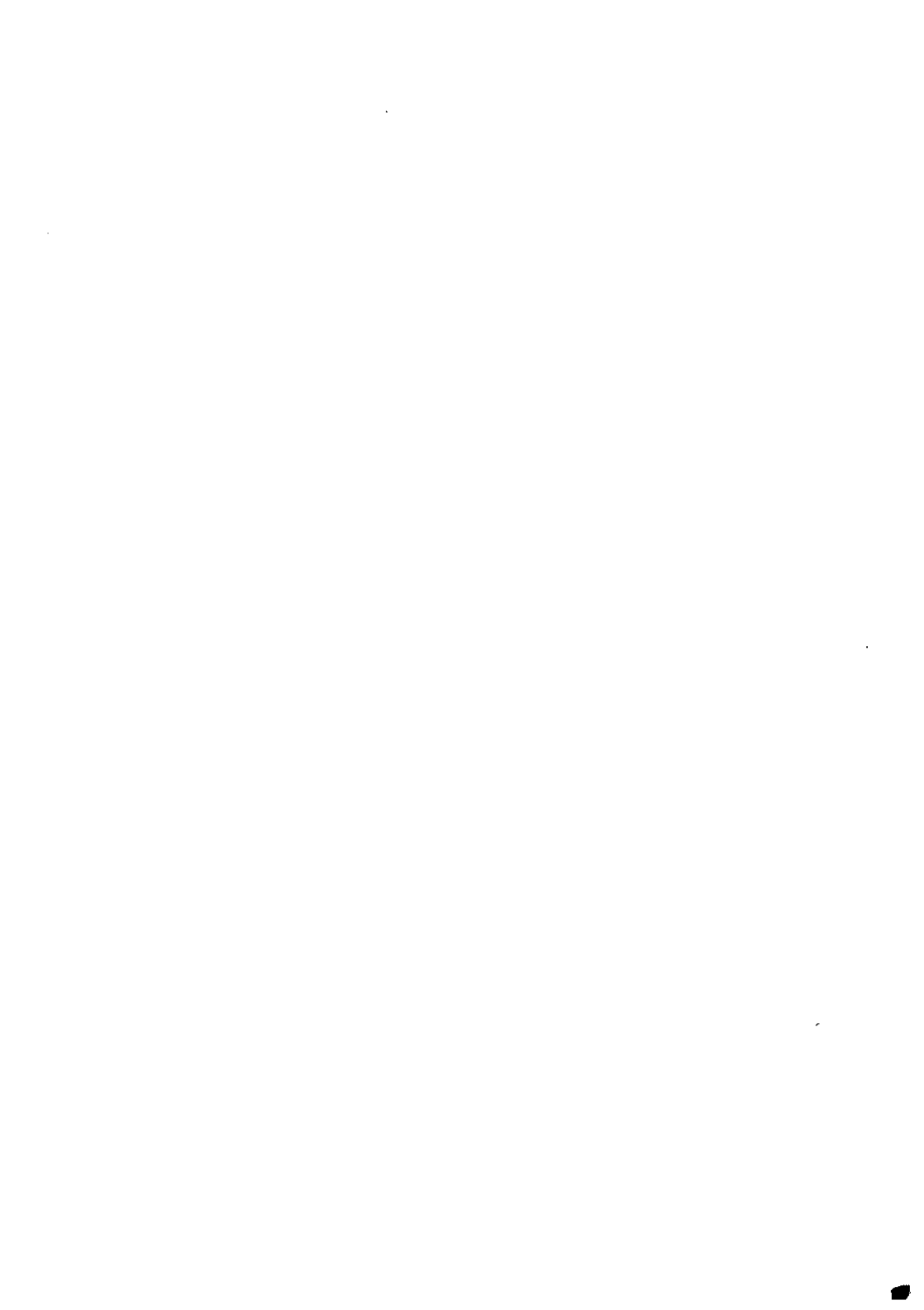




情報とは何か

森 洋久 著

国際日本文化研究センター准教授



目次

第1章	はじめに	7
第2章	生命は機械化する	11
2.1	シュレディンガー『生命とは何か』	11
2.2	細胞という環境－プログラミング・モデル	13
2.2.1	はじめに	13
2.2.2	情報の二つの形態 核酸とタンパク質	14
2.2.3	透明性	17
2.2.4	生体膜	24
2.2.5	プロセス間通信	31
2.2.6	生体膜間の物質輸送	34
2.2.7	パケット・ヘッダ	39
2.2.8	小胞体タンパク質輸送	40
2.2.9	細胞骨格	47
2.2.10	パケット・エンキャプスレーション	52
2.3	プログラミング・パラダイム	58
2.3.1	細胞とノイマン型コンピュータ	58
2.3.2	オートマトン	67
2.3.3	手続きとフローチャート	75
2.3.4	データフローと関数	79
2.3.5	マークアップ言語	86
2.3.6	シグナル・フロー	90
2.3.7	アナログ現象とデータフロー	97
2.3.8	連続性と離散との関係	101

2.3.9	アルゴリズム	107
2.3.10	モンテカルロ法	113
2.3.11	細胞のアルゴリズム	121
第3章	機械は生命化する	131
3.1	人工知能	131
3.2	道具としてのコンピュータ	145
3.3	ユビキタス・コンピューティング	152
3.4	位置情報システム	156
3.5	情報は誰のためにあるのか	159
3.6	歴史的地図の多様性	161
3.6.1	Google Map	161
3.6.2	経緯度の系譜と地球説	161
3.6.3	地誌としての地図	170
3.6.4	非地球説の地図の数々	181
3.7	ユビキタス・コンピューティングのゆくえ	184
第4章	連続と離散	187
4.1	乱雑性	187
4.2	情報エントロピー	190
4.3	熱力学エントロピー	194
4.4	情報を集めると情報エントロピーは下がるか	194
4.5	推論	197
4.6	平均化	200
4.7	共鳴・量子飛躍	202
4.8	振動・ループ	207
4.9	共振・崩壊	213
4.10	フィードバック制御	223
4.11	$1/f$ ゆらぎ	225
4.12	連続と離散の循環	235

第 5 章 主観的生命	239
5.1 生命	239
5.2 科学的な確からしさ	243
5.3 おわりに あるいは $\lim_{t \rightarrow -\infty} e^t = 0$	247
参考文献・関連図書	251

第1章 はじめに

総合研究大学院大学・学融合推進センターの研究費助成によるプロジェクト「日本における諸科学の編成と基礎概念の検討 — 文理融合の有効性をさぐる」では、「エネルギー」「生命」「情報」「科学政策」の四つの柱を掲げ、それぞれにチームを編成し、最終的に科学政策および文理融合型の研究のあり方に対して、提言を行うことを目標にしてきた。本書『情報とは何か』は、「情報」班の報告書としてここにまとめるものである。

「情報とはなにか」一言で答えよと求められれば、次のように答えるのが最も簡潔ではないだろうか。

情報は物事の確からしさである。あるいは、物事が役に立つかどうか、ということ。

これは、ありふれた答えである。だが、この定義には大きく二つの問題が潜んでいる。一つ目は、誰にとっての確からしさなのか、誰にとって役に立つかどうかなのか、ということである。

本書の構成はまず、情報は誰にとっての情報かということを機械と生命の対立で考え始める。2章 生命は機械化する では、分子生物学の視点が生命を機械として捉える視点であるということを具体的に示すために、細胞をコンピュータや情報通信のメカニズムと比較する。「細胞は機械である」と言いたいのではなく、「分子生物学的な視点で細胞を見る」という行為の中にア priori に「細胞を機械として見る」という行為が含まれているのである。3章はこんどは逆にコンピュータ・サイエンスのシナリオが生命や知能を意識しているということを描き出す。謙虚な情報科学者は依然、我々の開発しているのは生命や知能ではない、と言うのであるが、否応無しに世界中に広がっていくユビキタス・

コンピュータの行く末は生命と機械の混同ではなからうか。

「シリコンチップが人間を超えた頭脳をもち得るだろう。」と言うとにわかには信じ難い。だが、いま iPS 細胞などのリセットされた細胞の開発競争が加熱を始めている。理論的にはクローン人間も作れる時代に、元来知能をもち得る細胞を部品として組み込んだ、人間と同等の能力を備えた頭脳チップとなると、近い将来出現するのかもしれない。もはやシリコンチップだけが機械ではない。いま、機械と生命の境界線が薄れつつある時代にさしかかっている。西垣通、あるいは謙虚な情報科学者たちは、機械は情報を生まないと主張する [1]。その意味が薄れつつあるのではないか。

4章は、確からしさについての叙述となる。乱雑性を完全に定義することは難しいということ、情報エントロピーと熱力学エントロピーは本来違うものであるということをもとに説明すること、このへんから始まる。エントロピーが増大すると乱雑度が増す、ということは閉じた系では確かに正しいが、振動現象と拡散現象を考察することで、エントロピーの現象が必ずしも安定性を保証する訳でもなく、また、拡散現象というエントロピー増大を招くメカニズムで安定性を得ることが出来るという事実がある。このように安定性や識別可能性とエントロピーの増減は複雑な関係であり、単純に議論することができない。

この考察を通して見えてくるのは、乱雑性と秩序、連続と離散という関係である。情報の主体が他者を識別するためには、乱雑であってはいけない。また境界線の不明な連続なものも拒否される。識別されるものは秩序だてられ、離散的である。ところが、ある種の乱雑性と連続性は秩序と離散性を内在している。乱雑性が確率的に捉えられる時、確率分布として連続性的一种と捉えられる。さらに連続性が微分方程式という言葉で記述可能な法則性に則っている時、その解はいくつかの方法で離散的なパーツに分解可能である。こうして、我々は世界を離散的な言語で捉えられるようになってきているのである。

だが、いったん識別圏に入った離散的な事象も多量に集まってくると、事実上乱雑化する。しかしその乱雑性も確率分布として捉えることが出来れば再び連続性へと結びついて行く。このように乱雑性と秩序、連続と離散は大きく循環しているのである。この循環がある故に、真の乱雑性というものを考えることはきわめて難しい。また、我々はこの循環のどこか途中に居る訳だ。

本来このあとにもう一章を設けたかったが時間切れとなってしまった。その内容は、前章を受けた形で、乱雑性、無秩序、連続性、無境界について論じるはずだった。生と死といった問題、論理的には追うことのできない問題である。博物館やアーカイブ、継承という問題も考察の対象だ。3章で地図の話をするが、その帰結として現れてくる現代に失われつつある多様性の問題をもっと掘り下げ地図のもつ物語性とといった問題も扱う予定であった。語り得ないことについて「語る」ということについて語るはずだった。しかし、情報にとって最も難しいこの問題が、いま、空白となっている、ということも、一つの語り方かもしれない。

2013年3月30日、総合研究大学院大学・学融合プロジェクトにおいて「情報を考える」というシンポジウムを行った。吉見俊哉氏「ネット時代のコペルニクス」金子務氏「プレス・メディアと情報」今井浩氏「学術情報の価値—巨大ネットワークの情報」高野明彦氏「連想の情報学」森洋久「地図と情報、その歴史と技術」の五発表、その後、発表者に、鈴木貞美氏と村上直之氏を迎えて「学融合における情報の役割」と題してパネルディスカッションを行った。

学問には、文系と理系という強い二元論がある。しかし、文系とも理系とも着かない学問分野というのは確かに存在する。理系とはなにか、文系とはなにかということ突き詰めて行くとその違いをすっきりと定義することなどできない。リベラル・アーツの歴史、あるいは、孟子や荘子、または、産業革命と学問の関わりなどを具にたどることによってしか叙述することが出来ない。

歴史的に叙述することは、端的に言ってしまうと、学問と社会の関わりでの記述である。理系の学問の社会的関わりを議論すると、とかく政策との関わりが色濃くなる。科学的な論理を扱う者も、技術、産業の論理から逃れられないと言えるかもしれない。崇高な哲学を議論するも、本の売れ行きはその需要層の気分に左右される。文系も理系も崇高な学問の真理を求めようにも、どろどろとした人間の利害関係とは無縁ではいられない。シンポジウムの発表者の一人は、炙られたフライパンの上という表現をしていた。学者の社会への係わり方は、いろいろあるだろう。完全な学問を求めて食うことをやめるか、逆に、積極的に敵と関わり、血みどろのどろどろになるか。

最近、人類学者との付き合いの中で、面白い考え方があることを知った。たとえば、医療人類学という分野は、医療、特に現代科学の中核とも言える最先

端医療という場を人類学的な視点で考察していく。当然その研究は欧米的医療の世界から、非欧米的な医療の世界へと話が展開される。だが、非欧米的医療の方法論、形態の記述で終わってしまったら、ふつうの人類学である。研究の対象はあくまでも欧米的医療であり、これを他の医療世界とあわせて相対化して見せてこそ意味があるのである。主体と客体の逆転、つり下げていたものの横に自分もつり下げしてみる、ということである。研究対象が自分に跳ね返ってくる痛快さがある。このような相対化のプロセスは、かつてカフカに揶揄されたように¹人類学者には珍しいことではないかもしれない。しかし、このような相対化の思考を人類学の範疇にとどめておくことはもったいない。他の学問、とりわけ理系的な学問、科学や技術は意識するべきではないか。

学融合、学際的といった言葉は、政策的に好まれる言葉である。しかし実際に目的が達成できたと実感できるものは少ない。おおよそはただの寄せ集めで終わってしまう。それは、各研究者が自分の分野を他の分野の中で相対化して見ることが出来ないためである。会議に自分の方法論を持ち寄ることが出来ても、他者の方法論を持ち帰ることが出来ないためである。

最後の章は「心の理論」の発展系として、情報という観点から生物をリカーシブに見る。

¹ 『ある学会報告』

第2章 生命は機械化する

2.1 シュレディンガー『生命とは何か』

まず、シュレディンガー著『生命とは何か』[2]から始めよう。本書が出版された時期は、1944年であり、ワトソンとクリックによるDNAの二重螺旋構造の発見の1953年の九年前である。生命という概念が科学の領域に取り込まれようとしていた時期と言えよう。現代の分子生物学的に見れば、異なる点も多い。遺伝子がなぜ永続的に存在し生命を維持して行くことが出来るのかを説明するために本書は「非周期性結晶」という概念を提示している。遺伝子は分子であり、分子を構成する原子間には個体を形成する原子間にあるのと同じ強固な力「ハイトラー・ロンドンの力」が働き、遺伝子を破壊するような外部からの攪乱要因を軽減している。

遺伝子が分子で構成されている以上、構成する原子間にそのような強固な力が働いていることは事実である。しかし、遺伝子がこのような強固な結合によってのみ維持され、分子構造を強固なままでに変えないとなると、細胞の日々の営みの中で、タイムリーに遺伝子の中から必要なときに必要な情報をとりだし代謝に利用して行くことが出来なくなってしまう。むしろ逆である。遺伝子は細胞の様々なステージにおいて切り離されたり、また接合されたり、絶え間なく変化している。このような変化の中で、確率的に数々の間違いが発生し、またウイルスなどの外敵の進入によって、“意図的に”書き換えられる。これらの攪乱は、シュレディンガーの述べているX線による遺伝子の攪乱にくらべ遥かに高い確率で発生する。ところが、こうした攪乱に対して誤り訂正の代謝が働き、あるスパンで見ると、遺伝情報が維持されるようになっている。

シュレディンガーの考えは言い換えれば、堅牢な要塞で外敵が進入しないようにする方法なのに対して、実際の生命では、外敵も進入を許してしまうが、

進入した外敵をことごとく撲滅する仕組みである。動的平衡 [3] という営みが遺伝子にもあるということであり、量子力学的のみというよりはもっとマクロな化学的メカニズムによって遺伝子の永続性、生命の一貫性が維持されている。

このような、細胞のなかでおこる化学反応、つまり代謝はシュレディンガーの時代の物理学者、科学者にとっては信じられないものであろう。これらの化学反応は、生物の中ではそれがごく少数の分子の間でおこる。ときには一つの分子対一つの分子でおこったりする。よく知られていた、実験室のなかで行われる化学反応は途方もなく多量の分子同士の反応である。一つ一つの分子が選択的に起こす反応という化学反応の存在は思いも寄らないものだったのだろう。

この時代はまだ有機化合物の合成を自由にできる時代ではなかった。数十億個の塩基が非周期的に配列した DNA や、数百個のアミノ酸からなるタンパク質といったものが、細胞の中では自由自在に合成されているのである。人工的に合成できない有機化合物があるのに、それをなぜ生命はいとも容易く合成するのか、ということを考えてだけでも、生命活動が他の物理化学現象に比べ特別なものと見えたと言えよう。シュレディンガーは7章「生命は物理学の法則に支配されているか？」では、生命を支配する新しい物理法則があるに違いないとしながらも、「新原理は物理学と相容れないものではない。」と言っている。

当時、生命は物理学の法則とは異なる法則に支配されおり物質と生命は別物であるとする考えも優位にあった時代に、物理学者が「新原理は物理学と相容れないものではない。」と発言することは画期的なことだったと言える。その後、分子生物学は目覚ましい発展をとげ、今日、シュレディンガーが考えている以上に生命は物質的であることが分かったと、言えるのかもしれない。

だが、物事を物理科学的方法論によって理解することを可能にするということは、その知見によって対象をシミュレーション可能とすること、あるいは、人工的に組み立てることが可能となることを意味している。それらの研究が、人工生命の可能性や生命的な機械の可能性を示唆しはじめている。本章では、分子生物学によって明らかになった主な生命現象、あるいは、生命に関する様々な仮説を説明するとともに、これを情報理論やコンピュータ・サイエンスにおけるシステムと比較することによって、機械化していく生命観が浮き彫りになっていく。

もう一つの注目すべき点として、本書全体に渡って一貫して流れている思想

と言えるべき、量子力学と統計力学における物質の無秩序と秩序の関係性がある。ハイゼンベルグの不確定性原理による粒子の不確定性＝無秩序がプランクの量子飛躍の原理により相殺され原子レベルでは安定を保つ。それにより分子という秩序が生まれるが、分子の運動は無秩序である。だが、分子が多量にあつまった物質では統計的に捉えることによって統計力学的な秩序を得る。この無秩序、秩序、無秩序、秩序という階層構造は、我々が考察しようとする情報という概念に大変関係のあるものである。つまり秩序は識別可能な状態＝情報である。このようにシュレディンガーが無秩序と秩序の階層構造の中で生命を捉えようとするのは、本書には情報という言葉が現れないにもかかわらず、やがて到来する情報化社会を見据えたかのように、情報と生命の関係性を示唆するものである。

2.2 細胞という環境 – プログラミング・モデル

2.2.1 はじめに

この節では、生命科学というものが何を求めているのか、独自の視点で眺めてみたい。分子生物学の最近の教科書「分子細胞生物学」[4]を参考に、生命科学によって明らかにされた諸現象を情報科学的な発想へ置き換えることを試みる。ある意味そういう「遊び」であり、ジョークである。だがおそらく徐々に本当にジョークと言えるのかと思えてくるであろう。「科学的に考える」ということは「機械として捉える」ということとほとんど等価である。

ある科学的な学説を証明する方法は、その学説により現実世界をシミュレーションした場合に現実の観察や実験結果とそれがよく合致することを示すことである。となれば、得られた学説をシミュレーション＝機械として構築することが出来るということである。生命科学において「人工的に」細胞を構成することは難しい。したがっていろいろ議論の余地がある。しかし、生命科学に、「人工生命」への指向があることは事実である。それは科学という方法論に本質的に潜んだ性である。

2.2.2 情報の二つの形態 核酸とタンパク質

まずは、生命科学と言え言わずと知れた遺伝子、核酸の話からしよう。普段、遺伝子、染色体、DNA という三つの言葉を何気なく使っているのではないだろうか。だが、この3つの言葉は厳密には異なる概念である。

このなかで最初に発見されるのが遺伝子である。G.J. メンデルが1865年に提唱された遺伝形質を規定する仮想的な因子のことである。この時点では遺伝子が物質なのか、あるいはそうではないのか分からなかった。次に、染色体は、細胞を色素で染色すると染まりやすい、細胞内の核の中に存在する物質である。これはDNA (deoxyribonucleic acid デ・オキシ・リボ核酸) という分子といくつかのタンパク質から構成されたものである。

現在では、DNA が実際の遺伝子である、ということが分かっているのだが、そこに至るまでには長い歴史があった。メンデルの法則も、1900年に、ユーゴー・ド・フリース (オランダ)、カール・エリッヒ・コレンス (ドイツ)、エーリッヒ・フォン・チエルマック (オーストリア) によって再発見されるまでは不評であった。この再発見以降は、遺伝子が実際になんなのか問題となり、1903年に、ウォルター・S・サットンが遺伝子が染色体上にあることを提唱した。

染色体はDNAの他にタンパク質も含んだ物質である。従って、染色体のなかが遺伝子なのか様々な議論が繰り広げられた。最初はタンパク質が遺伝子であると考えられていたが、オズワルド・エイプリーによる肺炎球菌の実験により形質がDNAによって細胞から細胞へ伝搬することが証明された。ここで遺伝子がDNAである可能性が高いということになる。

1953年にジェームズ・ワトソンとフランシス・クリックによって、DNAは生体内で『二重らせん構造』をとっていることが示され、さらに、DNAの遺伝、形質発現に関する生物学の中心的メカニズムであるセントラル・ドグマが提唱された。

セントラル・ドグマは今日では大変に複雑なメカニズムであるが、まずは誤解を恐れずに極度に単純化してみよう。一言でいうなれば、細胞内で行われているDNAの複製とDNAをタンパク質に転写するメカニズムである(図2.1.(a))。これをコンピュータに例えてみるならば、(b)となる。データはDVD上に光学信号として保存されている。これをドライブに挿入し、PCで開くと、メモリ

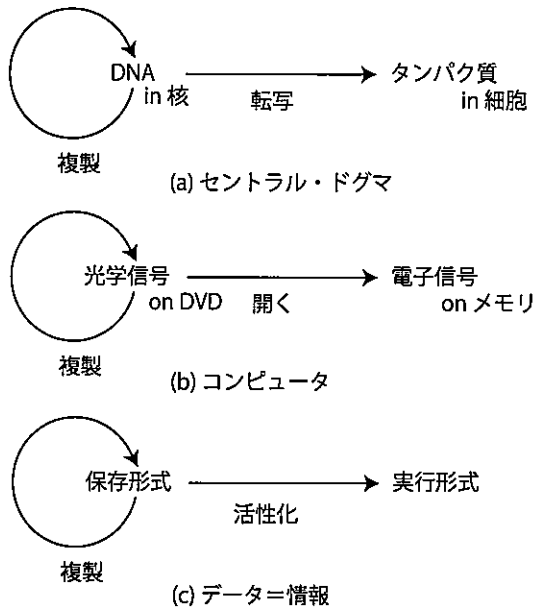


図 2.1: 情報の二つの形態

上へ転写される。DVDの中にはアプリケーションであったり、あるいは、音楽や画像データであったり様々な情報が入っており、それがメモリ上に転送されると、実際にプログラムや閲覧データとして機能し、我々はそのサービスの恩恵に預かることができる。

DVD上のデータと、メモリ上のデータは、同じデータである。ところが、どこにあるか、あるいはどういう形態で存在するかによってその役割が異なる。DVD上にあるデータは「保存された」データであり、非活性状態である。それに対して、メモリ上にあるデータはサービスを提供している「活性化された」実行状態である(図2.1.(c))。情報はいつ利用するのかということが重要である。全く利用されない情報は無意味であるということは誰でも気づくことであるが、ある特定の情報が常に参照されるようになると、むしろその情報が使われない状況の方が価値をもつようになる。

タンパク質は、細胞における様々な反応を促進したり抑制したりする触媒の機能をもつ。これによりタンパク質自身をも含む様々な物質の合成、分解、輸送に携わっている。また一方で、合成、分解、輸送の原料ともなる。細胞におけるほとんどの機能がタンパク質によって実現されていると言っても過言ではない。情報は、このタンパク質のもつ機能を指し示していると考えれば、DNAからタンパク質へ転写は、保存状態の情報を活性化するシステムがセントラル・ドグマと言えよう。

コンピュータにおいて、活性化された情報は、CPU(中央計算装置)を動作させる。では細胞においてCPUに相当するものはなんであろうか。CPUは、情報の計算の方法を決めているシステムである。計算の方法はCPUの仕様書によって与えられ、集積回路によって実現される。細胞においてタンパク質の機能を決定づけるのは化学反応の様々な理論である。これがいわばCPUの仕様書に相当するならば、CPUに相当するものは化学反応そのものである。CPUに相当する物質というのはない、ということはコンピュータ・サイエンス的には少々違和感があるかもしれない。どうしても物質に結びつけようとするならば、タンパク質その他を構成する原子やその原子の結合等を司る量子、素粒子がそれに相当する。これらの相互作用が化学反応というCPUの「仕様」を実現している。コンピュータでは記録媒体とCPUは別物であるが、そのような構造では自己増殖は難しい。細胞は自己増殖するシステムであり、つまり機能

と、データ、あるいはそれを担う様々な媒体、こういったものが相互に入れ替え可能なシステムとして構成されている。データとプログラムを区別しないというノイマン型アーキテクチャの究極の姿が細胞であると言ってもよいかもしれない。

2.2.3 透明性

我々エンジニアが、このようなセントラル・ドグマを実現するシステムを構成しようとするならば、DNAの二重螺旋(図2.2)は厄介な構造である。従ってこのような物質は選ばないだろう。エンジニアの立場として選ぶならば、核酸の左右のアームの偏向が0度または180度になるようなそういう物質を選び、螺旋ではなく、丁度2トラック・テープのようなリニアな梯子状の物質「リニア・ゲノム」を求めるだろう。

二重螺旋がなぜ厄介か、ちょっとした手元の実験で確かめることができる。ゴムバンドを両手で引っ張りテンションをかけてよくねじると二重螺旋が出来る(2.4.(a))。この状態で両手の間隔を徐々に狭めて行くとどうなるであろうか。二重螺旋化したゴムバンドがさらにねじれ二重の二重螺旋になっていくのが分かる(2.4.(b))。これを超螺旋構造と呼び、これはざっと染色体の中に折り畳まれたDNAの構造である。複雑な折りたたみ構造が、テンションをかけたり緩めたりするだけで折り畳み構造を作ることが出来る。

もう一度(a)の状態に戻ろう。両端を割り箸などで固定しておき、真ん中の二重のゴムバンドを引き離し1本鎖にしようとするとうどうなるか。結果は、螺旋のねじれが両端に寄ってしまい、無理なテンションがかかることになる(c)。この状態では決して2本のゴムバンドを引き離すことが出来ない。DNAの転写や複製、様々な過程でそのようなねじれを解消する機能が働く(図2.3)。

リニア・ゲノムを折り畳む場合は、カセットテープやオープンリールのように巻き付ければ良い。一方で、トラックを二つに分離する場合は、テープの真ん中にはさみを入れて切り離して行けばよい。実にシンプルだ。地球上に生命が誕生した45億年前、どろどろとした海底火山の火口付近には実はDNAをベースとする生命以外にも、様々な形態の物質をゲノムとする多種多様な生命が誕生していたかもしれない。その中にはエンジニアが標榜するリニア・ゲノム生

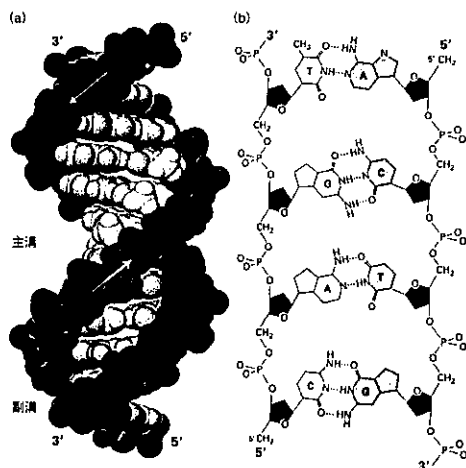


図 2.2: DNA の二重螺旋構造

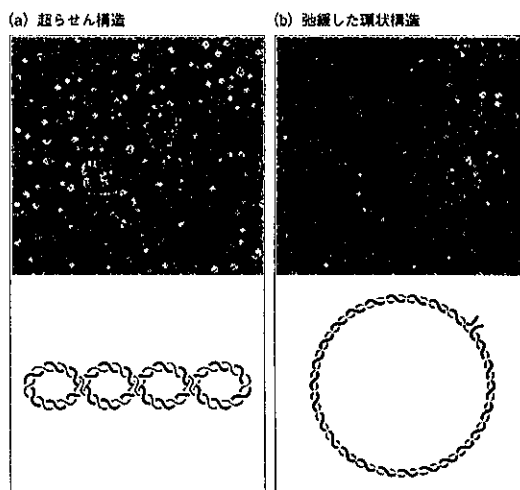


図 2.3: DNA のねじれの解消

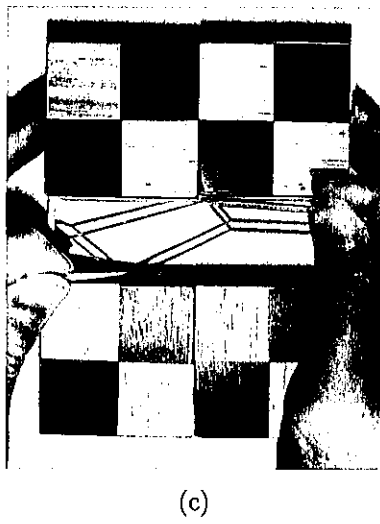
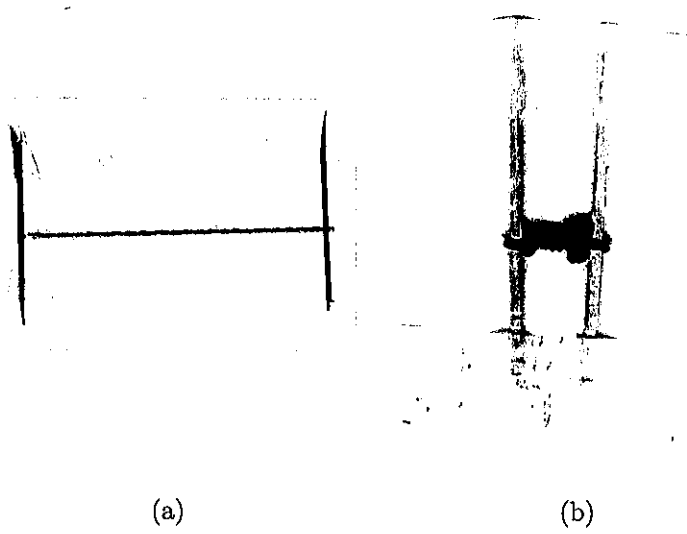


図 2.4: ゴムバンド二重螺旋

命体もあったかもしれない。だが進化の過程でそういったものは滅びて行った。複製や転写、折りたたみのしやすさは「便利」である反面、ウィルスの進入等により、破壊される可能性も高い。確かに CD-ROM や DVD、USB メモリなどの複製の容易い電子媒体は便利で扱いやすい反面、簡単に破壊され消去される。それよりも紙は水にもつよい。もっと言うならばアレキサンドリアの図書館にあった石の書物は火災の中も耐え抜くことが出来る。複製のしやすさと耐久性はトレードオフである。確かに複製を重視すれば、外部攪乱に対して耐久性がなくなる。一方で、耐久性を重視しすぎると複製が難しく、自己複製が出来なくなってしまう。ちょうどいい塩梅で進化の過程を勝ち抜いてきたのが唯一、核酸の偏向角度なのかもしれない。

全体をみると情報の二つの形態をスイッチするというシンプルな機能が実現されているのだが、実際にはこのように複雑である、ということは、じつはシンプルな大目標を様々な条件のもと実現する必要があるからだ。その条件の一つが、外部からの情報攪乱に対し耐久性をもつことなのである。そのほかにも様々な生物学的理由があって、このような複雑な機能となったということである。むろん細胞は人間が設計したのではないので、最初に条件を数え上げた人や神が居た訳ではない。攪乱をも含む、実際におこる現象に対して、適応性を備えたものがたまたま現在の細胞だったわけだ。

多くのシステムの場合、シンプルな機能を実現するために多くの機能が必要である。そして、目的とする大目標に対して、様々な小さな目標も実現しなければならない。最も重要な大目標に対して、それらの複雑な機能をどれだけ隠すことが出来るか、もし隠しきることが出来ればエレガントなシステムが得られる。これが透明性である。透明性という概念は隠す複雑な機能を中心に考えたものであるが、シンプルな大目標の方を中心に考えると、抽象化という概念になる。これは表裏一体の概念である。

人工的なシステムで、透明性の最も優れたものの一つとして通信技術における階層構造＝プロトコルが挙げられる。図 2.5.(a) を参照されたい。電話線は電気信号を伝えるものである。二番目の段は、電話は話者が音声を入力すると遠隔地の話者へ同じ音声を送るものであることを示している。(a) の最後は今あまり見かけなくなったが FAX は紙の文書を遠隔地へ伝えるものである。この三者はその伝える仕組みを透明化しているが、その仕組みが見えるように

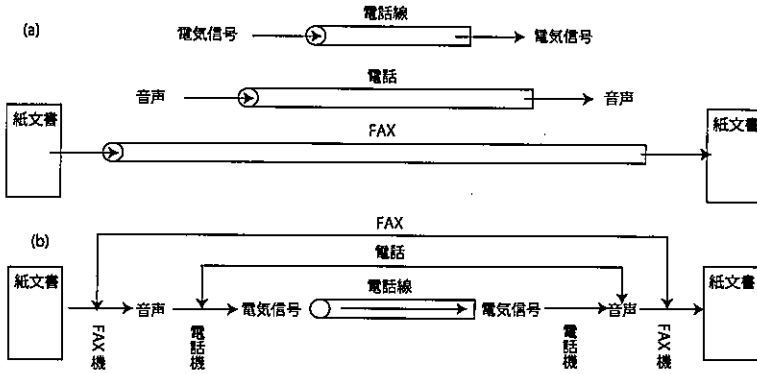


図 2.5: 電気信号、電話、ファックスにおける通信の透明性

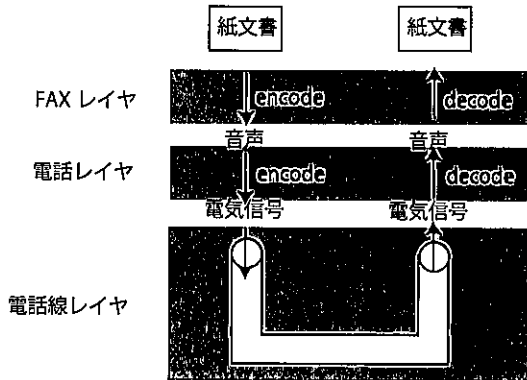


図 2.6: 電気信号、電話、ファックスにおける通信の階層構造

すると (b) のような構造になっている。FAX は紙文書のあるルールに従って音声に変換する。電話は音声を電気信号に変換する。電話線により伝えられた電気信号は、遠隔地では、逆のプロセスをたどり、音声、紙文書に再現される。

これはベルーナ、シャノンの情報理論におけるエンコードとデコードのモデルであるが、通信技術の階層構造はすこしこの見方を変える。図 2.6 は図 2.5(b) のパイプ構造を単純に折り曲げてみたものである。すると、FAX に係わるエンコードとデコード、電話に係わるエンコードとデコードがそれぞれ対となり、階層構造が見えてくる。これが通信技術における階層構造である。FAX にとっては、電話レイヤ以下は音声を伝える通信というかたちで抽象化されている。音声をどのような形で通信するかは考えなくてよい。つまり透明なのである。

一つのレイヤは上からの入力を加工して伝えるエンコードプロセスと、下のレイヤからくる情報を加工して上に伝えるデコードプロセスの対からなっている。このレイヤという構造を導入することによって、様々な通信のシチュエーションによってレイヤを変更することが可能となる。たとえば、電話レイヤを AD/DA 変換¹に置き換えることによって、デジタル電話、デジタル・ファックスが実現する。FAX の代わりにデジタルデータを音声データに変換する MODEM²に置き換えることによって、紙文書のかわりにデジタルデータを転送することが可能となる。このようにレイヤ構造は機能のモジュール化を促進し、様々な開発者が開発する通信システムを相互に組み合わせて利用することが可能となる。これが通信プロトコルなのである。インターネットや携帯電話といった最新の通信機器もこのような階層構造を基本的な考え方としていることには変わりない。表 2.1 は、国際標準化機構 (ISO) によって策定された、通信機器の持つべき階層構造のモデルである。OSI (Open Systems Interconnection) 参照モデルと呼ばれる。

図 2.7 はセントラル・ドグマを模式的に表したものである。これを階層構造として眺めた時、図 2.8 となる。(a) はタンパク質の合成、つまり情報の状態のスイッチであり、(b) は細胞分裂のときの DNA の複製に使われるレイヤである。生命は人間が作ったものではないので、各種の様々なベンダーの通信機器がそろっている.... ということはない。だが二つの見方が出来る。一つは、我々

¹Analog-Digital / Digital-Analog 変換

²MOdulator-DEModulator : 変復調装置

表 2.1: OSI 7階層モデル (wikipedia より)

第7層	アプリケーション層	具体的な通信サービス (例えばファイル・メールの転送、遠隔データベースアクセスなど) を提供。HTTP や FTP 等の通信サービス。
第6層	プレゼンテーション層	データの表現方法 (例えば EBCDIC コードのテキストファイルを ASCII コードのファイルへ変換する)。
第5層	セッション層	通信プログラム間の通信の開始から終了までの手順 (接続が途切れた場合、接続の回復を試みる)。
第4層	トランスポート層	ネットワークの端から端までの通信管理 (エラー訂正、再送制御等)。
第3層	ネットワーク層	ネットワークにおける通信経路の選択 (ルーティング)。データ中継。
第2層	データリンク層	直接的 (隣接的) に接続されている通信機器間の信号の受け渡し。
第1層	物理層	物理的な接続。コネクタのピンの数、コネクタ形状の規定等。銅線-光ファイバ間の電気信号の変換等。

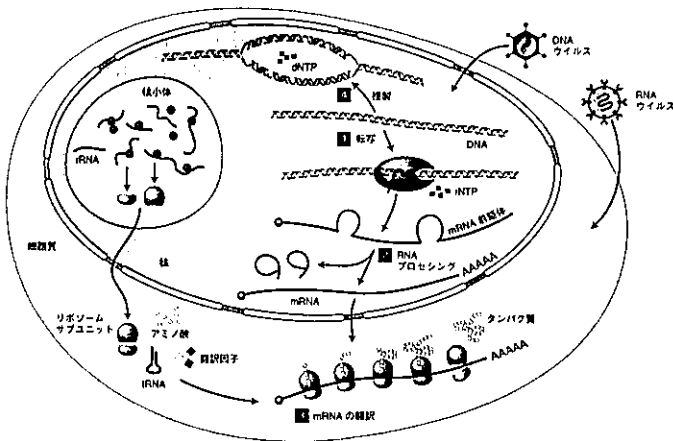
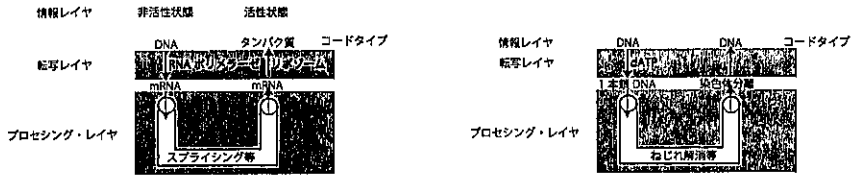


図 2.7: セントラル・ドグマ



(a) タンパク質の合成時のレイヤ構造 (b) タンパク質の合成時のレイヤ構造

図 2.8: セントラル・ドグマを階層構造でとらえると...

人間が分子生物学という手法を用いて、細胞を眺めるとき、それをやがて工学的な視点に転換する見方が生まれてきてもおかしくないという点である。もう一つは、一つの細胞はその種の細胞のみで閉じているわけではなく、環境の中の他の様々な生命との相互インタラクションを抱えている。インタラクションを可能とする理由は、ある種の共通性＝プロトコルが存在するという点であり、つまり、それがネットワーク・プロトコルへのアナロジーとなりえると考えられる。

2.2.4 生体膜

生体膜とは、細胞の外側を包み内部と外部を分け隔てるリン脂質の膜のことである。細胞の外壁だけでなく、細胞内部の様々な構造を作る。細胞に化学的な特質を維持する局所的な領域を作り出す効果をもたらす、細胞内の多くの化学反応に細胞膜が係わっている。生体膜は膜の内部と外部を分離し物質の出入りを制限する。一方で、生体膜上に分布するタンパク質は一部の許可された物質のみを輸送する。この生体膜の役割をコンピュータにたとえるならば、メモリ管理機能である。コンピュータ上のメモリでは、メモリ管理機能によって、情報はその機能ごとに類別され隔離される。そして許可された対象のみが許可されたものからアクセスできるようにコントロールされる。生体膜とメモリ管理機能を比較するにあたって、まずは生体膜の構造と機能から解説しよう。

図 2.9 は生体膜の構造を表したものである。リン脂質は (b) に示すように、親水性の頭と疎水性の尾を持つ構造となっている。このような2極の性質を兼ね備えているため、リン脂質の分子が水中で集まると、(b) のように尾の部分を

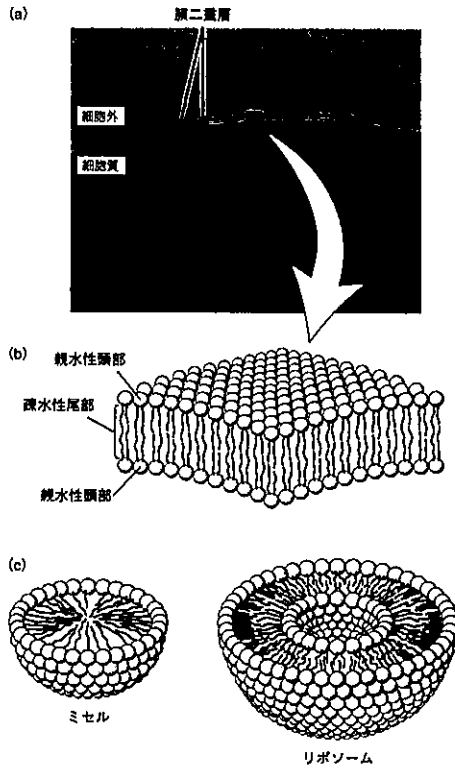


図 2.9: 生体膜の構造

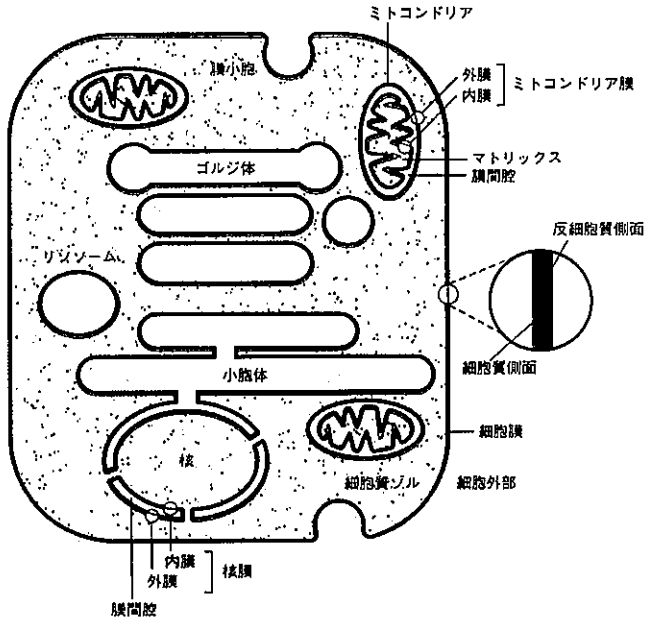


図 2.10: 細胞を構成する生体膜

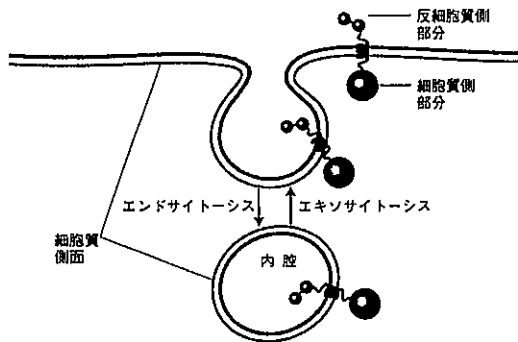


図 2.11: エンドサイトーシス・エキソサイトーシス

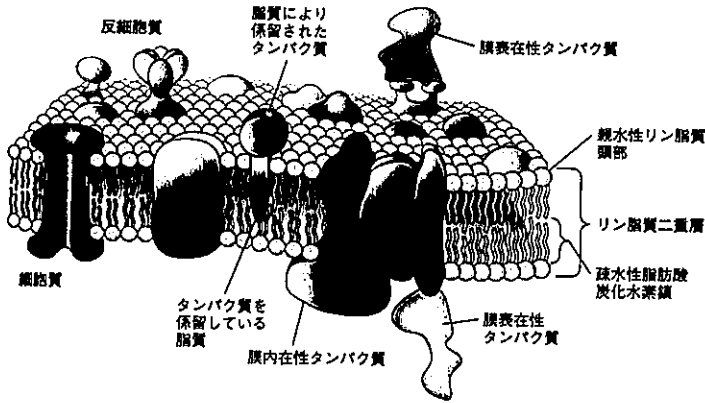


図 2.12: 生体膜上の構造

お互いに向け合ったリン脂質二重構造を作る。膜構造の端が存在すると、この部分は水に対して強い疎水力を発生するので不安定となる。そのため、平たい膜で存在することはなく、(c)に示すような端の部分が存在しない構造をとる。そのような構造には2種類あり、内部が疎水性部分で埋まったミセルか、内部に水溶液を蓄える球形を形成するリポソームである。このようにリン脂質の膜は内部と外部をもつことになる。細胞の一番外側の生体膜ではその外側は完全に外部環境である。この外部環境を反細胞質 (図 2.10 の紫の部分)、一方生体膜の内側を細胞質 (図の緑の点々の部分) という。細胞内の生体膜では「内側の内側は外」というトポロジカルな構造をとっている。図 2.10 では紫に着色された部分が反細胞質ということになる。このような構造によって、細胞質と反細胞質は厳格に分離され、直接は混じらないようになっている。それは例えば生体膜構造同士が結合した場合、あるいは、分離した場合でも、細胞質と反細胞質は混合しない (2.11)。

図 2.12 のように、生体膜上も様々な構造を備えた「領域」である。その構造はリン脂質に捕捉された様々なタンパク質である。これらのタンパク質の機能は膜の両側でイオンやタンパク質の輸送を行ったり、膜の表面に水流を発生さ

せたり、また細胞質、反細胞質にある特定のタンパク質を認識し結合、排斥したりする。さらには結合しているタンパク質によって様々な化学反応、たとえば、DNA や RNA の転写などを促進する。こういった機能により膜は細胞質、反細胞質内に様々な変化をもたらす、いわば細胞の駆動系である。

コンピュータに目を転じよう。メモリ管理機能がない、つまりハードウェアとしてのメモリは、0 から順番に番号＝アドレスがふられたワード³の集合である。アドレスはおおよそ 1word で表すことが出来るように設計されているのが常である。1word=32bit のマシンであれば、アドレスも 32bit アドレスであり、0 番地から $2^32 - 1 = 4,294,967,295$ 番地までである。64bit マシンであれば、アドレスは 0 番地から $2^64 - 1 = 18,446,744,073,709,551,615$ 番地となる。最近では 32bit アドレスの領域をすべて埋め尽くすメモリは存在するが⁴、64bit アドレスの領域を全部埋め尽くすメモリはそうそうあるものではない。従って、アドレス領域は全てがメモリで使われている訳ではない。また全てを記憶システムで埋め尽くす必要もない。特に便利な使い方として、アドレスの一部の領域を外部入出力として使う方法がある。たとえば、アドレスのある領域へデータを書き込むとそのデータが画像として表示されたり、あるいは、あるアドレスを参照すると、押されているキーボードのキーが分かるような仕組みが実装されることもある。図 2.13 は IBM PC/AT 互換機とよばれる、いわゆる PC のメモリマップ例である。この例でもビデオ出力用のデータを出力する領域がメモリ空間上に存在することがわかる。

さて、このようになりニアなメモリ空間上にプログラムを配置するならば、単純に考えてもプログラムを配置する領域、データを配置する領域を分けて考えた方がよい。さらには、異なる種類のプログラムやデータを分けて配置すべきである。これらをごちゃごちゃに配置したならば全体の構造が把握できない状態となってしまう。細胞で言うならば、生体膜がない状態である。生成されたタンパク質があったとしても、一様な水溶液の中に拡散して行くだけであり、生体としての機能は実現することが出来ない。

データやプログラムを整理し、配置することをプログラマが手で行っても良

³1word のビット数は提供されるシステムによって異なる。32bit マシンや 64bit マシンといった呼び方があるが、これはそれぞれ 1word=32bit,64bit であることを示している。

⁴32bit マシンであれば 16Gbyte のメモリがあればよい。

00000000h~000002FFh	BIOS 割り込みテーブル
00000300h~000003FFh	BIOS スタック領域
00000400h~000004FFh	BIOS データ領域
00000500h~0009FFFFh	コンベンショナル・メモリ領域
000A0000h~000BFFFFh	ビデオ・メモリ (EGA, VGA)
000B0000h~000B7FFFh	ビデオ・メモリ (MDA, CGA カラー, VGA モノクロ)
000B8000h~000BFFFFh	ビデオ・メモリ (CGA, EGA カラー, VGA カラー)
000C0000h~000C7FFFh	ビデオ BIOS – ROM (EGA/VGA)
000C8000h~000CFFFFh	拡張 BIOS
000D0000h~000DFFFFh	拡張 BIOS
000E0000h~000EFFFFh	システム BIOS
000F0000h~000FFFFFh	システム BIOS
00100000h~00EFFFFFh	メイン・メモリ/PCI/ISA メモリ空間
00F00000h~00FFFFFFFh	メイン・メモリ/PCI/ISA メモリ空間
01000000h~FFFFFFFh	メイン・メモリ/PCI メモリ空間

図 2.13: IBM PC/AT 互換機のメモリマップ

い。しかし、我々が使っているような Windows や Macintosh, あるいは Linux サーバなど複雑なシステムにおいてはそれでは手に負えない。また不特定多数のアプリケーションが混在する中、メモリ上の配置の管理を自動化する必要がある。そこでメモリ管理システムが登場する。メモリ管理は、メモリ上にプログラムやデータが配置されていて利用されている状態の場所、利用されていない場所を区別する。メモリを利用したいという要求があると、空き領域から要求された分のメモリを配置 (allocation) する。利用していたメモリが必要なくなった場合にはそのメモリ領域を利用されていない領域へ開放 (free) する。これをメモリアロケーション機能という。さらに高度なメモリ管理では、利用されているメモリ領域に、実行可能なプログラムの配置された実行領域、実行することは出来ないデータ領域、あるいはデータ領域については、読み込み、書き込みのいずれかまたは両方が可能かといったメモリ・アクセス・パーミッションを与える機能などある。

メモリ・アクセス・パーミッションの考え方を発展させると、論理アドレスという考え方にたどり着く。論理アドレスとは、先に述べたハードウェアとしてのメモリにふられたアドレス⁵に対して付けられた名前である。一つのマシン上で複数のアプリケーションがあった場合に、これらのアプリケーションは同

⁵論理アドレスに対して物理アドレスと言う。

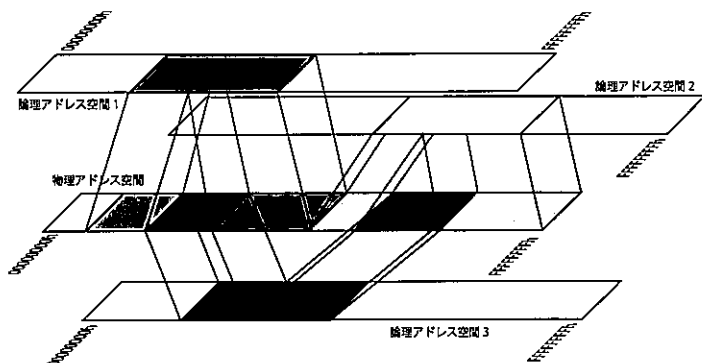


図 2.14: 物理アドレス空間と論理アドレス空間の関係

一マシン上にありながらお互いあまりインタラクションはないものである。従って、一つのアプリケーションは一つのメモリ空間を独り占めしたいものである。そこで、独り占め可能なアドレス空間を一つのアプリケーションのために提供するのが論理アドレスである。図 2.14 はひとつの論理アドレスの構成である。

真ん中にある、物理アドレス空間というのが実際にあるメモリに対応するアドレス空間である。この例ではアドレス 00000000h から FFFFFFFFh までを持っている。これに対し、周囲に 3 本の論理アドレス空間がある。その空間の大きさは物理アドレス空間と同じ 00000000h から FFFFFFFFh までのアドレスをもっている。それぞれ 3 つのアプリケーションがこの論理アドレス空間を利用している。

それぞれの論理アドレスは、与えられたテーブルによって物理アドレスへ変換される。論理アドレス空間のどの領域が物理アドレス空間のどの領域へ変換されるか、図 2.14 では論理アドレス空間と物理アドレス空間の同じ色の領域が対応している。こうして論理アドレス空間をアクセスしても、実際には物理アドレス空間のメモリを利用していることになる。

図 2.14 で分かるように、異なる論理アドレス空間に割り当てられる物理アドレス空間の領域は重ならないように注意深く割り当てられている。このため、異なるアプリケーションはお互いに同じメモリを取り合ったり破壊し合ったり

することがないように保証されている。このような論理アドレス空間は丁度、他の空間から完全に隔離された内部をもつ生体膜のリポソームに相当する。

論理アドレス空間の中にはデータの他にプログラムも配置される。このプログラムは基本的に自分の属する論理アドレス空間の外側をアクセスすることは出来ない。論理アドレス空間内のデータを処理するのみのプログラムである。一般に、このような論理アドレス空間とそこに配置されたプログラム、データをあわせてプロセスと呼ぶ。いままで使ってきたアプリケーションという言葉は、プロセスを利用するユーザサイドから眺めたときのネーミングである。時として一つのアプリケーションは複数のプロセスから構成されていることもある。

2.2.5 プロセス間通信

もし、プロセスが論理アドレスのなかで完全に閉じた世界を構成し、外部といっさい情報交換ができなかったとしたら、プロセスそのものの存在意義がない。プロセス同士が情報交換を行えてこそ、有用な情報処理が可能となる。そのため、多くの OS では、プロセス間通信という仕組みを持っている。

プロセス間通信にはいくつかの方法がある。最も単純な方法としては、通信したいデータと、そのデータをどのプロセスへ送るかを指定する `sendmsg`(SEND・メッセージ) 命令と、逆に送られてきたデータを受け取る `recvmsg`(レシーブ・メッセージ) 命令を使う方法である。この方法は `pipe` と呼ばれる方法である。明示的に通信したいデータを指定するので分かりやすい。

共有メモリという、もっと暗示的に通信する方法もある。図 2.14 をもう一度見てほしい。この図では物理アドレス上で異なる色に塗り分けられた領域は異なる論理空間にマッピングされている。従って、異なる論理空間で書き込まれたデータは論理空間同士では共有できない。ここでひとつ、異なる論理空間が一部の同じ物理空間にマッピングされていたとするとどうなるか。一方の論理空間でデータをこの部分に書き込むと、その論理空間のデータが書き変わるのみならず、別の論理空間のデータもいつの間にか書き変わっていることになる。これが共有メモリである。

pipe と共有メモリとどちらが良いかというのはプログラムの性質やプログラマの分かりやすさで決まる。よく、pipeの方が共有メモリよりも通信効率が悪そうに思うプログラマが居るが、良く出来た OS では、両者の差はほとんどない。なぜならば、すべてのコンピュータの源である CPU の機能では、論理空間同士の通信手段として用意されているのは共有メモリだけだからだ。OS が pipe を実現するにも OS が管理する特殊な共有メモリ領域を経由しているためだ。pipe では指定したデータを OS の領域へコピーするためのオーバーヘッドがかかるのではないかと思われるかもしれない。しかし賢い OS ではそのように実装しない。指定したデータが載っているメモリ領域のマッピングを変えることによって、別の論理空間にあたかも転送したように見せかけるのである。

かつて、このような共有メモリ機能を徹底的に追及した OS があった。現在のアップル・コンピュータが製造しているコンピュータ、iPhone などの OS である darwin や iOS の母体となった OS である MACH⁶という OS である。表向きは linux や BSD など posix 系の OS であるが、表向きのプログラミング・インタフェースは同一に保ちながらデータの共有の効率や方法論を徹底的に追及していた。

まず、上述のプロセス間の共有メモリであるが、共有メモリを使っているながら pipe と同じように見せかけるためにはもう一工夫必要である。データを受け取った側の論理空間で、受け取ったデータに別のデータを上書きした場合、通常であれば、送り元の論理空間でもデータが変化してしまうことになる。そこを MACH では、受け取った側のデータ領域に OS が密かに書き込み禁止状態を設定し、書き込みを検知する。書き込みが検知されると OS はそのときにデータ領域をコピーし、コピー先のデータ領域に書き込みデータを上書きする。このような詐欺まがいの作業はデータを受け取ったプロセスが知らない間に行うため、送り元のプロセスも送り先のプロセスも全く共有メモリで実装された pipe だとは知らないで済むのである。書き込みが発生するので、データをコピーするオーバーヘッドと同じように思えるが、実際には、受け取ったデータに新しいデータを上書きするニーズはあまり発生しない。ごくまれなので、データを転送すると必ず行われるコピーに比べずっと効率がよいのである。

さらに発展した機能として、メモリ・マップト・ファイル (memory mapped

⁶MACH マッハと発音する

file) という機能がある。データの送り元はプロセスや別の論理空間である必要性はない。ディスク上のファイルであっても良い訳だ。同じ方法をつかって、ディスク上のファイルがあるプロセスの論理空間のある領域にマッピングされ、その空間に書き込んだデータはディスク上に書き込まれるという機能を作り出すことが出来る。

通常、プログラムを実行する時、CPU への命令の集合体であるプログラムと、操作対象のデータをメモリ上へ展開する。メモリはディスクに比べ、アクセススピードが速く、アクセス方法が単純であり、プログラムを実行するときにはメモリ上でなければならない。しかし、コンピュータの電源を OFF にすると、メモリ上の情報は消えてしまう。このように電源を落とすと内容が消えてしまうメモリのことを揮発性メモリと呼び、これに対してディスクなどの電源を落としても内容が消えない記憶媒体のことを不揮発性メモリと呼ぶ。最近では、フラッシュメモリなどの高速アクセス可能な不揮発性メモリも登場してきているが、やはり製造工程が、揮発性メモリよりも複雑なので、高価かつ小容量である。

メモリ・マップト・ファイルの技術は、単純にディスクへのアクセスを楽にするだけではない。揮発性メモリと不揮発性メモリを結びつけることによって、両方のメリットをともに備えた記憶システムを実現することが出来る。プログラムの保存されているファイル、データの保存されているファイルを揮発性メモリへマップすることによって、コンピュータの電源を落したときに、プログラムのその時点での実行状態がそのままファイル上に残すことが出来る。コンピュータを再度起動した時、プログラムが最後に停止した状態から復活させることが出来る。プログラムの起動が速くなる、とか、不用意なコンピュータのシャットダウンによってデータが失われることが無くなる、というメリットがある。

しかしさらにそれ以上の機能が実現できる。この技術によって一旦ディスクに焼き付けられたプログラムの最終状態を、別のコンピュータで立ち上げることが可能になるのである。一つのコンピュータからプログラムやアプリケーションが変幻自在に別のコンピュータへ乗り移り、実行を続行する。また、ある最終状態から二つ以上のコンピュータへ乗り移り、実行することも可能である。同一のプログラムが別のコンピュータでクローン状態で実行されているという

ことが可能となる。このような技術をプロセス・マイグレーションと呼ぶ。

このような技術は、「絶対に停まらない」高信頼性コンピュータの実現にとって重要な技術である。一つのアプリケーションを実現するために、複数のクローンされたプログラムを動かし、お互いがお互いのバックアップとして常に働く仕組みが実現できるのである。高信頼性の他に負荷分散等様々な意味を持つ。

2.2.6 生体膜間の物質輸送

話を生体膜へ戻そう。コンピュータにおいて論理空間がプログラムの機能を分離するように、生体膜が細胞内を仕切り、分離する機能であるならば、プロセス間通信のように、その分離された空間を限定的、選択的に結合する機能も存在するはずである。だいたい戻すが、図 2.12 を参照されたい。空間を分離するリン脂質の膜にいろいろなタンパク質が突き刺さっているのが分かるだろう。これらのタンパク質は細胞質または反細胞質に浮遊する様々な分子、あるいはイオンを選択的に認識し反応する。これらのタンパク質の中に、一方の側のイオンや分子を、膜の反対側へ輸送する機能を持つものがある。これが一旦強固に分離された空間の間で物質の選択的交換を行う機能である。

イオンや分子の輸送は、通常、濃度の高い方から低い方へ拡散して行く。そのままの自然拡散では、ある種の疎水性の分子を除いては、生体膜を通過することが出来る物質はない。ほぼ全ての分子は膜タンパク質によってコントロールされている。一方、ある種の必要性によっては、イオンや分子を濃度の低い方から高い方へ輸送する必要がある。例えば、人間の血液中の NaCl(食塩)濃度は海水に近い濃度であるが、細胞質内の Na^+ の濃度はその 10 分の 1 である。このような状態を保つために Na^+ を細胞質内部から外部へ輸送するメカニズムが必要なのである。このような濃度勾配に逆らった分子、イオンの輸送にはエネルギーが必要でかつ、複雑な輸送メカニズムが必要なのである。

図 2.15 は、イオン、分子の輸送のメカニズムを 3 つに分類したものである。最も単純なものは、細胞質と細胞質外との濃度差に準じてイオンを転送する 2 のイオン・チャンネルである。しかし自然なイオンの拡散とは異なり、イオン・チャンネルに結合可能なイオンしか通すことは無い。また、条件によってはイオン・チャンネルは閉じていることもある。このようにして選択的にイオンを

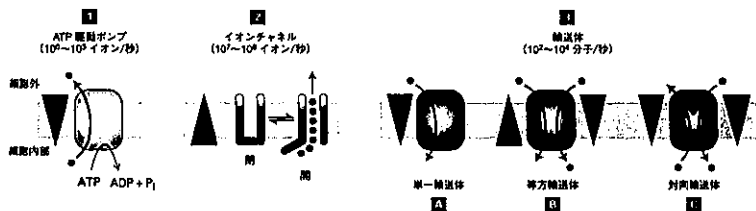


図 2.15: 膜タンパクによるイオン、分子の輸送

輸送する仕組みである。次に 3A は、イオン・チャンネルの分子版で、濃度勾配に従って分子を選択的に輸送する仕組みである。3B, 3C は、ある分子やイオンの濃度勾配を流れるエネルギーを利用して、濃度勾配に逆らって他の分子を輸送する仕組みである。3B はエネルギーを供給する分子の流れと同じ方向に他の分子を輸送するメカニズムであり、3C はエネルギーを供給する分子の流れとは逆向きに他の分子を輸送するメカニズムである。最後に 1 は、細胞のエネルギー源である ATP の加水分解のエネルギーを利用して、濃度勾配に逆らってイオンを輸送するメカニズムである。

このような分子、イオンの輸送により、膜内において様々な代謝（化学反応）を促進したり、膜の両側に新たな濃度勾配を作り、これをエネルギーにまた様々な作用や機能を実現する。またこれら 3 種類の輸送メカニズムが複数連携して働く場合もある。たとえば、ATP 駆動ポンプによって濃度勾配をさかのぼって輸送されるイオンが、等方輸送体、あるいは対向輸送体によって別の分子の輸送を駆動する。このようにして様々なイオンや分子が、ATP に代表される細胞のエネルギーによって濃度勾配に逆らって輸送される。例えば、神経細胞では Na⁺ や K⁺ の濃度勾配により細胞膜上に電位差の波を発生させ、これにより神経情報を伝達する。次に見て行くのは、タンパク質の選別による適切な配置を実現するメカニズムである。

遺伝情報はタンパク質のコーディングであり、それによってタンパク質が作られる。作られたタンパク質は適切場所に配置されなければその機能を発揮することは出来ない。そのタンパク質が使われる場所はどのように決定されるのだろうか。図 2.16 はそれを模式的に表している。DNA の情報は一旦 mRNA（メッ

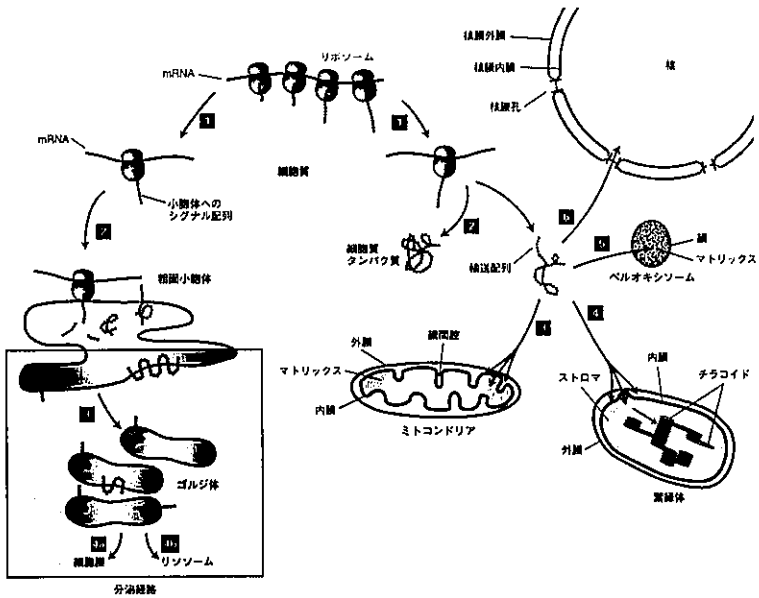


図 2.16: 合成されたタンパク質と、膜による選別

センジャー RNA) に転写され、核外へ放出される。核外へ放出された mRNA にはリボゾームという翻訳タンパク質が接続される。リボゾームは一種のテープ・リーダヘッドのような役割を果たす。mRNA の翻訳開始点から接続し、翻訳方向へ mRNA をなめて行く。三個の塩基を一つのアミノ酸に置き換え、一つ一つアミノ酸をつなげて行き、一続きのタンパク質を合成する。リボゾームが細胞質の空間の中でタンパク質を合成する場合 (図の右の方への流れ) と、小胞体の膜の表面でタンパク質を合成しすぐに膜内に取り込まれる場合 (図の左の方への流れ) がある。翻訳と膜内への転送の時間の関係で、前者を翻訳後転送、後者を翻訳時転送と呼び分ける。翻訳後転送は細胞内の各所で利用されるタンパク質の合成であるのに対して、翻訳時転送の流れは細胞外へ放出されるタンパク質の合成のプロセスである。

翻訳後転送の場合、翻訳されたタンパク質は細胞質内に放出される (番号 2)。このタンパク質の端に輸送配列と呼ばれるアミノ酸配列が着いているものは、その配列を認識できる膜構造に取り込まれて行く (番号 3~6)。図では、ミトコンドリア、葉緑体などへ取り込まれている。取り込まれたタンパク質からは、輸送配列部分は切り離され、本体のみが利用される。

翻訳後転送ではトランスコロンと呼ばれる膜タンパク質と、Sec63 複合体というタンパク質が関与している (図 2.17)。合成されたタンパク質のアミノ酸鎖の通り道を与えるのが、トランスコロンである。このタンパク質はアミノ酸鎖の N 端⁷にコーディングされている、輸送配列を認識しタンパク質を引き寄せ、この輸送配列部分を取り込み、直後に輸送配列部分はシグナル・ペプチターゼにより切断される (番号 1)。その後のアミノ酸鎖をトランスコロンのすぐそばにある Sec63 複合体と BiP・ATP が相互作用し、加水分解した ADP がタンパク質と結合し、アミノ酸鎖を内部へ引き込む (番号 2~4)。引き込まれたアミノ酸鎖から ADP が離れると、アミノ酸鎖は本来のタンパク質の折り畳み構造へ折りたたまれる (番号 5,6)。

翻訳時転送の場合においても、輸送配列 (シグナル配列) が意味をなす。図 2.18 を参照されたい。リボゾームが輸送配列までを合成すると、その転送配列をめがけて SRP が結合する (番号 2)。その SRP と小胞体膜上のトランスコロンと結合している SRP 受容体が結合し、SRP が離脱し、受容体も分解する (番

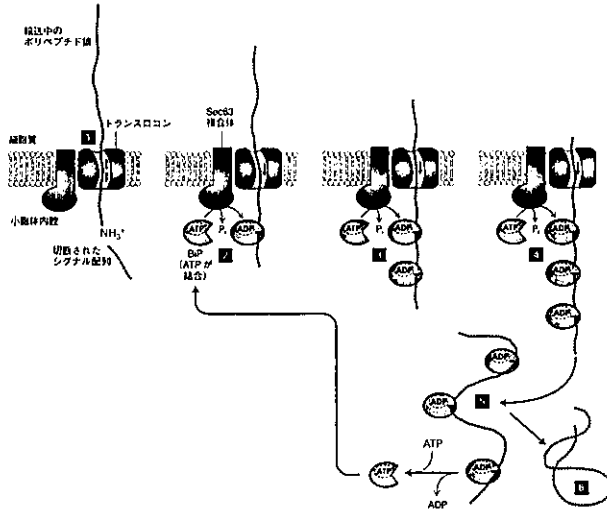


図 2.17: 翻訳後転送のメカニズム

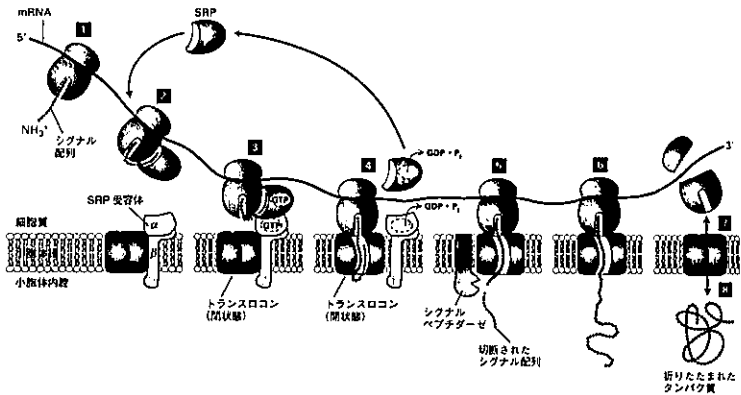


図 2.18: 翻訳時転送のメカニズム

Preamble	Packet header		フレーム タイプ	データ	CRC
	宛先 アドレス	宛元 アドレス			
8 bytes	6 bytes	6 bytes	2 bytes	46 ~ 1500 bytes	4 bytes

図 2.19: イーサネット・パケットの構造

号 3,4)、と同時に、合成しかかっているアミノ酸鎖はトランスコロンに吸い込まれていく。転送配列はシグナル・ペプチターゼによって切断され(番号 5)、合成が終了すると、リボゾームは分解し膜からはなれ、一方アミノ酸鎖は折り畳まれタンパク質となる。おそらく SRP と SRP 受容体が転送配列と小胞体の結合位置を結びつけている。

DNA にコーディングされている情報が活性化されると、対応するタンパク質が所定の位置で起動される。コーディング上は機能を特定する情報と、機能を配置する場所の情報(アドレス)がコーディングされていることになる。

2.2.7 パケット・ヘッダ

この転送配列の機能は、インターネットにおけるパケットの構造と類似している。インターネットにおいて最も基本的かつ、あらゆるところで利用されているイーサネットについて説明しよう。イーサネットとは、本来、1972 年~1973 年にかけて、米ゼロックスのパロアルト研究所 (PARC) においてロバート・メトカーフが中心となって開発したネットワーク・システムのことである。しかし現在、この規格の使いやすさから、規格を同じにした様々なネットワークが存在する。代表的なものとして、無線 LAN のパケットの構造が挙げられる。

イーサネットの基本的な考え方は、通信を相互に行うコンピュータ(ネットワークではノードと呼ぶ)が一本の電線につながっており、送信者が送信したデータを全コンピュータが受信できる仕組みである。そこには送信したデータを振り分けたり転送したりする物理メカニズムは存在しない。情報は届いてから、受信者が自分に届いたものかどうかを判断し、自分に届いたものではない場合、届いた情報を破棄する。このような仕組みにすることで物理的な構造が単純化できるのである。丁度、翻訳後転送のように、細胞質の広い空間の適当

な場所で合成されたタンパク質が、実際に配置される場所の膜構造によって認識され取り込まれるようなものである。

各イーサネットにつながったコンピュータが、自分に届いた情報かどうかを判断するために、各ノードにはイーサネット・アドレスというノード固有のアドレスを持っている。送られるデータは、最大でも 1500 バイトに小分けされ、パケットと呼ばれるデータ配列構造に構成され送信側からネットワークへ送られる。そのパケットの構造が図 2.19 である。パケットにはパケット・ヘッダが付され、ここに、送信先アドレス、送信元アドレスが含まれている。情報の受けては、この情報によって自分が取り込むべき情報かどうかを判断する。そして、取り込むべきパケットであると判断されると、パケットからパケット・ヘッダや CRC⁸が取り除かれ、本来のデータだけが残される。

2.2.8 小胞体タンパク質輸送

小胞体にはじまるタンパク質の輸送と細胞外への分泌はさらに高度なメカニズムを有している。図 2.20 はその様子を表した図である。前述のように小胞体の膜に結合したリポゾームはタンパク質を生産し、小胞体膜上、あるいは小胞体内へ送り込む (番号 1)。小胞体内部のタンパク質が小胞体表面に集まりその部分が小胞として出芽してちぎれ、細胞質内へ浮遊する (番号 2)。浮遊した小胞が複数あつまり融合してシスゴルジ嚢となる。一つのシスゴルジ嚢はその後、シスゴルジ、中間ゴルジ、トランスゴルジ、トランスゴルジ網へと形を変えて発展して行く。この過程を嚢成熟という (番号 4)。トランスゴルジ網の中のタンパク質は、そのタンパク質の性質によっていくつかのその後の分泌径路をたどる (番号 6~8)。番号 6 は、ある種の水溶性タンパク質の分泌で、常時分泌されるもの。番号 7 は、神経あるいはホルモンなどのシグナルなどを受けたときのみ分泌される調節された分泌である。また、番号 8 は、ゴルジ体や小胞体とは異なる別の膜構造であるリソソームへ輸送された加えられるタンパク質である。いずれも、トランスゴルジ網のある部分からタンパク質を納めた小胞が発芽し、細胞膜やエンドソームへ合体 (エンドサイトーシス) し、しかるべき

⁸CRC: Cyclic Redundancy Check / 巡回冗長検査。届いたデータが正しいかをチェックするためのコード。データを指定されたアルゴリズムで集積し 4bytes の値を計算する。この計算結果がパケット上の CRC 一致すれば、パケットのデータは正しいと判断される。

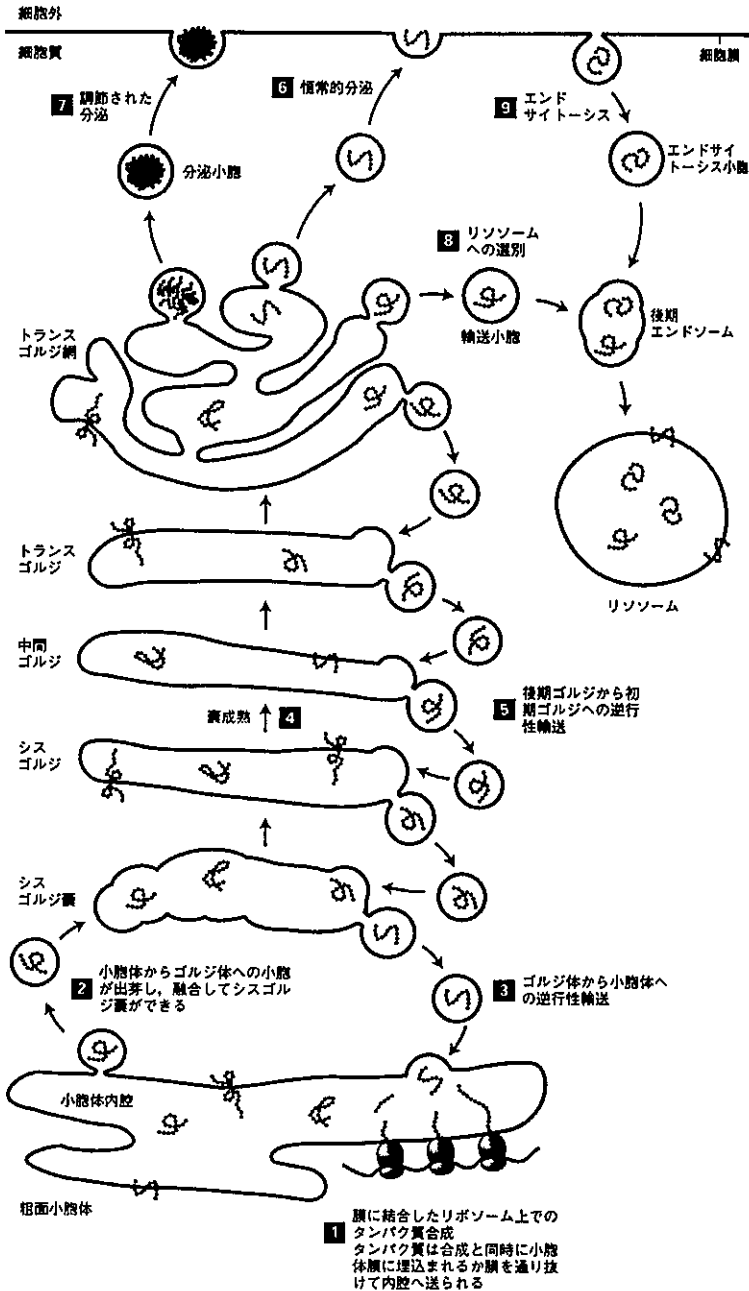


図 2.20: 生体膜構造による物質の輸送

ところへ輸送される。一方番号9は、細胞膜の表面からタンパク質を取り込むメカニズムであり、細胞膜が細胞質の方へ膨らみ、ちぎれて小胞となる(エンドサイトーシス)。この小胞が集まり、エンドソームを形成し、リソソームに取り込まれて行く。ゴルジ体の各過程とタンパク質は適切な場所があり、異なるゴルジ体に集まってしまった不適切なタンパク質は、逆行性輸送小胞によって適切なゴルジ区画へ戻される(番号5)。このように見ると、小胞体で生産されたタンパク質は、ゴルジ体において目的、用途、プロセスに応じて分類され、小胞に小分けされたものが必要かつ適切な場所へ輸送される様子が見て取れる。

タンパク質を区画分類しているのは膜構造であり、膜のもつ役割がいかに重要かが分かる。膜がタンパク質を区画分類できるためには、膜自身にタンパク質を認識するメカニズムや小胞を発芽させるメカニズムが必要であり、また、小胞が適切な場所へ移動できるということは、小胞の膜自身に移動先を認識する能力があるということである。次にそのメカニズムを見てみよう。

図2.21は、小胞体から小胞が出芽するプロセスを表したものである。(a)は出芽直前において関与しているタンパク質を表したものである。まず小胞によって輸送されるタンパク質は、小胞体内部に浮遊している水溶性積み荷タンパク質、および、小胞体膜上にある膜内在性積み荷タンパク質である。前者は小胞が出来るときに、膜表面にある積み荷タンパク質受容体によって集められる。次に、小胞の出芽のスタートを決めるタンパク質として、GTP結合タンパク質がある。そして、小胞としての丸い形状を形成するためのタンパク質として被覆タンパク質がある。v-SNAREタンパク質は、小胞が輸送先の膜上にドッキングするときに利用されるタンパク質である。

(b)はこれらのタンパク質によって出芽するプロセスを表したものである。出芽のスタートは、水溶性 Sar1 と小胞体膜表面にある Sec12 が相互作用することにより、Sar1 上の GDP が GTP と交換され、GTP 結合タンパク質に変化するところから始まる。GTP 結合タンパク質は疎水性 N 末端を外に広げ、小胞体膜上に Sar1 を係留する(番号1)。Sar1 は被覆タンパク質の結合部位となる(番号2)。現在小胞体での出芽を例としており、小胞体の被覆タンパク質 COPII による例である。被覆タンパク質が Sar1 に結合することによって、小胞の球面が形成され、最終的に小胞体膜から切り離され細胞質内に浮遊する。Sar1 が加水分解することにより、被覆タンパク質の結合は解除される(番号3,4)。

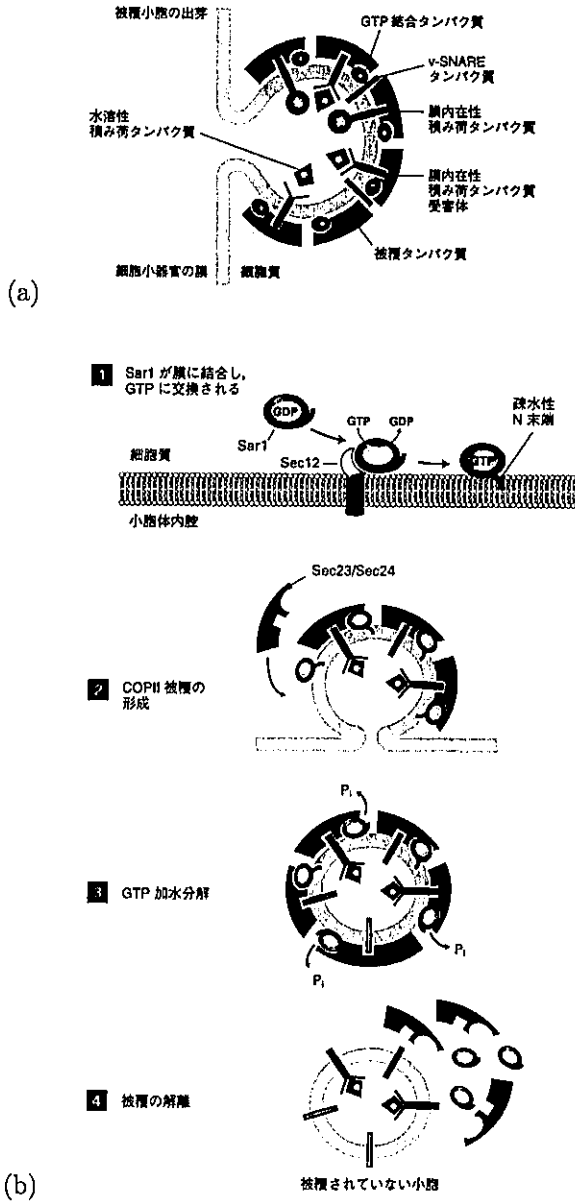


図 2.21: 膜構造からの出芽のメカニズム

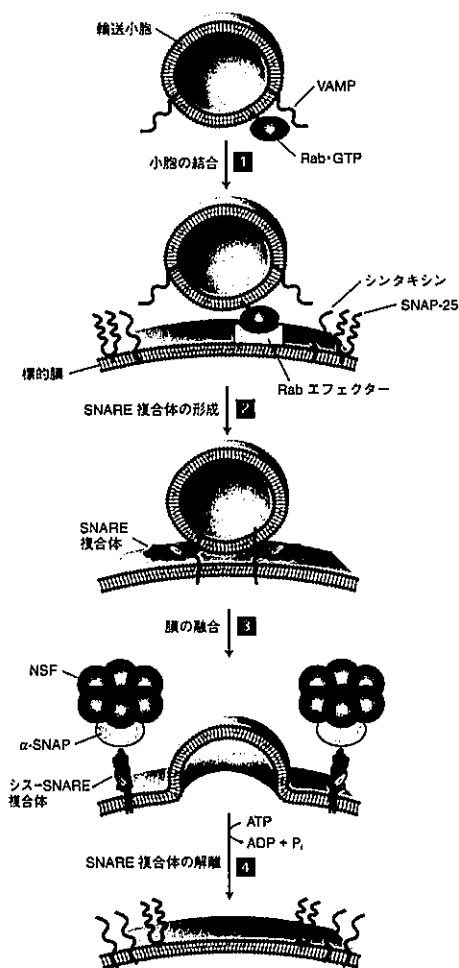


図 2.22: 膜構造への融合のメカニズム

このようにして生まれた小胞は、目的の膜構造を探し当て合体しなければならない。そのときのアドレスの役割を果たすのが、Rab・GTP である。情報科学的な目で眺めれば、Rab・GTP は、出芽の初期の段階で、積み荷タンパクの種類によって、小胞表面に接続されるかあるいは積み荷タンパク質受容体とペアで存在するかしなければならないはずだが、どの時点で小胞に接続されるのかよくわからない。この Rab・GTP と、相互作用する Rab エフェクターとよばれるタンパク質が標的の膜上に存在する。図 2.22 は、小胞が膜に融合する様子である。この二つのタンパク質の相互作用によって小胞は目的の膜へ接近する (番号 1)。標的の膜には、この図では、シンタキシン、SNAP-25 というタンパク質が存在している。これらは t-SNARE タンパク質という種類であり、v-SNARE と結合し、SNARE 複合体を形成し小胞を標的膜上にしっかり固定する役割を果たす (番号 2)。一方、小胞上の v-SNARE タンパク質にも何種類もあり、この場合は VMAP と呼ばれるものである。

SNARE 複合体が形成されるとすぐに小胞と標的膜が融合を始める (番号 3)。しかし、このメカニズムはよくわかっていない。小胞の融合のあと、NSF、 α -SNAP が SNARE 複合体と結合し、ATP の加水分解によって、SNARE 複合体をほどき、t-SNAP が新たな小胞融合で利用できるようにする。

最後に、図 2.20 において、細胞外からの物質の輸送について説明しよう (番号 9)。この輸送では、細胞膜が細胞質の方へ陥没し、細胞外の物質を取り込み小胞を作る。このメカニズムは、基本的に 2.21 と同じメカニズムと考えてよい。ただし、働いている被覆タンパク質がクラスリンに置き換わる。これをエンドサイトーシスと呼ぶ。小胞は集まり、後期エンドソームになり、これがしかるべきリソソームに取り込まれる。

エンドサイトーシスで取り込まれるのはタンパク質をはじめとした分子だけではない。細菌やウイルスといった大きな物質までもこの方法で取り込む。このようなエンドサイトーシスをファゴサイトーシス⁹と呼ぶ。これによって取り込まれたエンドソームをファゴソームとよぶ。また、エンドサイトーシスによって細胞外液を非特異的に取り込むこともある (ピノサイトーシス)。

図 2.23 は、エンドサイトーシスされた物質がリソソームに送り込まれるところである。後期エンドソームでは、細胞外から入ってきた小胞とゴルジ体から

⁹phagocytosis 食作用

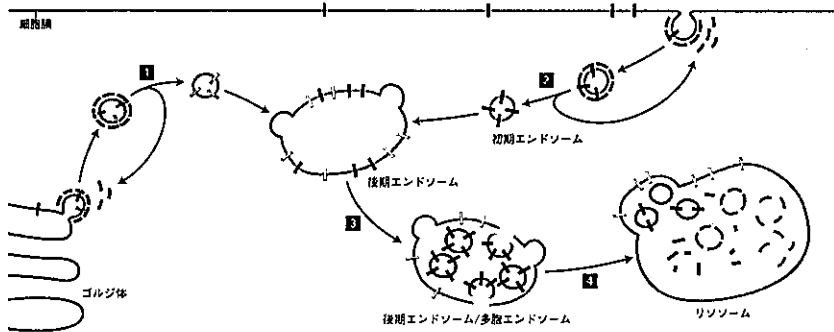


図 2.23: エンドサイトーシス

分離した小胞とが合わさっている (番号 1,2)。この後期エンドソームの膜上には、細胞外壁上の膜内在性のタンパク質と、ゴルジ体上の膜内在性タンパク質が混在していることになる。このうち、細胞外壁上の膜内在性タンパク質のある部分が、後期エンドソーム内腔へ発芽する (番号 3)。後期エンドソームはリソソームへ吸収されると、この内腔の小胞は分解される。たとえば、外界から取り込まれた細菌等も一緒にリソソームで分解され無害化される。

ところがこの理屈を逆手に取って増える、狂犬病ウイルスの例が、図 2.24 である。エンドソームは通常の間能として H^+ イオンをエンドソーム内へ組み上げている。これによりエンドソーム内の pH が低下し、狂犬病ウイルスの糖タンパク質が構造変化を起こし、ウイルスの膜と、エンドソームの膜が融合し破壊する。これにより狂犬病ウイルスの RNA が細胞質内へ放出される (番号 4)。放出された RNA は、細胞質内のリボヌクレオシド 3 リン酸を用いて複製され (番号 5)、また、mRNA をも作る (番号 6)。mRNA はやはり細胞室内のリボソームにより、ウイルスを構成するタンパク質に翻訳され、ウイルス内部の構造が生成される (番号 10,11)。同様、小胞体上でもウイルスの膜タンパクが合成され、番号 8,9 の径路をたどって、細胞膜上へ展開される。最後に、膜上において、エンドサイトーシスのように、細胞外液側に完成したウイルスが出芽する (番号 12,13)。

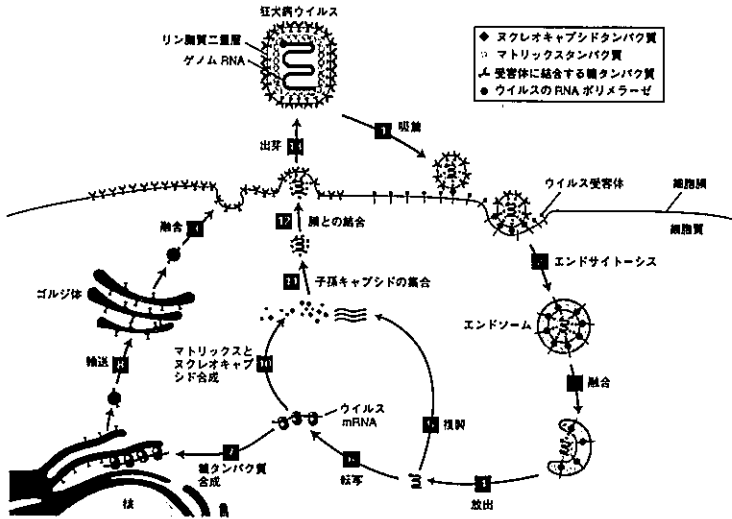


図 2.24: 狂犬病ウイルスの増殖メカニズム

2.2.9 細胞骨格

生体膜は細胞の中を、物質の出入りが制限された、小胞と呼ばれる領域に分解する。その領域への出入りを制限することにより領域に役割が与えられる。この膜構造による領域は細胞内で、変形し、移動し、また分離したり融合したりする。このような変形、移動、融合によって領域内のタンパク質その他の物質の、分離、混合、あるいは外部への分泌、外部からの取り込みが行われる。

だが、そもそも、こういった小胞は細胞質内を無意味に浮遊しているのではなく、目的値へ向かって浮遊して行く。また、細胞やゴルジ体など様々な膜構造は単なる球形ではない。「アメーバのように」という表現があるが、細胞は触肢を伸ばし、移動するものがある。また、細胞が分裂する時、複製された染色体は細胞の両側へ引き寄せられる。こういった細胞の動作、形状、あるいは物質や小胞の移動には細胞内に張り巡らされた繊維状物質、いわゆる細胞骨格が大きな役割を果たしている。

細胞骨格を構成する物質は3種類が知られている。マイクロフィラメント、微

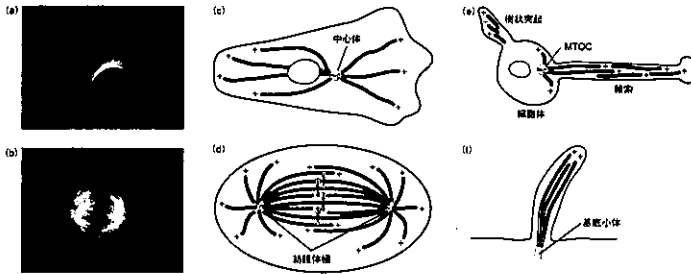


図 2.25: 細胞周期や種類による微小管地図

小管、中間径フィラメントである。マイクロフィラメントは細胞の移動、繊毛の動作に係わっている。微小管は細胞内に張り巡らされ、ゴルジ体やミトコンドリア、細胞内組織の移動や選別、分裂のときのコピーされた染色体の分離などに活躍する。このなかで、細胞内の組織の選別、輸送に係わる微小管について着目してみよう。微小管は細胞の状態や周期によってその構造を異にする。細胞の状態によって分解され、新たに張り巡らされる。図 2.25 の (c) は、細胞分裂から次の細胞分裂の間（間期）の細胞における微小管のネットワークの一例である。中心から四方に伸びているのが分かる。微小管はたいがい、細胞のある中心から四方に伸びる構造をしている。その中心のことを微小管形成中心（MTOC microtubule-organizing center）と呼ぶ。間期の細胞ではおおよそ、MTOC は核の近くにある。(d) は間期に対して細胞分裂時（有糸分裂）の微小管の様子である。微小管は、細胞の両側にある二つの極から放射状に延び、複製された染色体につながる。このあと微小管は染色体を引っ張りながら急速に短くなり、二つの核を作る。(e) は、神経細胞の場合で、時には 1 メータ以上にもなる軸索（ニューロン）の中には、微小管が入っている。これも軸索の根本にある MTOC から微小管が伸びることによって出来る。(f) は繊毛等の軸を表している。規定正体と呼ばれる MTOC から微小管が伸びている。微小管には方向があり、それぞれの方向を + 末端、- 末端と呼ぶ。多くの場合、微小管は MTOC を - 末端として + 末端の方向へ伸びる。なぜか、神経細胞の樹状突起においてはこれがない。

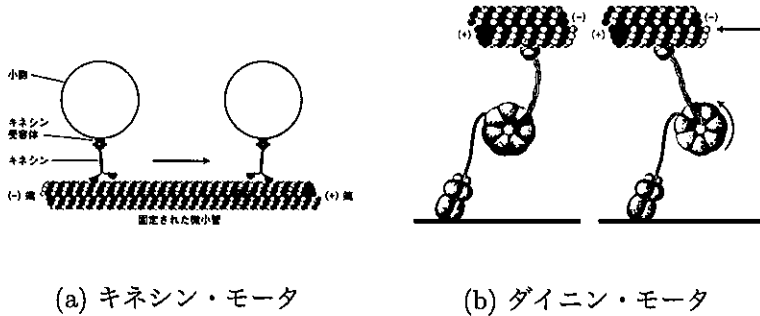


図 2.26: 微小管モータ

図 2.26 は、微小管と相互作用して動作する二種類のモータ・タンパク質である。キネシンは二つに分かれた頭部を微小管上と総合作用させ、微小管の-末端から+末端の方向へ移動して行く。キネシンのしっぽには、運ぶべき細胞内の小器官を接続する。一方、ダイニンは、微小管の+末端から-末端の方向へ移動して行く。図 2.26(b) ではダイニンの 1 回のパワーstroke で中心のドーナツ部分が回転して、腕の角度が変化する。

このような二つのモータによって細胞内に張り巡らされた微小管上を様々な細胞内小器官が移動して行く。図 2.27 では、既に述べたゴルジ体、小胞、エンドソームなどの小器官が二つのモータを利用して、微小管を伝って移動して行く様子が描かれている。

ここでいくつかの疑問がわいてくる。確かに細胞骨格は MTOC から放射状に伸びているので、微小管に移動を頼っている限り、細胞内の物質は細胞の中心と周辺の移動しか出来ないのではないか。おそらくそんなことは無いだろう。ここからは私の想像であるが、一つの考え方は、これはスター型ネットワークの考え方である。ある物質が目的地を目指す時、同じ微小管上に目的地がある場合は、キネシン、または、ダイニンを使ってそこへ移動すれば良い。同じ微小管上にない場合は、とりあえず中心 MTOC を目指す。MTOC にはあらゆる微小管が集まっているので、どれか目的地を目指す微小管があるはずだ。その微小管に乗り換えて目的地を目指す。だか情報科学的には理にかなった考え方であるが、現実的には遠回りである。もうひとつの考え方としては、途中で微

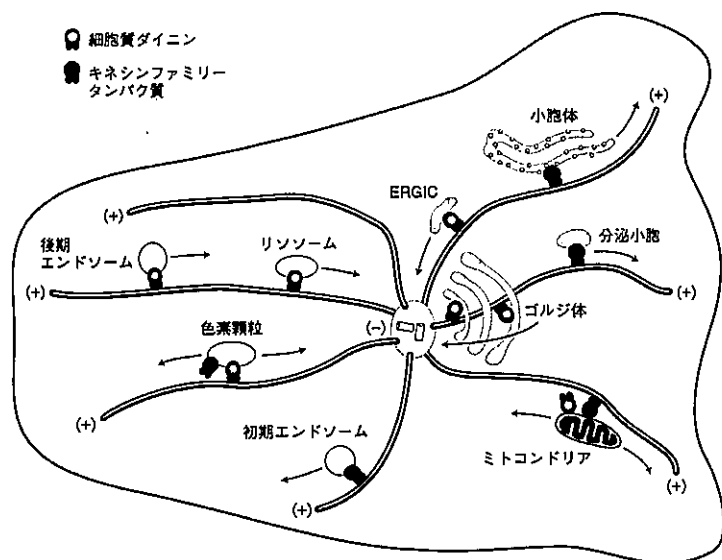


図 2.27: 微小管と物質移動

小管を乗り換える方法があるのかもしれない。目的地がどこにあるかというのは、目的地から発つされる何らかのシグナルの濃度勾配をたよりにその方向を目指す考え方で、濃度勾配が微小管の方向であれば微小管を伝って行くが、もし微小管と直角に近い方向に勾配がある場合は微小管を渡り歩くかもしれない。この二つの方法の折衷ということも考えられる。特に神経細胞やある種の筋肉細胞のように極端に細長く変形した細胞内では、濃度勾配が末端まで伝わらないかもしれない。そういった場合、たよりは微小管のみということになる。このように考えると、物質の輸送方法は複数あり得るのではないだろうか。

もう一つ注目すべきことは、図 2.16 の右側への輸送経路である、小胞によらない、タンパク質の輸送と配置である、水溶性タンパク質やイオンといった小さい物質は微小管を頼らずに細胞質の中を自由拡散して行く。自由拡散した物質は細胞質の中に濃度勾配を作る。これが様々なシグナルとなって、細胞内の新たなプロセスが起動される。このシグナル伝達は、微小管による明示的な輸送に比べアナログ的、連続的で、ある場所から細胞質内全体へブロードキャストされ、徐々に濃度が上がって行く。どこからがシグナル on なのか、それが分かる場合もあれば、まったく最後まで、連続的、アナログ的なコントロールである場合もある。このような自由拡散は細胞質内のみならず、ゴルジ体や小胞体、リソソームの中でもおこっているはずだ。そのときに、膜構造の形状は自由拡散に少なからず影響を与えるだろう。例を挙げてみれば、神経細胞の軸索のような細長いパイプのなかでは、自由拡散もおこりにくいかもしれない。ゴルジ体も扁平な形や襞のような形を多数見かける。これら狭い領域が毛細管現象的な状況を作り出すかもしれない。こういった膜の形状変化がタンパク質やその他物質の拡散を制御し、物質の選別につながっているのではないだろうか。

こういった膜構造に囲まれた自由拡散する空間のことを「環境」とよぼう。実は「環境」という用語は分子生物学の用語というより、情報科学の用語である。情報科学において、環境とは、あるプログラムを動作させたり、何らかのコンピュータを用いた作業を行うために必要になる、他の小さなプログラム・ツール、オペレーティング・システムなどや、あるいは様々な初期設定の集合体である。こういった環境があつて初めてプログラムは動作する。プログラムだけではなく、通信が可能となるためには通信に必要な環境が整っている必要がある。生体膜は生体膜内の物質が機能するための環境を与えている、と言え

る。細胞骨格は、その環境内の物質の伝達をになう微小管、膜構造の形状変化、強いては環境変化を起こすマイクロフィラメントなどである。

2.2.10 パケット・エンキャプスレーション

先に、タンパク質の輸送配列とパケットのアドレスを含むヘッダ構造の比較を行った。両者ともデータ本体にとっては不要な部分であるが、輸送するという点において必要な情報として輸送するときだけにアタッチする信号である。この考え方を発展させて行くと、パケット・エンキャプスレーションという技術がある。パケット・エンキャプスレーションは単なる情報の伝達の方法としての比較だけではなく、先に述べたアナログ的・連続的なコンピューテーションの一つの示唆を与えてくれる。この点について細胞のパラダイムと比較してみよう。

インターネットにおけるデータ転送技術を理解するためには、まずは郵便のたとえが良いだろう。郵便局に、内容についての簡単な申告を添え、表に配達先住所を書いた小包（パケット）を持って行く。しかるべき料金を払うと、郵便局は宛てどころにこの小包を届けてくれる。このように内容物を小包に入れる、ということがエンキャプスレーションである。そして、小包に書かれた住所、申告内容がパケット・ヘッダである。郵便局はこのパケット・ヘッダの内容に従って配送する。申告に冷蔵が必要であるとあれば、冷蔵車を使って転送する。申告にワレモノと書かれていれば、衝撃や加重をなるべくさけて配送する。そして、書かれた住所に配送する。基本郵便局はパケット・ヘッダに書かれていること以上に小包の内容のことは関知しない。小包の内容が出来るだけブラックボックス化されているということが配送の効率向上に重要なのだ。逆に、配送の依頼者にとっては、住所と申告書を書き上げ、料金を払ってしまえば、あとは、先方から、荷物が無事に届いたという連絡（ACK¹⁰）を待つのみである。その間の配送の方法は関知しない。つまり配送方法は出来るだけブラックボックス化されていることが重要である。

小包を受け取った郵便局はまずなにをするであろうか。おそらく配送先の都道府県ごとに大まかに分類する。末端の郵便局ではもっと簡単に、県内と県外

¹⁰ACK=acknowledge

へ分けるだけかもしれない。そして分類ごとにトラックに乗るくらいのカーゴにまとめられる。カーゴには「県外」「県内」「国外」といった宛先が書かれている。現在ではそれはバーコードかもしれない。つまり、受け取った小包をまとめてカーゴという小包でエンキャップスレーションしている。「県外」へ行くカーゴは、トラックではなく、鉄道輸送が使われるかもしれない。JR 貨物の基地へはトラックで運ばれ、そこからコンテナに載せられ、貨物列車で目的県の基地へ運ばれる。コンテナにも行き先を示す分類コードが振られているだろう。ここでも、カーゴがコンテナにエンキャップスレーションされている。基地ではコンテナからカーゴがおろされ、また、郵便局の配送センターではカーゴから小包がとりだされる。宛先に着くときには、小包だけが到着する。

途中、なぜトラックを使う必要があるのか、あるいは鉄道、飛行機が使われるのかは、依頼者の事情ではなく、配達者の事情で決まる。道を輸送するときにはトラック、遠方へ大量に送りたいときには鉄道かもしれない。陸路がないときは飛行機といった具合に。そして面白いのは輸送手段が変われば、パッケージもカーゴやコンテナといったように変わる。ある輸送手段を利用するとき、その輸送手段に適したパッケージに詰め込み、そしてその輸送を終え、おろされるときにパッケージから中身がとりだされる。

インターネットも同様な事情がある。まず、物理的な通信手段が幾つもある。思いつくだけでも、先に挙げたイーサネット、銅線、光ファイバー、無線 WiFi、携帯電話、衛星通信、海底ケーブル、といった具合である。それぞれの物理通信手段ごとに通信方法が異なる。たとえば、私の PC からヨーロッパにあるウェブサイトにアクセスしたとしよう。ほんの一瞬の出来事であるが、サイトの情報が私のところに届くまでに、実に様々な手段で、大西洋、太平洋を渡って私の家までやってくるのだ。このとき、情報の転送は、郵便小包と同じ原理が使われる。

ヨーロッパのウェブサイトは、私の家の PC のアドレスをヘッダに書き込んだ IP パケットを作り、それをネットワークに送り出す。IP とは Internet Protocol の略で、IP アドレスという世界で一意に決まり、かつ配達可能な性質を持ったアドレス方式を定義している。私の PC のアドレスがヘッダに書き込まれた IP パケットがヨーロッパのウェブサイトから送り出される。いわば、住所を書いて郵便局にもっていった小包のようなものである。

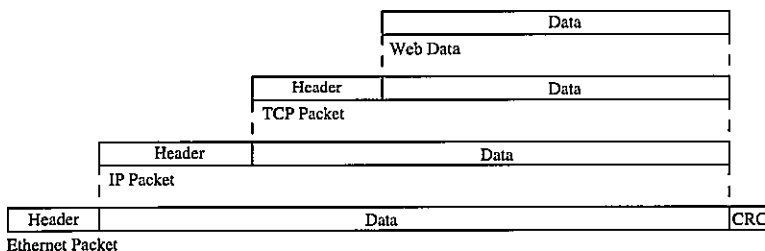


図 2.28: エンキャプスレーション

情報の世界では、物理的な通信手段の他に、通信を支える機能や性質（プロトコル）が様々なものがある。たとえば、暗号化して送ることであったり、圧縮して送るといったことであったり、信頼性を保証する通信手段がある一方、信頼性を保証しないが通信の手軽さやスピードを重視する通信手段がある。このような様々なプロトコルを組み合わせることが出来る。組み合わせた時、プロトコルに対応したパケット、およびパケット・ヘッダが存在する。それらを組み合わせごとにエンキャプスレーションするのである。

ウェブサイトをアクセスしたときの通信では、TCP というプロトコルと IP というプロトコル、および、通り抜けてくる様々な物理ネットワークのプロトコルを使う。少なくとも 3 つのパケットがエンキャプスレーションされている。図は 2.28 は TCP と IP および、物理ネットワークとしてイーサネットがエンキャプスレーションしたときの図である。IP パケットは、イーサネットにとってはデータとして扱われる。TCP パケットは同様に IP パケットのデータとして扱われる。IP プロトコルは、先にも述べたが IP アドレスという世界中のインターネット上のコンピュータに一意かつ、その場所までの通信経路を検索可能とするアドレスを提供し、実際に IP アドレスが与えられた IP パケットを相手先まで届ける。しかしこれは分かりやすく意識した言い方である。厳密に言うと若干複雑である。送信したパケットが、指定した IP アドレス以外のノードに届くことはないことが保証されている。逆に、送信したパケットが指定した IP アドレスに必ず届くことは保証されていない。ネットワークの途中の混雑状況によっては処理が間に合わず、パケットを捨てる可能性がある。ま

た、同一 IP アドレスから同一 IP アドレスへの通信経路は必ずしも一意とは限らない。混雑状況などの様々な条件によって毎回異なる通信経路を通る可能性もあり、そうすると、後から発進したパケットが先のパケットを別経路で追いついて先に届く可能性もある。IP プロトコルは送信したパケットを出来る限り宛先に届けようとするが、努力も根尽き果てるときもあるのである。

TCP とは Transmission Control Protocol の略であり、IP に対して通信における信頼性を保証するプロトコルである。信頼性とは、送ったデータが送った順番で到達することを保証するということである。途中のデータが抜け落ちたり、二度届いたりすることがなく、順番が入れ替わって届くこともない。また輻輳制御もする。ネットワークの途中の混雑状況に合わせて、送受信スピードをコントロールし、ネットワークが詰まってしまい、全く情報が届かなくなってしまうことを防ぐプロトコルである。

逆に TCP では相手先のアドレスへパケットを届ける機能は存在しない。IP と TCP を組み合わせることによって信頼性をもった配達が可能となる。このように異なる機能を異なるプロトコルが担い、それぞれが独立したパケットヘッダを持っているということは、機能の組み合わせをよりフレキシブルに出来るというメリット、および、一つのプロトコルを考えると、考慮すべき事項が絞られよりの確かな機能実現が可能となるというメリットがある。だいたい前に戻るが、情報の透明性について説明したときの図 2.5 および図 2.6 を思い出してほしい。同様な図式で TCP, IP, イーサネットを描いてみると図 2.29 となる。

この図では、発進されるデータがまず (1) の Web サーバで、上から下へと送り出され、TCP レイヤ、IP レイヤ、物理レイヤと通り抜けて行く。一方、(3) PC では、受け取ったパケットを物理レイヤ、IP レイヤ、TCP レイヤと逆順で送りデータを受け取る。このようにプロトコルは各コンピュータ (ノード) では同じ順番で積み上げられているのである。これをプロトコル・スタックと呼ぶ。上から下への矢印、encode のプロセスで、通り抜ける一つのプロトコルに着目すると、上から下りてくるパケットはそのプロトコルにおいてデータと見なし、しかるべき処理をしたあと、自分のプロトコルに対応したヘッダをつけ (エンキャプション) 下のレイヤへと送る。逆に上向きの矢印、decode のプロセスでは、一つのプロトコルに着目すると下からくるパケットのヘッダを取り外し、しかるべき処理をした後に上のレイヤへ送る。

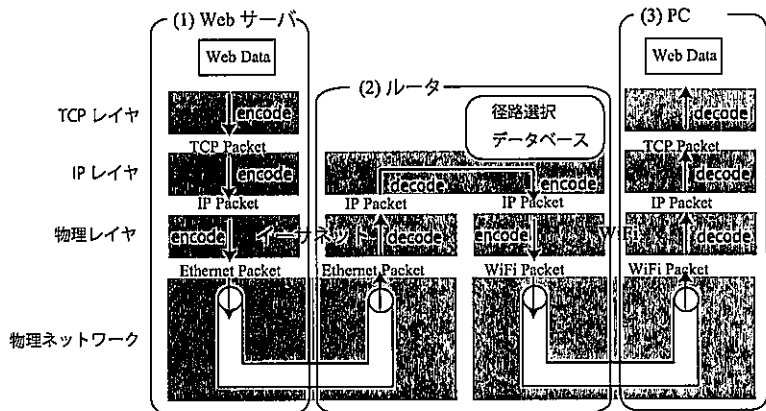


図 2.29: ネットワーク階層構造

下のレイヤは上のレイヤが何かということには関知しない。TCP レイヤはなく、データがいきなり IP レイヤに送られたとしても IP レイヤは適切に処理できるはずである。一方、上のレイヤは下のレイヤに対しては、ある一定の条件を課すのみで、その条件がどのように実現されているかは関知しない。TCP レイヤは IP レイヤに対して「与えた IP アドレス以外には届けないこと」ということを要求しているのみである。ネットワークにおけるインタフェースの透明化と機能の抽象化である。この点で興味深いのは、(2) のルータのところである。IP プロトコルは複数の物理ネットワークを繋ぎ合わせ、遠方のノードへデータを送る役割があることからルータ機能を有している。しかしこの図では TCP レイヤはルータには載っていない。TCP は IP がどのように働くかは全く関知しないからこれで十分なのである。TCP から見れば IP プロトコルは (1) から (3) へダイレクトにつながった一本の管のように見える。

ルータにおいて IP プロトコルは、受けたパケットの宛先を調べ、径路選択データベースより適切な宛先への径路を検索し、新たな物理ネットワークへ送り出す。このルータを境にして物理レイヤおよび物理ネットワークが変化している。IP プロトコルが物理ネットワークに要求する事項が満たされていれば、物理レイヤは何でも良いのである。これもネットワークにおける透明化と抽象

化の一例である。

ここで、IP ではクオリティーは 100% 求めず、IP を実現した後に TCP においてクオリティーを保証するという考え方は、実はゴルジ体におけるタンパク質分泌にもある。図 2.20 において、番号 5,3 における逆行輸送はゴルジ体がシスからトランスへ進化して行く間に、タンパク質の分類に生じたエラーによって混じった不純物をしかるべき配置へ戻す仕組みである。クオリティー 100% ではない正規の分泌過程と、この逆行輸送メカニズムの二重レイヤになっている。細胞内にはこのようなとりあえず行っておいて、後で訂正する、というプロセスがいくつも出てくる。たとえば、図 2.17、図 2.18 において、小胞内腔へタンパク質が送り込まれた後、タンパク質が本来のタンパク質の形に折り畳まれる。しかし、このときに間違っただけ異なる折り畳まれ方をする可能性があり、これをチェックし、間違っただけ異なる折り畳まれ方をするタンパク質が幾分現れ、これは分解破棄される。DNA が転写されるときにも同じように誤り訂正が何重にも働く。

工業生産におけるプロダクトの品質管理 (Quality Control) においては高度成長期からこのようなことは行われていた。半導体の生産は、プロセスが精密になればなるほど歩留まりが下がる。プロセスをもう一度見直し歩留まりを上げことはもちろん必要であるが、製品出荷時にテストを行い不良品をはじけば良い。歩留まり 100% を狙うよりは 95% くらいにしておいた方が、実際は生産効率はよい。

高度な三次元グラフィックをリアルタイムに描画するための高速計算装置であるグラフィック・アクセラレータは、1000 × 1000 くらいの高精細画像の中に 1 ピクセル程度計算エラーが生じ異なる配色になっていたとしても目立ちはしない。それよりも、高速な計算をある程度の価格で提供できることの方が重要である。だが、最近このグラフィック・アクセラレータを、気象予測などの高度な微分方程式の解法などの高速かつ大量の数値計算マシンとして利用することが多くなってきた。このような用途では 1 つの計算間違いが全体に影響し命取りとなる。そのための方法として、チェックポイントと呼ばれる計算の区切りのところで検算を入れる。検算が間違っていれば、もう一度やり直せば良い。やり直し回数は $1000 \times 1000 =$ 百万回に 1 回程度であるので、これでも高速に計算が可能である。

2.3 プログラミング・パラダイム

2.3.1 細胞とノイマン型コンピュータ

分子生物学を説明するならば、まず真っ先に説明するであろう、セントラルドグマについて私はあまり説明しなかった。セントラルドグマとは、遺伝子の構造とその発現のメカニズムである。DNAのコーディングがどのようになっている、DNAがどのようにして複製され遺伝するかまた、DNAがどのようにしてタンパク質に翻訳されるかということがセントラルドグマである。セントラルドグマを細胞システムの中心的パラダイムにすえることによって、主たる遺伝情報とその従として生まれる細胞のアクティビティーという二分性が際立ってしまう。しかし実際に、情報科学的な視点に立って眺めてみると、情報の伝達と変形はほぼすべてタンパク質が行っていると言っても過言ではない。細胞の内部で情報を担っているのはタンパク質である。まずはそこに目を向けるべきだったのである。ここにきて、DNAはその中でどのような役割をしているのか考えてみよう。それは、「情報の二つの形態 核酸とタンパク質」の項で述べたように情報の非活性な状態を作るためである。特に細胞分裂の時にはあるコンシステンシーの保たれた状態で情報を一旦停止しなければならない。そのための記録媒体がDNAである。

もう一つの役割は情報の論理的スイッチングである。もとよりタンパク質は物質であり、タンパク質同士の相互作用は物理法則に則る。逆の言い方をすれば、タンパク質同士の相互作用だけに任せておけば物理法則以上のことはおこらないということだ。例えば細胞内へ送り込まれたタンパク質Aをシグナルにタンパク質Bを生産するメカニズムが欲しいとしよう。さらに、結果として生産されるタンパク質Bはある細胞ではタンパク質Cであることが望まれるとしよう。BとCの間には物理化学的な共通性はなにもない、あるいは人間が作りたいDというタンパク質でもよい。AとB,C,Dそれぞれの間にも物理化学的因果関係が見いだせない状況の中で、化学変化だけでAがB,C,Dを、時と場合、細胞の違いによって作り出すことは困難である。

これを可能にするのがセントラルドグマである。セントラルドグマの機能を一言で言うならば「何らかのタンパク質を認識したときに遺伝子のある部分

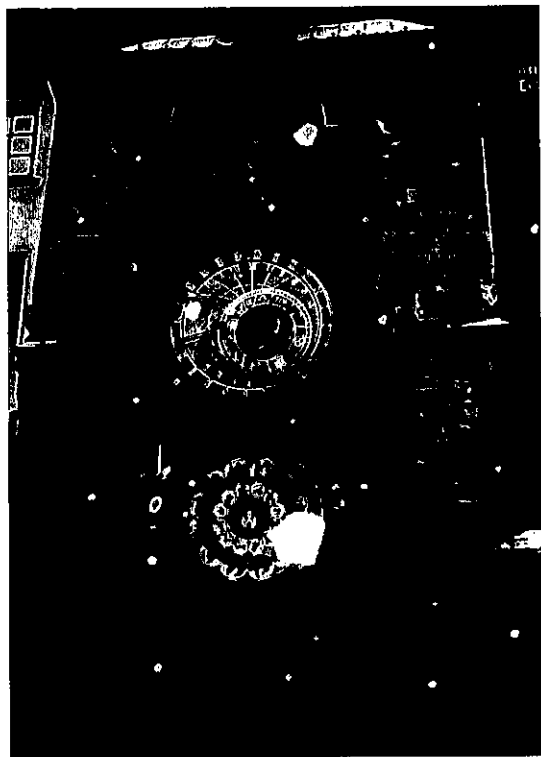
を読み出し翻訳する」というメカニズムである。タンパク質 A を認識するタンパク質を準備し、そして、そのときに呼び出される領域に B,C,D のコードをコーディングしておけばよい。コーディングの方法は純粋に塩基配列によるものなので、任意のタンパク質の合成が可能である。つまり直接的な物理化学的反応によらなくても、任意のタンパク質から任意のタンパク質を作ることが可能となる。

たとえて言うならば百貨店のインフォメーションである。お客が何を買いたいかインフォメーションに聞くと適切な売り場を案内してくれる。もしインフォメーションがなければお客は自分の少ない知識と情報で売り場を探し当てなければならない。このお客の知識の限界がタンパク質のもつ物理化学法則である。インフォメーションが機能し始めるとそれはただの便利な案内役ではなくなる。もし百貨店がなんらかのフェアを開催しようとしたとき、あるいは、様々な売り上げの状況から、ある売り場のプロモーションが必要と考えた時、インフォメーションの対応を変えればよいのだ。いままでのお客は彼ら自身の知識の限界を超えインフォメーションの対応の変化に合わせて、百貨店の目的と戦略に則った行動をする。

物理化学法則を書き換えることは出来ない。しかし様々な操作を物理化学法則から開放した「論理的スイッチ」の本質的なことは、書き換えることが出来るということだ。ある細胞ではタンパク質 A をシグナルとしてタンパク質 B を生産していたところを、その細胞が別の細胞から形質をもらい突然変異をしたとする。するとタンパク質 A のシグナルからタンパク質 C を生産するようになる。突然変異や交配による遺伝といういわゆる生命とその進化の本質が確立される。

そもそもコンピュータという機械が誕生した背景には同様な現象がある。コンピュータ誕生以前にも古来から伝説的な機械たちがある。プラハの市庁舎にある天文時計オルロイは 1410 年に時計職人であったカダンのミクラシュ (Mikuláš of Kadaň) とプラハ・カレル大学の天文学教授であったヤン・シンデルによって制作され、その後多難な歴史のなか何度にもわたる修復、改造を経て現在に至る。文字盤は、外周が日没が 0 時とする古チェコ時間、内周がプラハ現地時間そして、赤道十二宮の記号をあしらった小さい環で出来ている。

1783 年にマリー・アントワネットがブレゲに発注したとされる懐中時計は完



AKIKO SATO

図 2.30: 雪のオルロイ

全なグレゴリウス暦を実行する機械式時計である。それだけではなく、重力によるゼンマイの狂いや落下による衝撃の吸収などの様々な実用的な機能などブレゲの発明を懐中時計に納めた。だが完成したのは実に発注から44年を経て1827年のことである。依頼者はすでにフランス革命の嵐の中断頭台に消えていた。

時計というのは、暦や時間という宇宙を構成する重要なパーツを司ると同時に、複雑で高度な機能、それでいて日常性も兼ね備え、エレガントかつコンパクトな形状が求められる。時計職人の数々の伝説を生んだ。だが機械としてのエレガントさが求められたものは他にも、オルゴール、自動機械人形（オートマタ）といったものがあつた。

それでもこういった機械はお金持ちの趣向品でしかなかった。19世紀は産業、経済の世紀であり、ワットによる蒸気機関の発明により機械がその主役となった。最初に機械にまかされた仕事が紡績である。特にジャガード織機において、縦糸と横糸の織をパンチカードに記憶し、送られるパンチカードによりループを動かし目的の織を実現する方法は機械式メモリである。

このころイギリス人の数学者チャールズ・バベッジ（1791年～1871年）は当時計算には欠かせなかった数表の完成度の低さは人力によるものと結論し、これを機械化しようと考えた。演算を行う階差機関、さらにパンチカードによるプログラム機構を備えた解析機関を発明した。彼の機械式計算機は完成には至らなかったが、1989年、ロンドン・サイエンスミュージムは当時の設計と技術水準によって階差計算機2号機を制作した。図2.31¹¹がそれであるが、実際に動くことが示され、バベッジの設計の正当性が証明された。

産業革命は大量生産と大衆文化を生み出す。そのための原料、資源を求め、各国がしのぎ合ひ、やがて前世紀の二つの大きな戦争へとつながって行く。戦争も機械化され、巨大軍事力と多量の兵士が投入され、敵の顔が見えない戦争へと発展して行く。人と人が戦うのではなく、機械と人、機械と機械が戦う。その中の頭脳部部品として開発されたのがいくつかの計算機である。コンラッド・ツーゼ（1910年～1995年）がいる。この大戦中ツーゼはZ1～Z3、S1、S2という多くの計算機を残している（図2.32¹²）。Z3はプログラマブルだったと言

¹¹A photo of the Difference Engine constructed by the Science Museum based on the plans for Charles Babbage's Difference Engine No. 2 Photo by User:geni wikipedia

¹²Nachbau des Z1 im deutschen Technikmuseum in Berlin

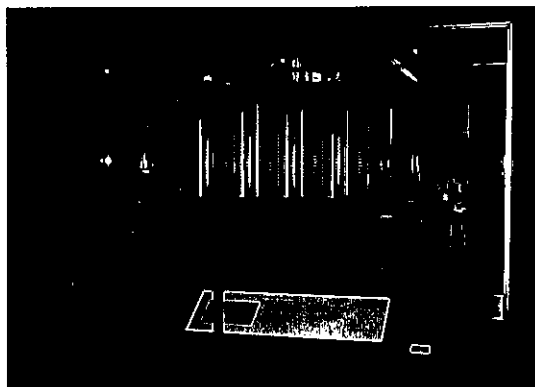


図 2.31: バベッジの階差機関 2 号機

われている。バベッジの機械式計算機はすべての情報伝達が運動力学的な構造によってなされる。ツェーゼの計算機にはリレーという電気的なものが入り入れられた。リレーとは磁石の力によってスイッチを on/off して電気の導通をコントロールするものだ。機械式だが電磁気的な構造をもつため完全な機械式ではないかもしれない。

ただ、真空管による回路とリレーによる回路にはさほど差はない。図 2.33 は

表 2.2: NAND 回路の動作

$$\text{出力} = \neg(\text{入力 A} \cdot \text{入力 B})$$

入力 A	入力 B	出力
0	0	1
0	1	0
1	0	0
1	1	0



図 2.32: Zuse Z1

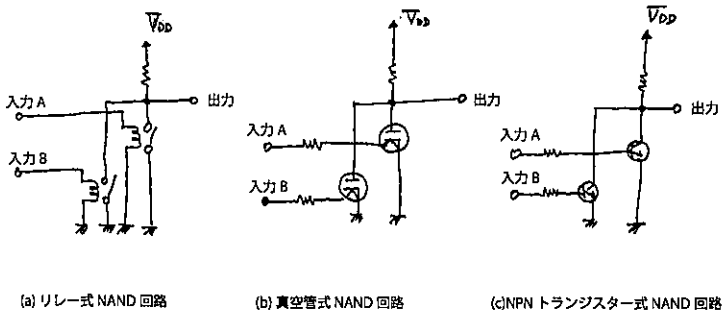


図 2.33: NAND 回路



図 2.34: ENIAC

表 2.2 の演算を実現するデジタル回路例である。同じ第二次世界大戦中にあまり時間差もなく、英国でコロサス (Colossus Mark I 1943 年, Colossus Mark II 1944 年)、米国で ENIAC(1946 年 図 2.34)、アタナソフ&ベリー・コンピュータ (ABC) という真空管式計算機が発明された。

ペンシルベニア大学のジョン・モークリーとジョン・エッカートを始めとした ENIAC の開発チームは ENIAC 稼働も間近という 1944 年に、様々な経験を生かした新たなコンピュータ EDVAC の設計と構築を、数学の天才と言われるジョン・フォン・ノイマンをコンサルタントとして迎えて開始した。EDVAC の論理設計を話し合いまとめた結果を「First Draft of a Report on the EDVAC[5]」というタイトルで発表する。

このレポートで提案されている新しい計算機とはプログラム内蔵方式と呼ばれるものである。元来の計算機、あるいは機械というのはプログラムとその操作対象であるデータは別のものと見なし、別々の方法、場所に格納されていた。たとえば、プログラムはパンチカードで与え、データはダイヤル式入力装置で与えるといった方法である。このドラフトでは、演算装置とその制御装置、およびメモリ、入出力装置からなる機械が提案されている。そして、プログラムはデータと同じようにコード化され、データと一緒にメモリに記憶される。こ

のちよつとした工夫が、このあとのコンピュータの飛躍的な発展につながる。言うまでもなく現代のコンピュータの応用範囲の広がりはこちらから来ているのである。

コンピュータ初期の段階では、新たなプログラムを入出力装置から読み込むプログラム「ローダ」というものが可能となる。プログラムもデータもすべてパンチカードに打ち込み、ロードすることが可能となる。また、プログラムがプログラムを生成する仕組みを作ることが出来る。この代表的な技術がコンパイラ、インタプリタだ。EDVACの時代のプログラムは機械が理解するもので、命令をコードで表し、実行の順番でコードを並べて行くという方法のもの、つまりマシン語と呼ばれるものである。コードをニーモニックと呼ばれる人間に分かりやすい命令の略号のようなもので置き換え、さらに若干の補助的なマクロを備えたアセンブラが開発される。さらにプログラムの書きやすさを人の立場で考え、構造化、抽象化、オブジェクト指向といった概念を導入した高級言語が開発される。高級言語を自動的に再編集し、機械が理解出来るマシン語を生成するシステムがコンパイラである。インタプリタとは、人間の入力する高級言語をそのまま理解し実行するプログラムのことである。しかしインタプリタと言えども、一旦高級言語を中間コードというマシン語に置き換える。このマシン語を解釈する仮想的な機械=ヴァーチャル・マシンによって実行するという形式をとる場合が多く、インタプリタとコンパイラの差を厳密に分けることはなかなか難しい。

プログラムとデータに区別がないということは、ネットワークで遠隔地でプログラムを実行するという事も可能となる。逆に遠隔地にあるプログラムを手元のPCで動作させることも容易にできる。誰かがネットワークで公開したPDFドキュメントを手元にダウンロードして画面に表示するという単純なことを考えてみよう。おそらくPDFドキュメントというのはデータだと思っている人も多いだろう。実はPDFの中身はプログラムなのだ。コンパイラやインタプリタはコンピュータに精通した人にしかなじみがないが、ホームページの表示やデータベース検索、こういったネットワーク・アクティビティの多くはプログラム内蔵方式によるのである。

ヴァーチャル・マシンは、プログラムをデータとして取り込み、解釈して実行するプログラムのことである。例えばWindowsにおいて実行されるプログ

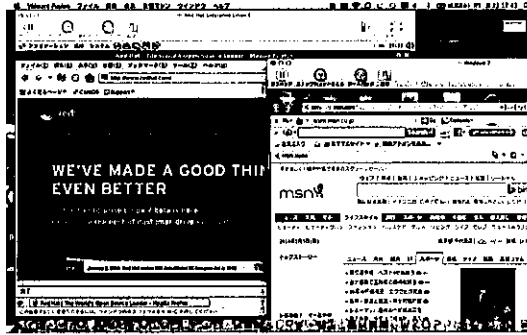


図 2.35: ヴァーチャル・マシン

ラムを解釈して実行するプログラムをマッキントシュの上で動かすことが可能である。図 2.35 はマッキントシュの上で、Windows と Linux を動かしている様子である。これらのコンピュータの機能はプログラム内蔵方式の賜物である。

現在、このようなプログラム内蔵方式のコンピュータのことをノイマン型コンピュータと呼ばれている。それは報告書 [5] がノイマンの名前で発表されたからである。だが、実際このアイデアがノイマンのアイデアであったのかということには疑問の声もあるようだ。ワトソンとクリックのセントラル・ドグマもそうだったように、偉大な名の後ろには寡黙な天才が隠れている。世の中のメルクマールのアイデアというのは、単純かつ明解である。その割には、大きな仕事は一人の手では出来ない。様々な人が智慧を絞り合って成し遂げられるものである。モークリーとエッカートの功績があってしかるべきであろう。

バベッジの解析機関、ツーゼの Z3、あるいは ENIAC はプログラム内蔵方式ではない。しかしアルゴリズムを実行することが出来た。一方、プラハのオルロイ、ブレゲのマリー・アントワネットの懐中時計はプログラムは出来ない。だが、オルゴールにはシリンダーやディスクを取り替えると音楽を変更できるものがある。オートマタも同様のメカニズムで動きを制御できるものがある。あるいは、ジャガード紡績機械にも織を変更するためのパンチカードがあった。これらはプログラムであると言うことはできるが、しかしアルゴリズムを実行できる機械ではない。ではアルゴリズムとはなにか、プログラムとアルゴリズム

ムはどう違うのか、そして最後の問いとして DNA はアルゴリズムか。

2.3.2 オートマトン

様々な計算機、コンピュータが登場するようになって、様々な機械（オルロイも含めて）を数学的にモデル化しようという試みが行われた。その中の一つが有限状態オートマトンと呼ばれるものである。そのほかに代表的なものとして入計算、チューリング・マシンがある。有限状態オートマトンと入計算、チューリング・マシンの中の万能チューリング・マシンが等価であることが、アラン・チューリングとチャーチによって示され、この3つのモデルに等価であるものをアルゴリズムと名付けようという、チューリング=チャーチのテーゼがなされたのである。現在この定義をもってアルゴリズムの定義としている。

では、その手始めにオートマトンという概念を説明しよう。ここに一台の機械があったとする。その機械には初期状態と終了状態がある。そして、何らかの一つの入力に対して反応し、機械は状態を変え、かつ、出力を出す。この機械の構造がどうなっているかということは棚に上げブラックボックスと見なす。そうした場合、着目点としては以下の3点になる。

1. 入力と出力を有限個のコードでコード化出来る。
2. 状態の数は有限個。
3. 同じ状態、同じ入力に対していつも同じ状態へ遷移する。

この3つの条件を満たすとき、この機械を有限状態オートマトン (Finite State Machine) であると言う。図 2.36 は 20 円のあめを 1 個買うことのできる簡単な自動販売機の例である。商品はあめ 1 個しかないとしよう。コインの投入口と返却ボタン、商品受取口と釣銭の口がついたものである。投入できるコインは 10 円玉だけとした場合、10 円玉を 2 個入れるとすぐに商品が出てくる。返却ボタンが有効に働くのは 10 玉を 1 個入れたときである。それ以外は返却されるコインはない。この場合のオートマトンを図 2.37(a) に示した。オートマトンを書くときの基本的な事項として、初期状態を黒丸、あるいは黒丸からでた矢印の先に書く。一方、終了状態を黒の二重丸で書く。(a) においては、初期

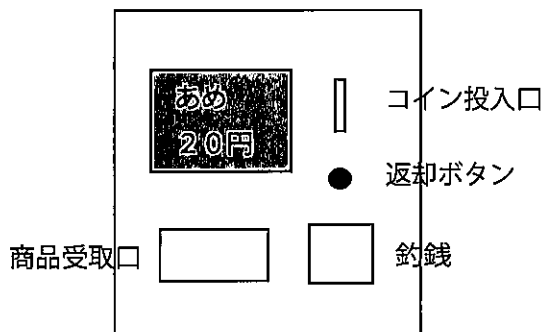


図 2.36: 自動販売機

状態、終了状態の他に、状態は事実上の初期状態である「待機」と「10円玉」（10円玉が入っている状態）がある。「待機」のときに10円玉を入れると、「10円玉」状態へ遷移する。矢印が状態遷移を表す。矢印の下に「10円玉」と書いてあるのは、投入したコインであって、入力にあたる。入力に対して出力があるとき、"/"で区切って出力を書く。「10円玉」状態に10円玉を投入したときに、「10円/あめ」と書いてあるのがそれである。出力として「あめ」が出てくることを示している。なにも出力がない場合は、出力の記述は省略される。おそらく「待機」状態のときでも返却ボタンを押すことは可能である。すべての状態においてすべての入力の可能性があるが、全ての入力に反応するわけではない。「待機」状態における返却ボタンのように、入力が記載されていない場合は、その入力があった場合は、デフォルトの動作をすると約束する。この場合は状態遷移を起こさないというのがデフォルトの動作である。オートマトンによってはエラーとする場合もある。このデフォルトの状態を考えながら、自動販売機の電源を切った場合のことを考えてみよう。10円玉が入っている状態で電源を切ると、お客さんに迷惑をかける。(a)ではそのような場合は電源が切れないように設計されている。

問題を少し複雑にしてみよう。10円玉と5円玉が投入できるようにしたのが、(b)のオートマトンである。コインを入れる順番によっては、25円まで投入可能で、その場合は5円のおつりが出てくる。真横に向かう状態遷移は10円を

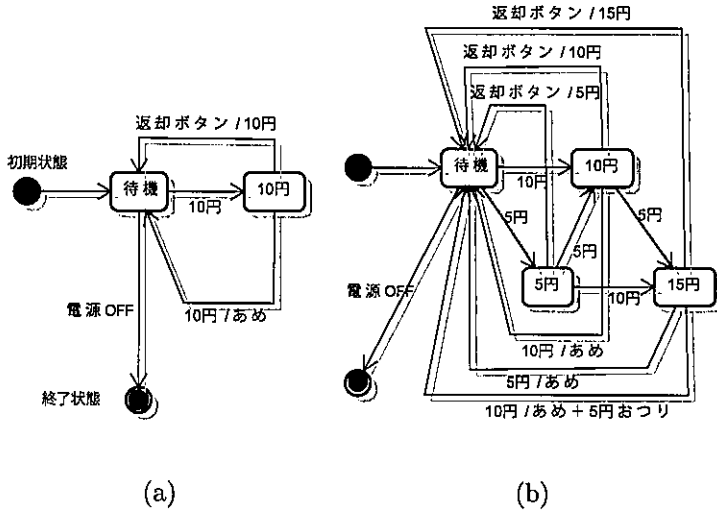


図 2.37: 自動販売機のオートマトン

投入した時、そして、斜め下に向かう状態遷移は5円を投入したときである。状態は5円刻みで「15円」までとなるので、返却ボタンもそれに対応しなければならない。全体としてきれいな菱形になるのが面白い。

図 2.38 は単車線の道に左右からくる車をさばく2つの信号機の例である。右の信号機と左の信号機は独立していて、お互いが通信し合うことによって両方から車が入らないように制御する。図 2.39 はその信号のオートマトンである。初期状態、終了状態は省略されているが「通行可」つまり両方とも青信号のところを初期状態としよう。右と左の信号は同じ仕組みで一見バラバラに動いているように見えるが、お互いが LOCK 信号を送受信してコントロールしている。「通行可」の状態、センサから車を検知する信号を受けると、「通行ロック」状態になる。「S」(=sencer) という入力がある。これと同時に、対向の信号に、「L 送信」LOCK を送信する。解除センサが車が通りすぎたことを関知すると「S 解除」信号が入力される、これと同時に、対向に信号に「L 解除」LOCK 解除信号を送信し、「通行可」の状態へ戻る。もし「通行可」のときに対向の信号から LOCK を受信すると「停止」状態へ遷移する。こちらの遷

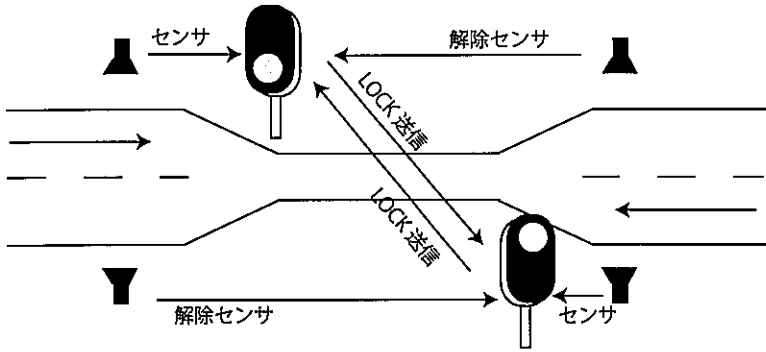


図 2.38: 信号機の例

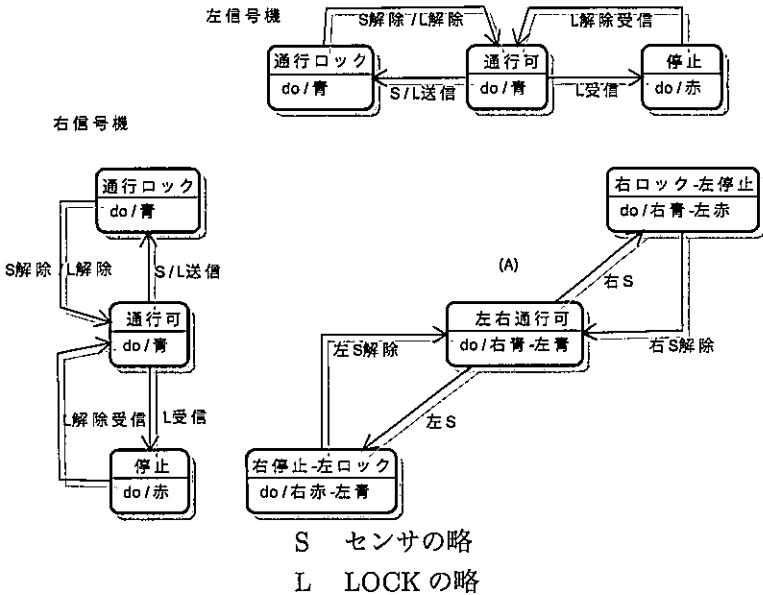


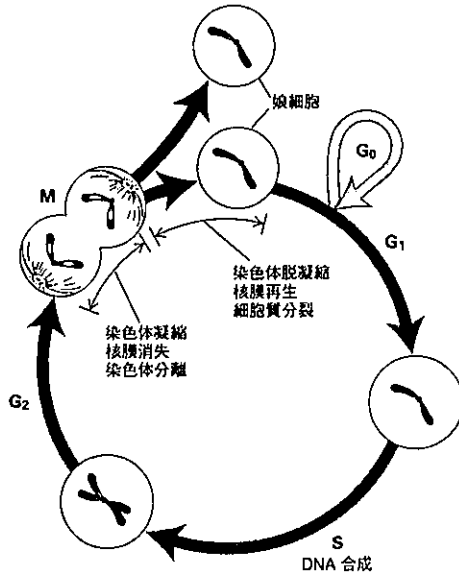
図 2.39: 信号機の例

移は皆さんで解析していただきたい。

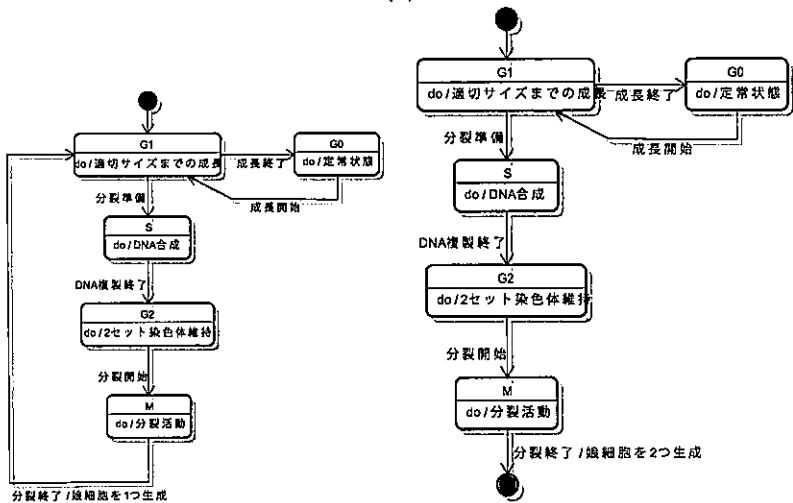
オートマトンの書き方が新しく増えた。状態を表す矩形が上下に分かれ、上段は状態名である。下段に、do / と書いて、この状態の間行うことを書く。ここでは、「青」「赤」というように信号の色を示している。さて、それぞれの信号機は3つの状態を持っているので、お互いがバラバラに動いていたならば、ある瞬間における二つの信号機の状態の組み合わせは合計9通り=3×3ということになる。しかし、LOCK信号をやり取りしながら、お互いが同期しているので、あり得ない(あってはいけない)組み合わせというのがある。単純な話、両方が青のまま車を通してはならないので、両方が「通行ロック」の状態はあってはならない。二つのオートマトンをマージしたオートマトンが図中(A)である。これは、並列に動くシステムの正当性を検証する一つの方法である。

図 2.40 は細胞周期と呼ばれ、細胞の状態遷移を表す図である。図の (a) は [4] にある図であり、これをオートマトンで表したのが (b) である。細胞周期について説明しよう。細胞周期とは細胞の一生であり、一つの細胞が分裂し、DNAを複製し、分裂を行い、娘細胞へ遺伝子を引き継いで行く周期である。細胞が成長し大きくなって行く G1 期、次の分裂に備え DNA を複製する S 期、複製終了後、DNA を染色体として構成する G2 期、分裂を行う M 期に分かれる。これをオートマトンとして見た場合、興味深いのは、(a) ではエッジになっている部分が (b) ではオートマトンの状態を表すノードとなっている。だがこれは書き方の違いである。おそらく (a) では、たとえば S 期においては DNA を複製する作業がずっと続いているということを矢印で表したかったのだろう。一方 (b) の図では、複製作業は、その状態における実行活動として「do / DNA 合成」と書く。だが (a) と (b) は全く同じことを表している。

次に、初期状態と終了状態について考えてみよう。初期状態や終了状態をどの状態にするかは概念的である。自動販売機の例では電源を入れたときを初期状態、切ったときを終了状態としたが、これが、自動販売機の中に電力で動くプロセッサが入っており、このプロセッサが内蔵するプログラムが状態遷移していると考えているためである。自動販売機の筐体をも含めたライフサイクルを考えたとなると、工場で自動販売機が製造されたときが初期状態である。その後は、「通電されていない状態」というのがあり電源がオン/オフされるたびに、「待機状態」と行き来する。最後に、耐用年数を過ぎ廃棄されるときが終了



(a)



(b)

(c)

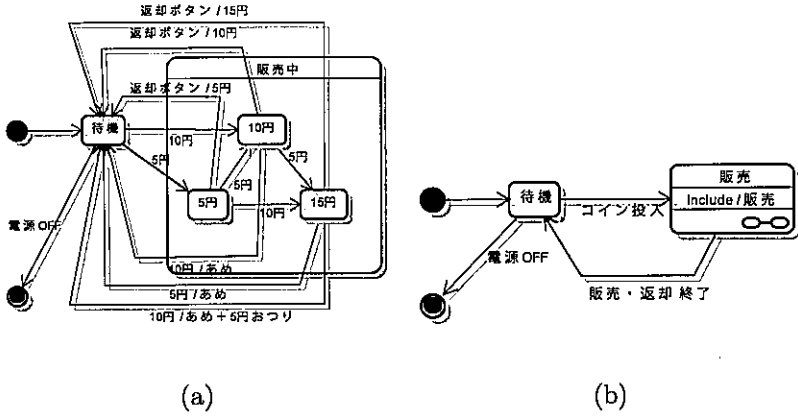
図 2.40: 細胞周期

状態である。

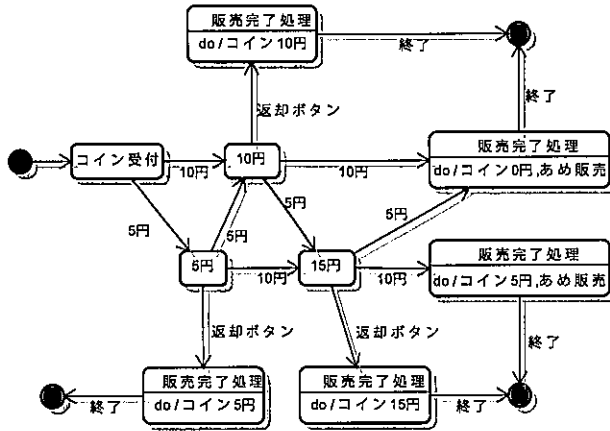
細胞の場合、細胞質が連続的に受け継がれながら状態がいつまでも巡るために初期状態と終了状態をどう考えるかというのは様々である。図 2.40(a) では、分裂して出来る 2 つの娘細胞のうち一方だけが新しいものとする考え方である。もう一方は G1 の段階で親細胞よりいったんは小さくなるが、親細胞そのものであるという考え方である。(c) は別の考え方で、分裂して二つの娘細胞に引きちぎれる瞬間に親細胞の寿命はつきたとする考え方である。二つの娘細胞が平等にこの時点から新しく初期状態となる。どのように考えるかは、オートマトンを書くことによってどのような議論をしたいかによる。例えば、細胞周期以外の細胞のもつ性質が親細胞から娘細胞に受け継がれるかどうかを議論したいとしよう。特に幹細胞と呼ばれる細胞では、親細胞の形質を一方の娘細胞だけが受け継ぎ、もう一方は別の性質となる。脊椎動物などの多細胞生物において、一つの臓器が生成されるとき、臓器を生成する元となるのが幹細胞である。幹細胞は目的の臓器を構成する別の新しい細胞を生成していく非対称分裂を繰り返す。このような場合は (c) ではなく (a) のように考えるのが正しい。発生の最初の過程では幹細胞も少数であり、目的を達成するにはある程度の多数に増殖する必要がある。この場合はおそらく (c) のように対称な分裂をするだろう。

細胞が有限オートマトンか、このレベルであれば有限オートマトンである。初期状態と終了状態を決めることが観念的であるのと同じように、どのように複数の状態として分けるかということは観念的である。信号機の例のときに、二つの信号を二つのオートマトンで分けたり、ひとつのオートマトンにまとめたりすることを示した。この二つの方法は動きとして「等価な機械」を示している。状態をどうとらえるかの違いの例である。また、一つの状態の下半分に do と書いて実行内容を記載した。信号機の場合は「青」「赤」というように一見変化の無い実行内容であるが、たとえば、これが盲人用の音声信号機で、青の音楽、赤の音楽が流れる仕組みだったらどうだろうか。音楽は複数の音をシーケンスで繋ぎ合わせた音になっている。いま音楽のどの部分を放送しているかという状態管理を伴う新たなオートマトンである。これを内部遷移という。

自動販売機の例において、内部にたまっている投入金額ごとに状態を分けた。使えるコインの種類がさらに増えると、投入金額の組み合わせは無数に多くなって行く。オートマトンとして捉えるには煩雑になってくる。そこで、図 2.37(b)



(a) (b)



(c) 「販売」サブ・オートマトン

図 2.41: 自動販売機の状態の抽象化

を図 2.41(a) のように書き換えてみるとどうだろうか。これは、図 2.37(b) においてコインが内部にたまっている状態を「販売中」という状態にくくったのみである。しかし「待機」と「販売中」の間の状態遷移が煩雑なので、サブ・オートマトンという方法を導入したのが、図 2.41(b) である。コインを投入すると「販売」へ遷移し、販売または返却が終わると「待機」へ移行するというシンプルな構造になる。コインの処理というややこしいプロセスは、(c) の「販売」サブ・オートマトンにまとめられる。このように状態というものをどうとらえるかは観念的である。

2.3.3 手続きとフローチャート

オートマトンとデータフローのバリエーションを見て行こう。まずはオートマトンのバリエーションとして手続き、あるいは手続きを図にしたフローチャートと呼ばれるものである。手続きというのはやるべきことを一つ一つ順番に記述したものである。これを図にしたものがフローチャートである。オートマトンは、一つの状態において「それに対する入力受付、そして入力に対して判断を下し、別の状態へ移る」という「手続き」を与えるものと考えれば、オートマトンを手続きに変換することが出来るはずだ。

図 2.37(a) のオートマトンを手続きに直してみよう。この図には2つの状態があるので、「それに対する入力受付、そして入力に対して判断を下し、別の状態へ移る」という手続きが2組出来る。図 2.42 はそれを表したフローチャートである。黒丸から開始し、最初のグレーの矩形が図 2.37(a) における状態「待機」とそこから出る状態遷移の処理に対応する部分で、2つめのグレーの矩形が図 2.37(a) における状態「10 円」に対応する部分である。このようにオートマトンはフローチャートで表すことが出来る。

次に見るのがオートマトンの構造化プログラムへの書き換えである。オートマトンは図として見ていて分かりやすいが、実際にプログラミングをするときに、図を書いても、それをコンピュータは認識することも解釈することもできない。もちろん最近の技術では図で書かれたプログラムを認識し実行する、ビジュアル・プログラミングという技術も存在するが、プログラマにとって効率的、迅速なプログラミングはやはりテキストで書くプログラムだろう。図は二

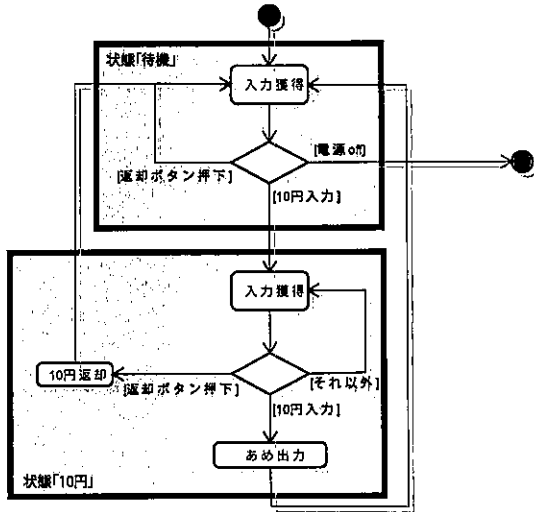


図 2.42: 自動販売機のフローチャート

次元に広がっているのに対し、テキストはシーケンシャルな文字の列である。このようなシーケンシャルな文字の列でもオートマトンと同等の意味をもったプログラムを記述する手法がある。これが構造化プログラミングである。

図 2.42 では菱形の条件分岐のところの出力が 3 本以上ある。これは複数の 2 本の出力からなる条件分岐が組み合わさったものと解釈し書き直すことが出来る。図 2.43 がそれで、条件分岐は条件にマッチした場合と「それ以外」の出力からなる。さらにループに注目してみると、loop (a) ~ (c) の 3 種類があることが分かる。

二本の出力からなる条件分岐は、`if (条件) then { } else { }` という文に置き換えることが出来、ループは、`while { }` 文と、`while` 文から抜け出すための `last;` 文に置き換えることが出来る。この置き換えを施したものを図 2.44 に示す。

図 2.44 のプログラムを起動すると、まず、loop (c) loop (a) の `while` 文の中に制御は入り「入力待受 1」のところで入力をまつ。得られた入力が入った if 文を通る。一つの目でもし入力が電源 off であつたら、その時点でプログラム

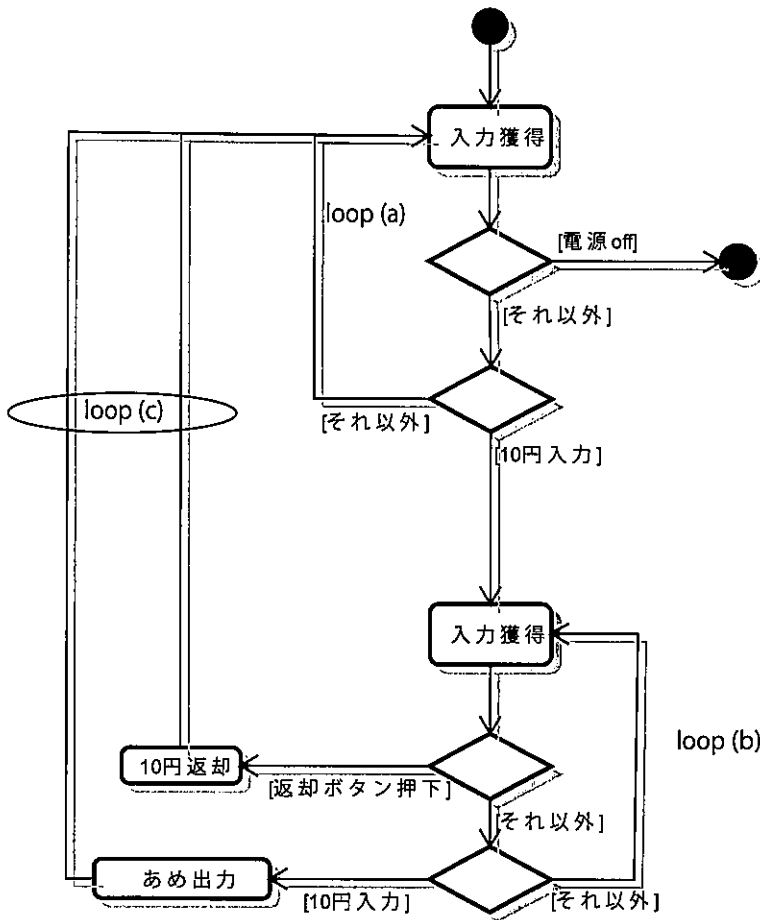


図 2.43: 2 本条件分岐に書き換えたフローチャート

```
while { // loop (c)
  while { // loop (a)
    inp := input; // 入力待受 1
    if ( inp == [電源 off] ) then {
      stop;
    } // else 文の無い if 文
    if ( inp == [10 円入力] ) then {
      last;
    }
  }
  while { // loop (b)
    inp := input; // 入力待受 2
    if ( inp == [返却ボタン押下] ) then {
      10 円返却;
      last;
    }
    else if ( inp == [10 円入力] ) then { // else 文のある if 文
      あめ出力;
      last;
    }
  }
}
```

図 2.44: 構造化されたプログラム

は停まる (stop;) 10 円入力だった場合は、last 文で loop (a) のループを抜け出し、loop (b) の方へ制御が移る。else 文の無い if 文と else 文のある if 文を用意してみた。else 文のないものは、条件が満たされない場合、then の中身は実行されずにそのまま先へ進むという意味である。else 文が空と考えてよい。このようにオートマトンをテキストに置き換えることが出来た。構造化プログラミングはプログラマにとって見やすいと同時にオートマトンと同等の記述力を維持できるという性質を持っている。

2.3.4 データフローと関数

状態遷移というのはある時間におけるトリガーによってある状態から次の状態へ移ることを示しており、その矢印の方向は状態の時間的順序関係を表していることになる。手続きやフローチャートも同様に時間という概念が明確に存在する。では、以下のような式はどうだろうか。

$$3 \times 4 + 5 \times (6 + 3) - 7 \times 2 = 43 \quad (2.1)$$

この数式はオートマトンに比べると一見時間というものが見えない。しかし計算するということは行為であるが故に時間が係わらざるを得ないはずである。実際上記の式は次のように計算される。

(a) $3 \times 4 = 12$

(b) $6 + 3 = 9$

(c) $5 \times (b) = 5 \times 9 = 45$

(d) $7 \times 2 = 14$

(e) $(a) + (c) - (d) = 43$

このように時間的順番が暗示的に内在している。しかしこの計算の順序は曖昧である。たとえば、

(a) $6 + 3 = 9$

$$(b) 3 \times 4 = 12$$

$$(c) 5 \times (a) = 5 \times 9 = 45$$

$$(d) 7 \times 2 = 14$$

$$(e) (a) + (c) - (d) = 43$$

というように順番を変えても結果は同じである。このように計算には順序関係に時間的曖昧性があるのである。これをもう少し厳密に見るために図 2.45 を描いた。この図においては矢印でたどれる関係が時間的順序関係があるということである。たとえば、(a),(e),(f) の演算はこの順番で行う必要がある。一方で、(a),(b),(c) はどれを先にやっても良い。場合によっては同時に行っても良い。

どのような二つの要素の間でも、かならずどちらかが先で一方が後と決まる関係を順序関係、そして、上述のような中には後先が決まらない2要素もあり得る曖昧な順序を半順序関係と呼ぶ。計算はたいてい半順序関係であり、このような半順序関係の計算をグラフで書いたものをデータフロー・グラフとよぶ(図 2.45)。図 2.45 のようにデータフロー図は、長方形で囲まれた初期値、丸い図形に演算が記されたテンプレート、それらを結ぶ矢印で出来ている。矢印に添って、データが流れる。テンプレートは入力される矢印からのデータがすべてそろったところで、演算が開始され、結果を送り出す仕組みである。

データフローにおいて条件式も表現できる。たとえば、以下のような式を考えてみよう。

$$(a) = 2 \times x$$

$$(b) = x^2$$

$$(c) = x^3$$

$$(d) = (a) > 4 \text{ ならば、} (b) (a) \leq 4 \text{ ならば、} (c)$$

$$(e) = (d) / 4$$

(d) の値を計算するとき条件が与えられている。これをデータフロー図にするならば図 2.46 となる。新しいテンプレート「条件」が加わっている。これ

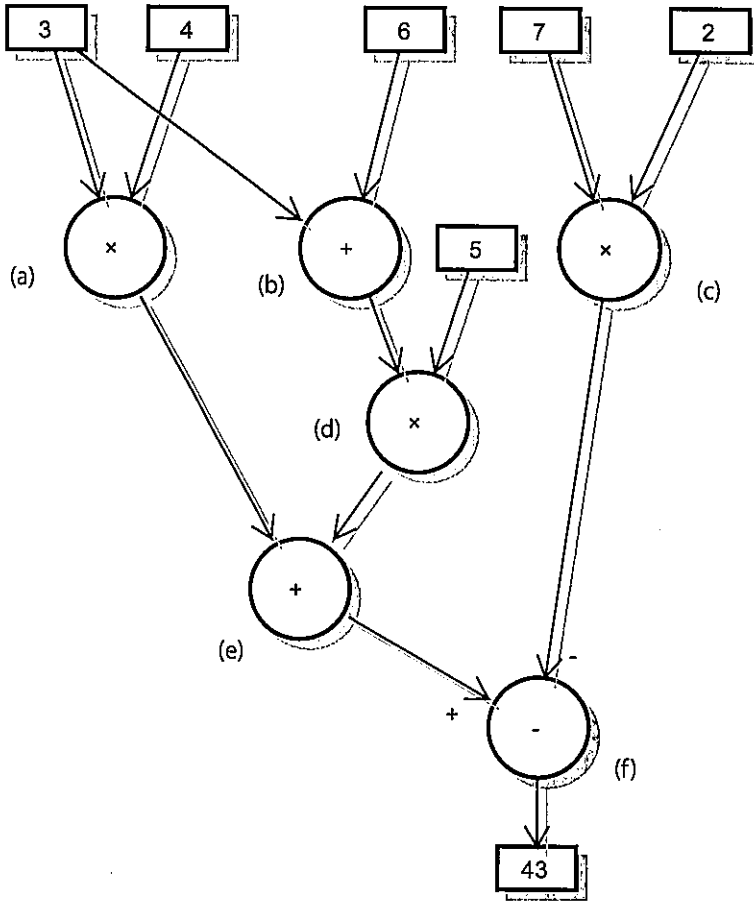


図 2.45: 式 (2.1) のデータフロー

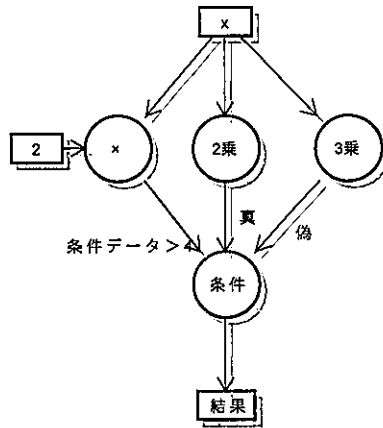


図 2.46: 条件分岐のあるデータフロー

は、条件データが4より大きい場合、「真」の入力に入ってきている値を返す。4より小さい場合は「偽」に入ってきているデータを返すものである。

この図で、入力は x という変数になっている。ここに値を入れると「結果」が出力される。 x が同じ値であれば「結果」も同じ値である。つまり、 x を次々と変えて入力しても、過去の結果が蓄積されることはない。これに対して図 2.47 はどうだろうか。最初の一つ目の x の入力は、右から入ってくる0と加算される。結果は「分岐」テンプレートによって2つにコピーされる。一方は出力結果となるが、もう一方は、加算テンプレートの右の入力 (a) に戻ってくる。 x の次の値と加算されるのを待つことになる。こうして、得られる「結果」は x として入力される値の総和となる。このようなデータフローのループによって「状態」を作り出すことが出来る。

このようにデータフローで状態を作り出すことが出来るならば、例えば、図 2.37 のようなオートマトンもデータフローに置き換えることが出来る。よく考えてみれば、コインを入れるとコインが内部に溜まり、ある条件が見たされるとあめが出てくる、という自動販売機の構造はデータフローそのものである。図 2.48 は、自動販売機のデータフロー図である。約束として「返却」入力は、返却ボタンが押されると、「yes」という値を入力したものと同一と約束しよう。

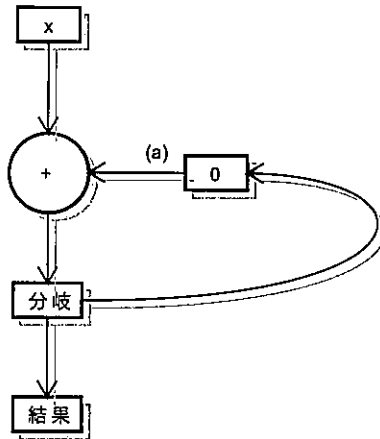


図 2.47: ループのあるデータフロー

「コイン」入力はコイン玉の値である。全体の出力としては、「あめ」と「おつり」がある。新しいテンプレートについて説明しよう。「ゲート」は入力があると、名前のついている出力にはその名前の値を出力。名前のない方の出力には入力と同じ値を出力するものとする。(b)の「ゲート」は入力があると、右の方に常に「0円」を出力し、下の方へ入力された値、つまり「yes」を出力する。(c)の「ゲート」は、左の方に常に「no」を出力し、下へコインを送る。「マージ」というテンプレートは、二つある入力のうちどちらか一方がそろった場合にその値を出力するものとする。「条件分岐」は「条件」入力に「yes」または「no」という値を受け、データ入力を、「yes」「no」のそれぞれの出力へ振り分ける機能を持っている。

このように見ると (a) のところに、入力されたコインが貯まる。コインが入力されるたびに (d) のテンプレートで 20 円より大きくなったかをチェックし、20 円をオーバーすると (e) のテンプレートによって商品のあめ、(f) によっておつりが出力される。返却ボタンが押されたときには、(a) にたまっていた値がフラッシュされ、(g) の条件分岐で、直接「おつり」へ送られる。(a) はフラッシュされてしまうので、次のコインの入力に対応するために (b) のゲートは (a) にも 0 円を送る。あめは (e) のテンプレートをループしているが、これは、商

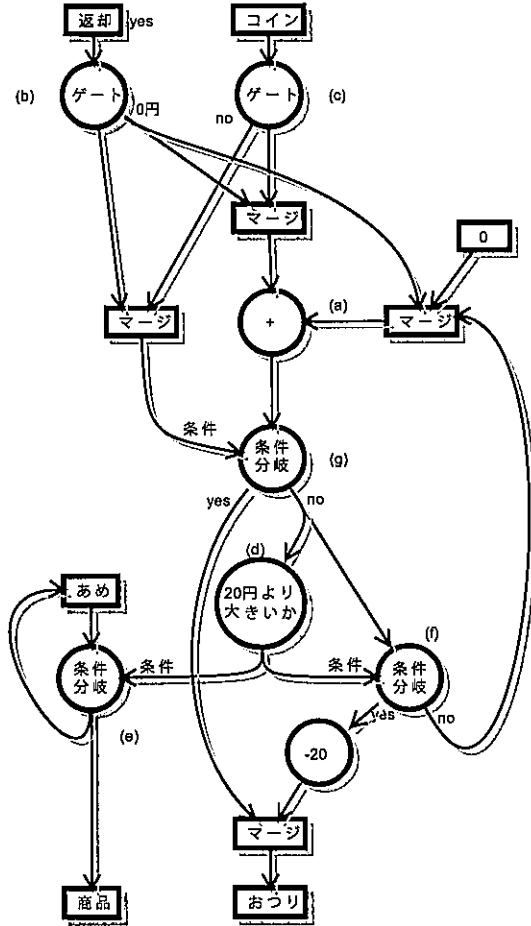


図 2.48: 自動販売機 of データフロー

品が無くならないためである。もし、商品の数に限度があり、いずれあめは無くなるとするならば、商品の数だけ初期値のあめをかさねればよい。どうやら、オートマトンはデータフローに置き換えることが出来そうである。

データフローは関数と密接な関係を持っている。今一度式 (2.1) と図 2.45 に戻ってみよう。各演算は関数であるということは、さほど理解に苦しむものではない。式 (2.1) を、

$$\text{sub}(\text{add}(\text{mul}(3, 4), \text{mul}(5, \text{add}(6, 3))), \text{mul}(7, 2)) = 43 \quad (2.2)$$

と書き直すことも出来る。各関数、 add , sub , mul は足し算、引き算、かけ算であり、図 2.45 の各テンプレートと対応している。今度は、図 2.47 のようにループがある場合を考えてみよう。 x には値が次から次へと入ってくるのが想定されている。その順番を添字として、 x_i と書いてみよう。同様に出力も y_i と置くことにしよう。もう一つ図 (a) の所に最初は 0 が入っている。 x の入力があるたびにここに値が次から次へとやってくる。従って、ここの値も a_i というように数列に置き換えることが出来る。ただし、 $a_0 = 0$ である。すると、このデータフローは、

$$\begin{aligned} y_i &= x_i + a_i \\ a_{i+1} &= y_i \end{aligned} \quad (2.3)$$

と変換することが出来る¹³。ここでは、数から数を計算する式、ではなく、数列から数列を計算する式と捉えることが出来る。このような数列によって入力と出力が時間遷移することが表現される。むろん、式 2.2 も入力される数を取り替えて行くとそれにつれて出力される数も 43 とは異なる数になる。これを並べれば数列を得ることが可能である。しかし図 2.47 および式 (2.3) は、 a_i が右辺にも左辺にも現れ、さらに、添字が一つずれることによって漸化式になっていることである。漸化式になることで、前の添字の入力列に影響された出力が得られることになる¹⁴。

¹³関数的な書き方を徹底するならば、 $\text{add}(x_i, a_i)$ というような書き方をすべきかもしれないが、演算というものが関数であるということが分かっているならば、ここまでする必要もないだろう。

¹⁴この漸化式は、以下のように書き換えることが出来る

表 2.3: オートマトン、データフロー、関数

オートマトン	データフロー	関数
単一状態	ループなし	数式
複数状態	ループあり	数列と漸化式

この状況を整理すると表 2.3 のようになる。オートマトンでは状態遷移というかたちで時間の変遷が表現されていた。それがデータフローになるとフローのループという形になる。関数では数式上は一見時間の存在が見えないが、数列というかたちで時間が表現されている。

2.3.5 マークアップ言語

データフローを言語化したものが関数型言語と言って良いだろう。関数型言語の代表例として良く言われるのが LISP や Haskell と言った言語である。しかしこれらを知っている人は少ない。だが、HTML(Hyper Text Markup Language)と言ったらどうだろうか。これも知っていると諸手を上げて答えられる人はプログラマと呼ばれる人に限られてくるかもしれない。しかし、HTML を知らぬ間に使っている人は沢山いるはずだ。

$$\begin{aligned} y_i &= x_i + a_i \\ \Delta a_i &= y_i - a_i \end{aligned} \quad (2.4)$$

さらに、 y_i を打ち消し、

$$\Delta a_i = x_i \quad (2.5)$$

$a_0 = 0$ の初期条件を加味し、

$$a_i = \sum_{j=0}^{i-1} x_j \quad (2.6)$$

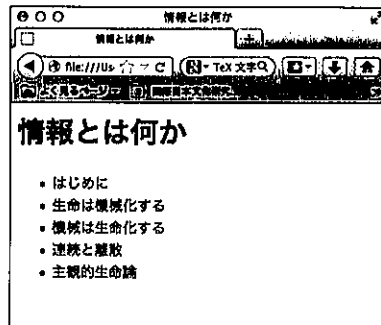
となる。式 (2.4) は差分方程式と呼ばれるものである。漸化式はこのような差分方程式へと変換できる。

```

<HTML>
<HEAD>
<TITLE>情報とは何か</TITLE>
</HEAD>
<BODY>
<H1>情報とは何か</H1>
<UL>
<LI>はじめに</LI>
<LI>生命は機械化する</LI>
<LI>機械は生命化する</LI>
<LI>連続と離散</LI>
<LI>主観的生命論</LI>
</UL>
</BODY>
</HTML>

```

(a) HTML



(b) ブラウザ

図 2.49: HTML と表示された HTML

PC やスマートフォンを利用したことがあれば、ブラウザを開いたことがあるだろう。ガラケーであっても、iMode も一種のブラウザである。ブラウザに表示されるページはほぼ全て、HTML という言語で記述されている。

図 2.49 の (a) は記述された HTML であり、これがネットワークを通しブラウザに読み込まれると、(b) のような画面が生成される。(a) は一見関数のようには見えない。文章が $i \dots j$ と $j \dots i$ という開始タグ、終了タグと呼ばれる特殊な文字列で囲まれたマークアップ言語というものである。開始タグと終了タグの \dots の部分は同じ文字列でなければならない。 $jH1i$ という開始タグには $iH1j$ という終了タグが対応する。この中の文字列を合わせてひとまとまりである。このひとまとまりを要素と呼ぶ。また要素の中に入れ子に要素を含めることが出来る。しかし必ず入れ子構造になっている必要がある。 $iULi \dots jLIi \dots iULi \dots jLIi$ のように交差しているような書き方は禁物である。ここで、要素一つをタグを関数名とする関数と解釈できる。LI という関数は、文字列を受け取り、アイテム化された一行の画像を出力する関数である。UL は、複数の LI の出力を受け取り、全体をアイテム化されたリストへ構成された画像を出力する。H1 は文字列を大きく表示したタイトルの画像へ変換する。このようにして最終的にブラウザに表示すべき全体の画像が完成する。HTML をあえて関数のように記

述すると、

```
HTML(
  HEAD(
    TITLE("情報とは何か"))
  BODY(
    H1("情報とは何か"),
    UL(LI("はじめに"),
      LI("生命は機械化する"),
      LI("機械は生命化する"),
      LI("連続と離散"),
      LI("主観的生命論"))))
```

となる。とても見にくくなったが関数として記述できることが分かる。

このようにタグで記述された言語をマークアップ言語と呼ぶ。一見文章の中にタグが埋め込まれているように見えるが、実際は関数の中の引数として文字列が渡されているという解釈が成り立つ。HTML ではタグには論理的な意味が与えられていて、これを大きく変えることは出来ない。たとえば、TITLE であれば、ブラウザのタイトル領域に表示する文字列を与えるという意味である。しかし多量の WWW のページをインデックス化するような、データベースでは、TITLE を「このページの表題情報としてデータベースへ登録する」という意味に解釈するかもしれない。このようにマークアップ言語の大きな特徴として関数=タグの意味を柔軟に解釈し直すことが出来る。

関数型言語において、関数の意味は後から定義可能である。たとえば、 $f(x)$ と書いた場合、 x は変数であることは自明であるが f も変数として扱うことが出来る。図 2.50(a) はすでにおなじみの固定的な関数に変数 x を投入するデータフローである。一方 (b) は関数も変数化し、データとしての関数と x を同時に投入すると、関数に x をアプライした結果を返す、という「関数」apply である。

HTML では関数定義は、HTML を読み込むアプリケーションにゆだねられている。このため HTML 言語としては不完全かもしれない。しかしこの曖昧性こそが、インターネットでの驚異的な普及の原動力となっている。HTML を見ると、フローチャートや構造化プログラミングのような直接的な手続きを記述したものだけがプログラミング言語ではないということがよくわかる。プログラマの多くは、与えられたプログラミング言語を操っているだけである。だが

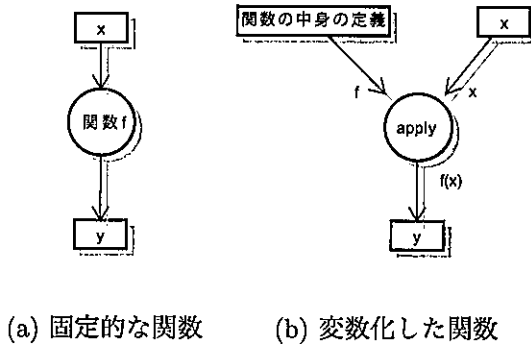


図 2.50: 関数の apply テンプレート

プログラム開発の真骨頂はプログラミング言語を作ることである。開発しようとしているアプリケーションの仕様書が与えられたときに、まず考えることは、この仕様を満たすためにはどのようなプログラミング言語が適しているかである。このように言うと、多くのプログラマは C++ が良いのだとか、java が良いのだ、あるいは LISP である、といった議論を始める。そうではない。アプリケーションで扱うデータの構造が表現しやすい言語、あるいは実行スピードや記述のコンパクトさを考慮したもの、といった条件を重ねて行くと、大抵の場合すべての条件を兼ね備えた汎用言語は存在しない。従ってアプリケーションごとに言語を設計するのが正しい解だ。設計された言語を実行するインタプリタやコンパイラを作成し、そこからアプリケーション開発となる。

かつて、マシン語やアセンブラといった言語しか無かった時代、カーニハン、リッチは新しいオペレーティング・システムを開発するために必要な言語をまず模索した。A 言語、B 言語と試作を重ね C 言語でおおよそ目的が達成できると判断した。UNIX は C 言語で開発された。HTML も、インターネット上でハイパー・テキストを交換するために適した言語としてティム・バーナーズ・リーが開発した。少なからず優れたシステムを構成するときにはそこには優れたコンピュータ言語が介在する。

優れたコンピュータ言語は必ずしも豊富な機能、アルゴリズム記述のために完全であることが必須ではない。時として、不必要な機能や性質を思い切って

削ぎ落したときに利便性が向上し、強力な制約を課すことによって、自然に不要なバグや欠陥をプログラミングから排除することが出来るようになる。しかもそれはアプリケーション・オリエンテッドなのである。

HTMLはその点で優れていると言える。関数定義可能な完全な関数型言語についての解説はもう少しあとにしよう。もう少しデータフローにまつわるバリエーションを見て行くことにする。

2.3.6 シグナル・フロー

データフローによく似たものを世の中で探してみる。工場や科学プラントの生産ライン、あるいは事務所の書類の決済、あるいは物流や交通網なども含めて、物事が流れて行くことによって処理が進んで行く事象やシステムなど類似のものはいろいろある。厳密な意味でのデータフローは、各演算プロセスにあたるテンプレートの中身は状態を持たない。過去に到着した事物の如何によってテンプレートの状態が変化して行くモデルは、後に述べるベトリネットやアクター・モデル、あるいはオブジェクト指向といった概念で議論する必要がある。

オートマトンの特徴が状態とその遷移、また、手順や手続きといった操作の主体がどう動くかであるのに対して、データフローの基本的な特徴は、データという操作の客体と、テンプレートと呼ばれる手順や手続きを単純化した演算プロセスである。これを工場に例えるならば、フォード社のアッセンブリー・ラインは一種のデータフローである。アッセンブリー・ラインに頼らない従来型の工場生産では、一人または一グループの作業員が一つの製品に全ての責任を持つというものであった。この方法では同じ製品を量産するために作業員を多く導入したとすると、作業員間のスキルのばらつきのために、製品の品質にばらつきが生じる。アッセンブリー・ラインは、製品を完成させるための一連の作業を複数の単純作業に分解することにより、作業員のスキルのばらつきの影響を受けないようにする。また、作業が単純化するために、作業員の技能レベルを低く設定しても、品質の維持が可能で、かつ、大量雇用、低賃金化が可能となる。これは、データフローのテンプレートからの状態の排除、単純化の考え方に似ている。

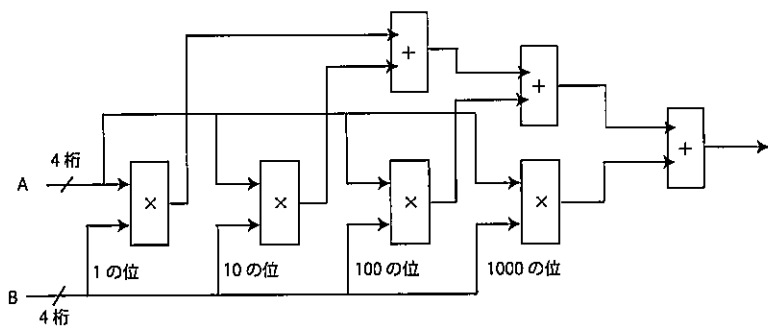
アッセンブリー・ライン的な情報処理として、CPUの演算回路でよく取り入

れられるパイプライン処理について見てみよう。たとえば、2つの4桁の数値 A と B を入力すると、 $A \times B$ を出力するかけ算回路を考えよう。1桁 \times 4桁にかかる時間が 10ns ¹⁵、二つの数値の加算にも 10ns かかるとする。この演算は、 $A \times B$ の一桁分を4回行い、結果を足し込むのは3回であり、うち2回の加算とかけ算は同時に実行可能である(図 2.51(a))。単純計算で実行時間は合計 $5 \times 10 = 50\text{ns}$ である。

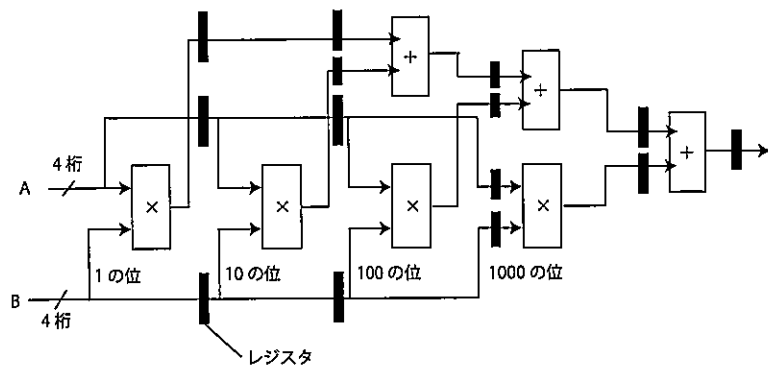
50ns は、回路の入力から出力までが安定する時間でもあるので、この間入力を新しいデータに取り替えることが出来ない。ここにレジスタという記憶装置を導入する。レジスタは 10ns ごとに入力からデータを取り込み、記憶し、常に記憶したデータを出力する。データを取り込む時点と時点の間では、入力データが変化したとしても、それを出力に伝えることはしないのが特徴である。このようなレジスタを、図 2.51(b) のように、各段に挟み込むと、前の計算結果を 10ns ごとに次へ送り込む役割を果たす。すると、最初の A, B は 50ns 間保持しておく必要はなくなり、 10ns ごとに次の計算対象を入力することが出来るようになる。1セットのかけ算を行うには依然 50ns かかるが、 10ns ごとにデータを送り込むことが出来るので、大量にかけ算をこなさなければならないときに計算速度が5倍になる。これは全体をかけ算という一つの演算回路でこなすより、一つ一つをより単純なプロセスに分解した方が効率がアップする例である。

次は、AM ラジオの例を見てみよう。これはアナログの問題の例である。図 2.52(a) は代表的な AM ラジオの回路図である。回路図をそのままデータフローと比較しようとも、それは難しい。実際回路図の動きを一般解として求めようとするならば、各素子の性質を微分方程式で表し、多元連立微分方程式を解くことになる。SPICE という高性能のシミュレータも存在し、なかには回路が発生する電磁波ノイズまでもシミュレーションしてくれるものまである。だが、回路設計者が回路設計を始めるとき、こういったプリミティブな電気信号の挙動から考えることはない。まず最初にもっと概念的な信号のフローを考えるはずだ。AM ラジオであれば電波を受けそこから、チューナによって指示された目的の局を抽出し、搬送波から音信号をとりだす、ということがしたい。するとそのための信号のフローが描けるはずで、それが図 2.52(b) である。こう描くとデータフローになんとなく近づいてくる。

¹⁵ $10 \text{ ナノ秒} = 10/10^9 \text{ 秒}$

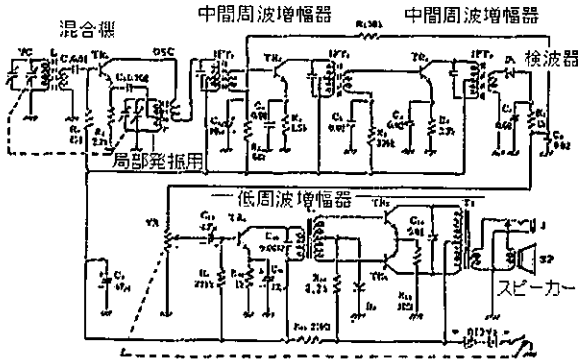


(a) かけ算回路

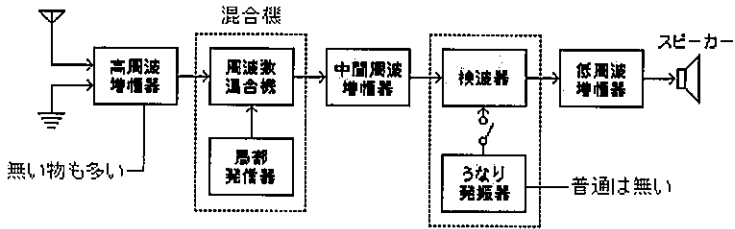


(b) パイプライン化したかけ算回路

図 2.51: かけ算回路



(a)



(b)

図 2.52: AM ラジオの回路 super-heterodyne 方式

http:

//www.icom.co.jp/beacon/backnumber/technical/radio/011.html

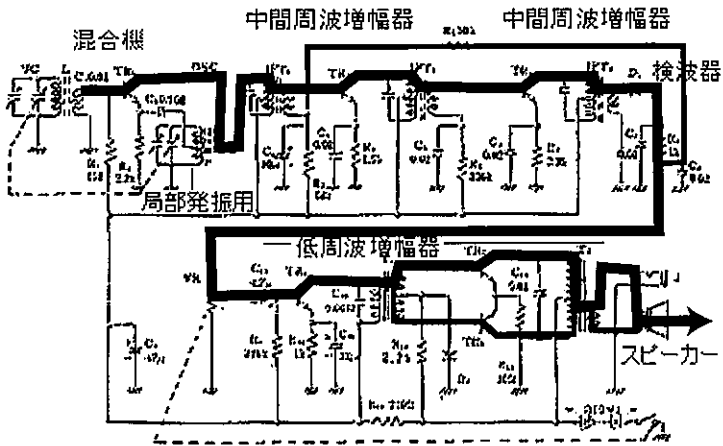


図 2.53: 信号の径路

http:

//www.icom.co.jp/beacon/backnumber/technical/radio/011.html

(b) の場合、アンテナから入ってきたアナログ信号は一つ一つのテンプレートの処理を受けながら左から右へと流れて行く。決して右側の信号の変化が左側の信号の変化に影響しない。つねに矢印の方向へ向かって行く訳だ。信号がデジタルではなくアナログであるということを除いてはデータフローと同じである。一方、(a) のアナログ電子回路図は、素子と銅線のネットワークをそのまま図にしたものである。一見データフローのように見えるが、回路のなかでおこっていることは電流と電圧の相互作用であり、素子の両側でおこる変化は、どちらかが一方へ影響を及ぼすのではなく相互に影響を与える。回路のデータとして捉えられるべきはこの電圧と電流であり、データフローはその相互作用を表すものでなければならない。従って、アナログ電子回路図はその相互影響を描いたものではないために「アナログ電子回路図はデータフロー図ではない」ということになる。だが、アナログ回路自身がデータフローでないことを意味しない。だがこのデータフローはアナログ回路図とは似ても似つかないものになるはずだ。

SPICE シミュレータは電子回路全体を公平にシミュレーションするかもしれない。しかし、アナログ電子回路全体を一様にしか見ることのできない電子回路設計者は未熟であり、ことの本質を見落としてしまう。電子回路は、本来実現したい (b) を実現するのに直接的な部分と、それをサポートする部分にわかれる。図 2.53 は、図 2.52(b) に相当する信号の通り道を (a) の中に太く書き入れたものである。もっとも、アナログ回路では信号を 1 本のラインとは限らない。最も太いラインが信号本体とした。それに共役に働く、フィードバックやプッシュプル信号が中間の太さのラインだ。これ以外のラインは、信号を処理する素子たちに電力を共有するラインであったり、ノイズを除去する回路であったり、信号伝達をサポートするものである。この太いライン上の、(b) における一つ一つのモジュールのつなぎ目の部分にあたるポイントでは、右から左へ信号は流れ、「逆流」するようなことはないよう設計されている。この太線の部分においては、電圧や電流を信号と見なすと (アナロジー)、アナログ回路のもつ本来のデータフローと、回路上に図示される太線が一致するようになっている。このようにしてみれば、全体としてはデータフローとはならないアナログ電子回路も、回路設計者はその一部にデータフローが内包されるように設計していることがわかる。

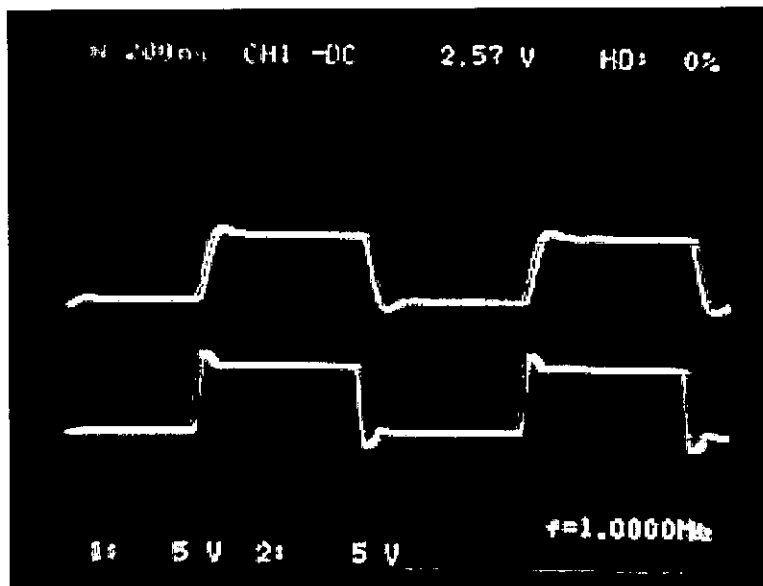


図 2.54: デジタル回路の信号

<http://easyaudiokit.hobby-web.net/bekkan/DISOUSB/disousb.html>

図 2.51 のようなデジタル電子回路もアナログ電子回路で出来ていることを忘れてはならない。図 2.54 はデジタル回路のある部分をオシロスコープを使って観察したところである。デジタル信号は回路の中では 0 と 1 の二値を取っている訳ではない。デジタル回路の代表例である TTL¹⁶回路においては、おおよそ 0.6V 以下を論理的 0、2.5V 以上を論理的 1 とみなしている。論理的 0 から 1 へ変化する時、あるいはその逆の時、電圧上昇・下降には少々時間がかかっているのがわかるだろう。つまり途中で論理的 0 でも 1 でもない時間帯がある。しかしデジタル回路が論理的に正しく動くのは、たとえば、図 2.51(b) のように間にレジスタを設け、過渡期の影響を注意深く排除する設計がなされているからだ。このようなレジスタのことを別名ラッチと呼ぶ。過渡現象で動く入力データを動かさないように掛けがねを掛けるという意味である。

マックスウェルの電磁気学に従って動くアナログ電子回路の完全な微分方程式の世界、そして、そのなかから論理的に抽出されるアナログのデータフロー、アナログのデータフローに論理的 0,1 の解釈を載せたデジタル回路、こういった抽象化の階層構造がコンピュータなのである。

2.3.7 アナログ現象とデータフロー

アナログ回路では、なぜ、太線のところがデータフローと見なせるようになるのか、もう少し詳しく見て行こう。アナログ回路を構成する素子には受動素子と能動素子という大きく分けて二つの素子がある。能動素子とは、トランジスタや真空管などの電力利得のある素子である。電力利得とは入力する電力信号の振幅を出力側で拡大できるものである。こうして図 2.53 をみると、太線の上にはトランジスタが並んでいるのが分かる。トランジスタは入力信号を増幅して出力側に伝えることがあるが、出力側の信号を入力側に伝えることはない。このように信号の一方方向性、つよい非可逆性を示す素子が太線上にある。もし回路がすべて可逆性の素子のみで構成されているとすると、信号は池の中を寄せて返す波のように、回路の中を行ったり来たりすることになり、おおよそ人間が求める機能を実現できるものではなくなる。

¹⁶Transistor-Transistor-Logic

理想的なコンデンサーやコイルは完全な可逆素子である。一方、受動素子であっても抵抗は、電力を熱に変える素子なので非可逆素子であるが「熱」は回路上記載しない。抵抗は熱を利用するものではなく、寄せて返す波を減衰させる役割を担っている、準可逆素子なのである。「熱」も回路上記載し信号として取り扱うならば可逆素子としてデータフローを作り出す素子と言えるかもしれない。事実、能動素子である真空管のカソード部分にはヒータ（抵抗素子）が入っており熱伝達が行われている。

一方、受動素子ではあるがスピーカは電力を音に変える素子である。スピーカは逆に音を電力に変えることも出来るので、厳密には抵抗を含んだ可逆素子である。しかし実際には変換時に大部分のエネルギーは熱になってしまうため、事実上の非可逆素子である。図 2.53 では太線の一番最後の部分に配置されている。音ではないが、電力と圧力を可逆的に相互変換する素子はピエゾ素子と呼ばれ、代表例としては水晶発振子がある。

このように能動素子を代表とする非可逆的に電力信号を伝える素子があるからアナログ回路の一部がデータフローとして認識できるようになる。さらにデジタル回路は能動素子がなければ構成不可能である。図 2.33 を思い出していただきたい。NAND 回路というデジタル回路の三種類の構成法を示したが、リレー、真空管、トランジスタと、そのいずれにも能動素子が使われている。特に (c) のトランジスタを使った回路では、片方の入力に着目すると、似たような回路が、図 2.52(a) にも登場する。もともと、デジタルの NOT 回路は、入力をマイナスの方向へ数十倍から百数十倍へ増幅する増幅回路である。

供給される電力が 0V から 5V であるとしよう。そして、2.5V を中心として -100 倍に増幅される回路であったとする。入力電圧を E とすると出力は次のような式で与えられる。

$$OUT = -100(E - 2.5) + 2.5 \quad (2.7)$$

入力電圧が 5V であるとする、-247.5V まで増幅される。逆に入力電圧が 0V であったとすると 252.5V まで増幅される。しかし実際にはそこまでの電力が供給されている訳ではないために、出力側は 0V と 5V で頭打ちとなる。5V の入力に対しては 0V、0V の入力に対しては 5V の出力となる。逆に出力を中間

的な 3V にするためには、入力に 2.495V を維持しなければならない。しかも、0.025V の変化ですぐに出力は 5V, 0V へ振れてしまう。入力を $2.495V \pm 0.025V$ に保つのは至難の業であり、不可能と言ってもいい。結果的に「2値しかとらない回路」が出来上がるのである。逆の発想をすれば、TTL の NOT 回路の出力を、抵抗を介して入力側にフィードバックさせることによって、大きい利得を小さく調節することが出来る。昔、このようにして簡易アナログ・オーディオ・アンプを作ったりすることもあった。このようにしてデジタル「桁」へアナロジーするアナログ回路としてのデジタル回路の様子が見えてくる。

こうして生まれるデジタル回路には、まだ時間方向のアナログ要素がある。図 2.51 では、入力信号からかけ算の結果が出力されるまでの時間は、回路の特性のばらつきによってかわる。回路全体の性能のばらつきのほかに、計算出力の一つ一つの桁の値が定まる時間も桁ごとにまちまちである。特に後者の問題は切実である。デジタル回路では通常一つの値を転送するために複数の桁を束ねたパラレル・バスをつかう。その桁がバラバラに送信されるとなると、どの時点の桁のセットが正しい値を示しているのかが分からなくなってしまう。この問題を回避するためには、時点時点で桁をそろえる必要がある。レジスタはバスの出発点、終着点には必ずと言ってよいほど存在する。このようにレジスタは時間を「量子化」する装置である。レジスタに対して、ラッチする時間を定期的に指示するクロックが送られる。コンピュータの性能を表す時に 2.5GHz Intel Core i7 といったように周波数を言う。この周波数は、CPU を構成するレジスタに供給されるこのクロックの周波数である。

デジタル回路の場合、一番安定化に時間がかかる回路に合わせてクロックを回路全体で同一化するのが望ましい。すると、デジタル回路は、状態を持たず、入力を演算し出力へ伝える回路とレジスタで構成されたものと見なすことが出来る。これを同期回路と言う。アナログ回路では、太線を通り抜ける信号は、時間の全ての瞬間が意味を持っていた。しかし、同期回路では、クロックとクロックの間は、過渡的な状況として「無意味」な信号、そして、クロックが入った瞬間を「意味」のある信号として区別する。このような差は、回路を眺める設計者の概念的な差であるとも言える。しかし、回路のある点の過渡状況が別の場所へ波及することはない。波及を阻止するのがレジスタの役割である。この意味で回路自体が本質的にこのような概念の差を受け入れているとも言える。

クロックを統一化するのは高速なデジタル回路では大変なことである。2.5GHzのクロックでは、1回の演算は、0.4ns以下で行わなければならないことになる。回路内のあらゆるレジスタへのクロックの到達時間の誤差はこのさらに数十分の1の時間差に納めなければならない。たとえば0.01nsの誤差に納める必要があったとしよう。信号の伝達スピードは光の速さを超えることができない。1cm四方のLSIチップでは、1辺を信号が伝わるには0.3nsくらいかかるのである。LSIの中身は伝達距離が同一になるような工夫等がされており、時間誤差を実用レベルまでに押さえている。だが、LSIとLSI、あるいはコンピュータとコンピュータとなると、GHzレベルではクロックで同期させることは徐々に不可能となる。転送距離が長くなればなるほど、複数の桁を横に並べることをあきらめ、1本のラインに時間差でビットを運べる非同期型のシリアル通信が主流となる¹⁷。

さて、目を細胞に転じよう。細胞の中でおこっていることは化学反応である。化学平衡、拡散方程式などの様々な方程式に支配された世界は、アナログ回路と同様そのままではデータフローとはなり得ない。アナログ回路からデジタル回路への階層化と比較するならば、化学反応においても可逆なものと同様に非可逆なものがあるだろうか。そして、細胞の中にレジスタに相当するものがあるだろうか。答えはイエスである。

化学反応には可逆性のものと非可逆性のものがあることはよく知られている。もし可逆性の化学反応しか無かった場合には、やはり細胞内はアナログ回路の時と同様、物質濃度が波打ちながら漂うしかない。しかも拡散方程式によってやがてはどんな物質も一様な濃度になってしまう。いわゆる平衡状態、死である。ここにシグナルのフローが生じるためには非可逆な変化が必要である。膜タンパク質の能動輸送や、リボゾームによるRNAのタンパクへの翻訳といった化学反応は非可逆である。このような非可逆な化学反応により信号は単一方向へと伝えられる。当然化学反応には可逆変化も存在する。可逆変化は受動素子が能動素子をサポートするように、非可逆変化をサポートするように働き、受動素子が能動素子にドライブされるように、可逆変化は非可逆変化にドライブされるように働く。

次に、生体膜は内部と外部を隔離し、お互いが影響し合わないよう制御す

¹⁷Gbit Ethernetは4ビットを一度に送るバス形式となっている。

る。小胞の中で起こっている化学変化は外界と一定時間隔離される。外界にとってこの間の小胞内での変化は「無意味」である。そして時期が熟すと結果のみが、膜タンパク質によって、あるいはサイトーシスによって外界へ放出されたり別の小胞へ送り込まれる。これは膜という時間的隔離装置によってレジスタと同じ役割を果たしている。この意味で細胞内の諸現象はデータフローと酷似している。

さらに細胞では、時間的過渡期の制御だけではない。化学反的過渡期の制御も行われる。たとえば、図 2.17、図 2.18 では、合成されたタンパク質を小胞内に転送するメカニズムを説明した。タンパク質がその機能を発揮するためには、タンパク質の鎖が折り畳まれていなければならない。しかしアミノ酸の鎖が長いタンパク質になると、そのままの電気的特性に任せておくとどのように折り畳まれるかは運次第であり、小胞内に注入されたタンパク質は目的の折りたたみ状態のものとそうでないものが混在することになる。目的の折り畳み状態でないものは再度折り畳まれたるか、または選別され廃棄される。

2.3.8 連続性と離散との関係

デジタル回路がアナログ回路から作ることの出来る背景にもう一つ条件がある。一見矛盾するように見えるが、それはアナログ回路が微分方程式で記述化可能であるという点である。ある性質が微分方程式で記述できるということは結果得られる波形には、数学的に正則であるという、連続性よりもさらにきつい性質が備わっている。直感的に言えば、その波形の変化は十分に緩やかで滑らかであるということである。厳密に言えば、与えられた関数 $f(x)$ がある点 $x = x_0$ で正則であるということは、 $x = x_0$ の点の近傍でこの関数が無限階微分可能であり、この点の周りでのテーラー展開¹⁸ が収束し、その値が関数 $f(x)$ の値と一致するということである¹⁹。

¹⁸テーラー展開：

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n \quad (2.8)$$

¹⁹厳密な議論をしようとする複素関数論を展開しなければならないが、直感的な議論をするならば、次のようである。たとえば、微分方程式

滑らかであるということは、信号の変化が予測しやすいということである。それは逆に滑らかでない信号を見てみるとよくわかる。たとえば、

$$y = x \sin \frac{1}{x} \quad (2.10)$$

という関数を考えてみよう。これをグラフ化すると図 2.55(b) となる。この関数は $x = 0$ の周りで連続であるが、微分不能、非正則である。たとえば、電圧 x が入力されると出力 y がこのような関数となる素子があったとするとどうなるだろうか。入力電圧が $x = 0$ の点を通過するたびに出力は正負に大きく振れる。正確に $x = 0$ でなくとも、0 にとても近い値を取る時、多少の誤差で出力は大きく振れることになる。到底、入力に対して安定的な出力は期待できなくなる。(b) は $y = x^2 \sin 1/x$ である。こうすると、 $x = 0$ の周りで 1 階微分可能になる。多少は緩やかかつ滑らかな曲線となるが、それでも、0 の周りでは不安定と言ってよい。このような素子は性質が関数で与えられ、確かに決定論的な動きをする。しかし実際に安定的な動作は期待できず、確率的な動きをする素子と考えてよい。

AM ラジオが受けた電波から放送音声とりだせるのは、AM ラジオの回路の性質が微分方程式によって特徴づけられているため、受信電波と出力音声の間に正則的な関係があり、その両者に予想可能性が設定されているためである。

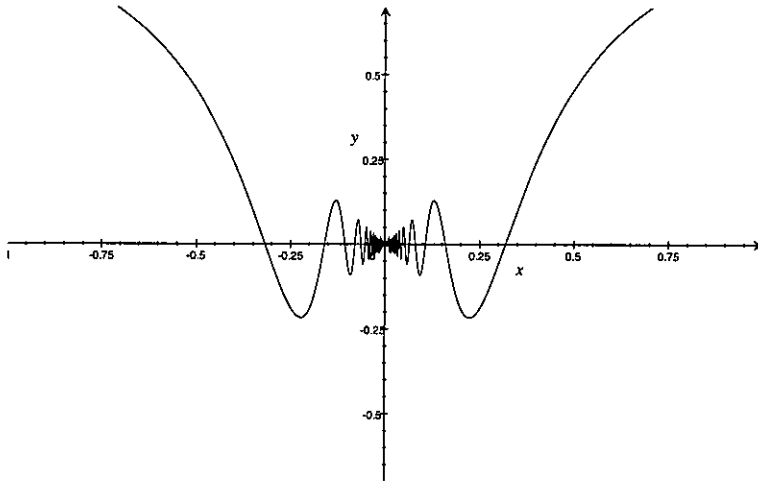
ある点からある点に変化する正則な変化量があったとする。この変化量は連続的であるだけでなく、滑らかかつ緩やかな変化量であるために、始点と終点が十分に近い時、始点にいるときから終点の位置がある程度予想できる性質が生まれる。図 2.56 を見てみよう。(a) は正則な関数で、二つの点の間がどうなっているか予測可能である。一方、(b) では点の並びとそれを載せている非正則な

$$f(x) = \frac{df(x)}{dx} \quad (2.9)$$

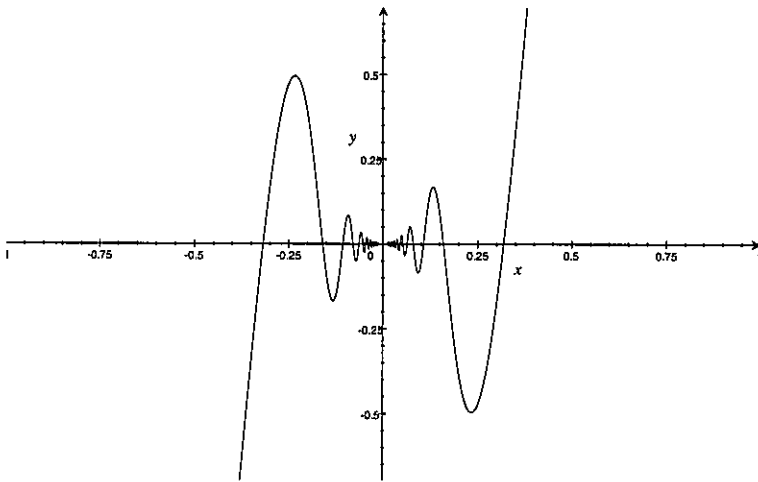
を考える。これは、自分自身を微分すると自分自身になる x の関数 $y = f(x)$ を求めようとするものである。この方程式は、右辺で $f(x)$ を 1 階微分しようとしているので、解は 1 階微分可能であることを要請している。

$f(x)$ が n 階微分可能で、 $n + 1$ 階微分可能ではないということはある得ない。なぜならば、 n 階微分した結果も $f(x)$ なので、さらに 1 階微分可能だからである。従って f は無限階微分可能である。実際、 c を定数として、上記方程式の解は $f(x) = ce^x$ であり、これは正則関数である。

このように微分方程式は微分階数の違う式を組み合わせて作られた方程式であるために、解は無限階微分可能性を暗黙のうちに要請されている。



(a) $y = x \sin \frac{1}{x}$



(b) $y = 10x^2 \sin \frac{1}{x}$

図 2.55: 非正則関数

関数の間に何らの関係性も見いだせない。このような予測可能性は、関数のもつ膨大なバリエーションを一気にリダクションする。連続な関数上で高々可算個、あるいは有限個の点で関数の性質をうまく表してしまうことができるようになる。つまりアナログ信号の量子化というデジタル化の一つの手法が生まれる²⁰。このように微分方程式という連続性に関する性質は暗示的に離散化、デジタル化の可能性を示唆するものなのである。

もっとも、対象となる信号の領域が二次元、三次元と次元が大きくなってくると微分方程式の解は全域で正則とはなり得ず、ある点にしわ寄せを生じることがある。つまり特異点である。台風の目は、流体方程式を解いたときの非正則点である。この周りでは何が起きるか分からない。バタフライ効果は計算機のパワーの小さかった時代に、計算誤差の累積による異常解として言われたが、計算が緻密になってきた現在においても、非正則点近傍における解の挙動は予測困難である。

²⁰式(2.8) テーラー展開は $(x - x_0)^n$ という、高々可算個の n 次関数の一時結合で表されている。つまり、テーラー展開を

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n \quad (2.11)$$

という式で表すならば、 a_0, a_1, a_2, \dots という高々可算個の数の組みによって一つの正則関数を表すことができる。これは別の言い方をすれば、正則関数は、グラフ平面上にプロットした高々可算個の点を通る曲線として性質づけられる。たとえば、ベジエ曲線やスプライン曲線などである。

このような n 次関数のような一次結合の基本となる関数を基底と呼ぶ。このような関数の中で直交性という性質をもつ関数群のなす一次結合をヒルベルト空間と呼ぶ。その代表例がフーリエ級数である。

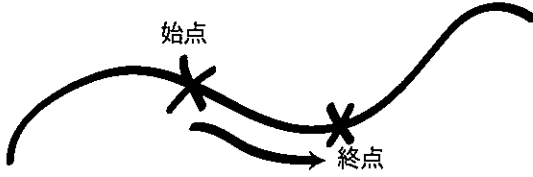
$$f(x) = \sum_{n=0}^{\infty} (a_n \sin 2\pi n x + b_n \cos 2\pi n x) = \sum_{n=0}^{\infty} c_n e^{2\pi i n x} \quad (2.12)$$

信号処理の人たちには、こちらの方がなじみ深いかもしれない。時間領域の微分方程式をフーリエ変換すれば、周波数領域の関数の代数方程式へと展開できる。だが、周波数領域へ展開できる条件として、解および解のフーリエ変換が L^2 収束する必要がある。 L^2 収束とは、

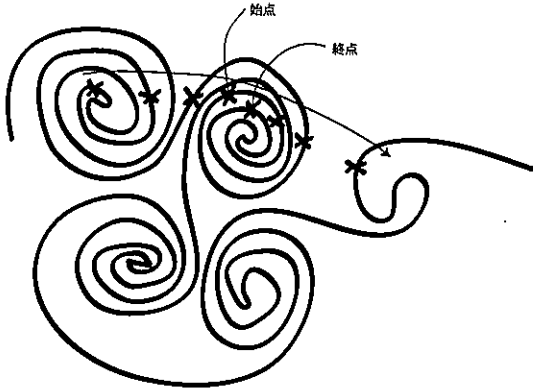
$$\int_{-\infty}^{+\infty} |f(x)|^2 dx < +\infty \quad \text{または} \quad \int_{-\infty}^{+\infty} |\hat{f}(x)|^2 dx < +\infty \quad (2.13)$$

という性質である。 \hat{f} は f のフーリエ変換で、俗にいう信号スペクトルである。この性質があると、 $x \rightarrow \pm\infty$ で、 $\hat{f}(x)$ は 0 に収束する。これは電子回路の信号には十分に高い周波数帯域は無視できることを意味している。高周波がカットできるということは、次のサンプリング定理が適用できる。

信号に含まれる周波数成分の最高値が ω である場合、 2ω 以上の周波数で時間方向に量子化された離散的な関数からは、再びもとの信号を復元できる。



(a) 正則であれば十分近い始点と終点の「間」は予測可能



(b) 正則でない、始点と終点が近づいてもその「間」は挙動不審

図 2.56: 正則関数と予測可能性

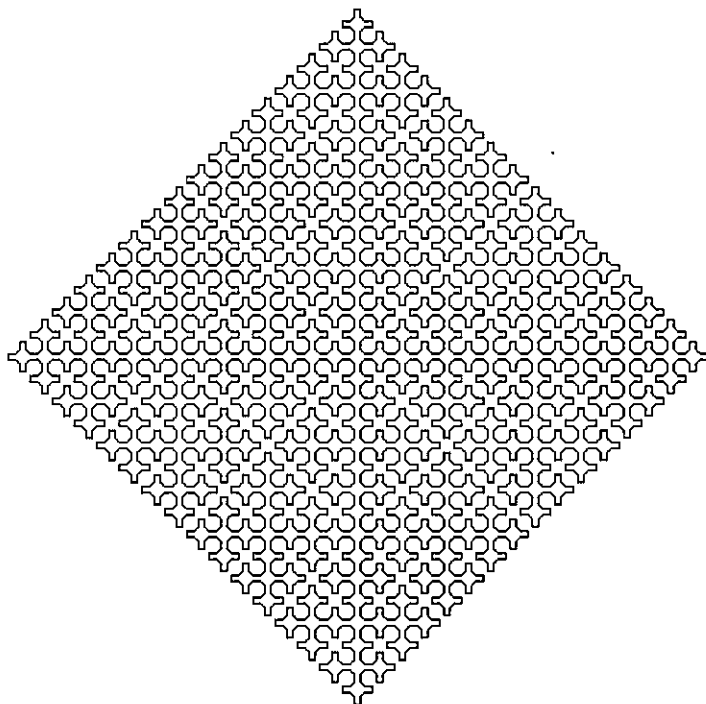


図 2.57: ペアノ曲線

<http://www001.upp.so-net.ne.jp/seri-cf/gallery/f12/f120204.html>

さて、最後にとっておきの非正則関数をお見せしよう。図 2.57 はペアノ曲線である。ペアノ曲線は、平面を埋め尽くす連続曲線である。平面上の任意の 2 点がどんなに近くても、それがペアノ曲線上でどこにくるか、近い点にくるかどうかは予測不可能である。決定論的でありながら事実上のランダムであるが、なにか全体的に美しい。正則関数は微視的にも中間的にも、巨視的にも正則である。しかしペアノ曲線は、微視的な決定論的連続性、中間的には事実上のランダム、巨視的な法則性、という階層ごとの違った性質を醸し出す。このような図形をフラクタルと呼ぶ。章をあらためて議論しよう。

2.3.9 アルゴリズム

さて、オートマトンに代表される、手順、手続き型の方法によって機械をモデル化する手法とデータフローや関数という概念によって機械をモデル化する手法を見てきた。機械がどのように動作するかを記述する方法、つまりプログラムにはこのように様々な手法がある。しかし、記述法、あるいは実現方法が異なっていたとしても、振る舞い方は同じといえる場合がある。例えば、数値計算をする機械は、機械式で実現されていても、あるいは電子的に実現されていても、また、オートマトンで記述されていても、データフローで記述されていたとしても、計算結果は同じでなければならない。

機械が実現する動作、手順や手続きをなんらか可読な方法で記述したものがプログラムである。その手法がオートマトンやデータフローということになる。異なる記述であっても同じ動作をするプログラムというのがあるはずだ。「同じ」動作として指し示された、動作そのものがアルゴリズムである。その動作を記述したものが特になくても動作そのものは成り立ちうる。時計のアルゴリズムと言った場合、時計のプログラムが特になくても時計は成り立ち得る。もっと言うならば、時計という物体がなくとも、だれでも時計の動作を思い浮かべることが出来る。

アルゴリズムすなわち動作というものは、動作開始があつて終了がある。そしてその間入力されたものに対して出力がある。これは何度やっても、同じ入力に対しては同じ答えが返ってくる、と約束しよう。もう一つ条件がある。人間が作れるものでなければならない。そのため有限なものである。しかしこれ

がくせ者である。

たとえば、次のような機械を考えよう。開始ボタン、終了ボタン、次ボタンの3つのボタンがあり一つの0から9まで表示可能な表示装置がついている。開始ボタンが押されるとすべてがリセットされ機械は初期化され、表示装置に3と表示される。次ボタンを押して続けると、1,4,1,5,9,2,...と順次円周率の桁が点灯されて行く。終了ボタンを押すと、そこで表示は消え機械は動作を終了する。この機械は動作の開始と終了があり、一応、何度やっても同じ結果を返す。円周率の魔法のランプと名付けよう。一見簡単そうなこの機械を厳密な意味で人間は作ることは出来ない。なぜならば、終了ボタンはいつ押されるかはわからないからだ。何百億年後に終了ボタンを押すというのでもいい。何兆年後に押されても良い。いつ返されるかわからない借金のように「いつかは押される」ということだけが保証されている。内情は、とにかく円周率を限りなく計算し続ける機械である。どんなコンピュータでも、次ボタンが押され続けて行くと、いつかはメモリが足りなくなり破綻してしまうのである。

有限であるという枷をどう考えるか、この考え方によって「アルゴリズム全体」がなにかが決まる。一つの考え方は有限状態オートマトンである。もし高々可算個状態オートマトンというのが実現できるのであれば、この魔法のランプは実現できる。現実的（そうな）一つの解法としては増殖するコンピュータである。メモリが足りなくなれば勝手に自己増殖してメモリを増やし計算をし続ける。増殖スピードがボタンを押すスピードに追いついて行きさえすれば実現可能である。しかし、地球上には、コンピュータを製造する材料や増殖したコンピュータを置く場所には限界がある。太陽系や宇宙全体に広げれば可能かもしれない。宇宙全体の資源が有限であるならば、残念ながらこの高々可算個状態オートマトンは実現できない。

そう考えてみると数学という学問は変である。「高々可算個」というのは数えることが出来る数という意味であり、自然数や整数のことを指す。1,2,3,...と数えて行くことが可能であるということの意味し、「高々」と言いつつ無限なのである。数えることの出来ない無限の例としては実数や関数の個数がある。こういった途方もない量の敵を相手にしながら、数学は様々な定理を証明して行くが、その過程は有限でしかない。つまり証明過程はアルゴリズムだ。そもそも有限な人間が無限な対象を敵にまわしている訳で、とうてい太刀打ちできそう

にない。結局、デーゲルの不完全性定理が成り立ってしまう。

さて、有限状態オートマトンで表すことの出来る動作全体がアルゴリズム全体と言えるならば、データフローで表すことの出来る動作全体はそれより大きいのだろうか、小さいのだろうか。データフローという図示の方法は少々曖昧である。データフローから導出した関数という概念を厳密に定義したものが、チャーチによる λ 計算という概念である。 λ 計算は世界はすべて関数であるという発想から始まり、いくつかの関数の変換、同値規則を導入したときに、計算可能な範囲を証明する理論である。3つ目にアラン・チューリングが示したチューリング・マシンという概念がある。チューリングは計算機が停止するかどうかをあらかじめ調べる一般的な方法がないことを示した。

最終的に達成されるべき問題は、この有限状態オートマトンと λ 計算、チューリング・マシンが相互変換可能であり、従ってそれぞれで定義される動作全体の集合は一緒であるということである。これによって三つの手法どれをとっても、持っている状態は有限であり、計算可能な範囲は同一であり、計算が停止するかどうかはあらかじめ知ることの出来る一般的な方法はないということが言えるのである。チャーチはこれをもって、この三つと同値な動作をもって定義される手法をアルゴリズムと名付けようと提案した。

この同値関係とチューリング機械停止問題の証明は別の専門書にゆだねよう。我々の関心事は生命はアルゴリズムかということである。多細胞生物は話がややこしい。ここでは細胞はアルゴリズムかという程度にとどめても十分難問である。小林秀雄の随筆に『将棋の神様』という文章がある。将棋のすべてを知り尽くした神様が二人対決したらさぞかし見応えのある接戦になるだろう、そう考える筆者に中谷宇吉郎は「そんなら、結果は出るさ。無意味な結果が出る筈だ」と答える。

細胞がアルゴリズムであるか否かを考えるにはオートマトンをまず想定すれば良い。オートマトンの定義は、2.3.2に示した。この前提条件について検証しよう。最初に細胞の中には偶然性がある（といわれている）。これはアルゴリズムにとってはまずい。しかし、機械にとっても偶然性がいろいろ役に立つことがある。そのためにアルゴリズムに付随して取り入れられることがあるのだ。動作を図 2.58 に示すように偶然性の部分である確率過程と必然性の部分である決定論的部分に分けることが出来る。



図 2.58: 確率過程を含む動作

このような分解を経て元の問いは、細胞の諸動作の確率過程を除いた部分がアルゴリズムになるかという問いに変形できる。次に、アルゴリズムは開始と終了がある。これも細胞はとりとめ無く分裂して増えて行く。どこが切れ目か分からない。そこで、細胞分裂から次の分裂または死までと定義しよう。次に、入出力が有限パターンということを満たさなければならない。まず、細胞へはどのような入力信号があるのだろうか。開始された細胞への最初の入力は、親細胞から受け継がれる形質である。それに続いて、外部から入力される様々な化学物質、光や熱などの物理量である。出力信号は、通常では外部へ出力される様々な化学物質であり、分裂するときは、二つの娘細胞に引き継ぐ形質である。これらのパターンは無限ではなかろうか。2.3.8節で議論したことを思い出そう。これらの物質や物理量の挙動が微分方程式に従うならば、物質など一見連続量であるものは正則な関係となる。これは、十分細かく量子化すれば、離散化可能であり、図 2.58 の決定論的部分の入力信号として、高々可算個の範囲までは押さえられる。もしたとえ微分方程式に従わない挙動があったり、微分方程式の特異点など、非正則な部分があったとしても、事実上の確率過程と考えることが出来る。出力も同様に考えられる。入出力が高々可算個であるということは、入力記号を整数で番号付けすることが可能である。整数はたとえば、10 個の記号で表すことが出来るので、本来の入力記号一個を 10 個の数字と一つのセパレータ合計 11 個の記号列として表すことが出来る。全体はセパレータで区切られた 11 パターンの記号列に置き換えられる。これでとりあえず前提条件はクリアした。

今一度、中には感度の良い細胞が生まれ、入出力は一定の量子化などで表せないだろうという反論があるかもしれない。それだったら、量子化精度を $n/2^m$ という有理数にすればよい。まずなんらかの入力の入力値 $(n - 0.5)/2^m$

$\sim(n+0.5)/2^m$ に対して、 $n/2^m$ という値を対応させる。感度のいい細胞が出現したならば m を十分大きくしてやればよい。これに対して実際の入力は、 n を表す 10 進数、セパレータ 1, m を表す 10 進数、セパレータ 2, ... という入力列を対応させる。このように考えれば、12 個の記号があれば、どんな感度の細胞にも対応可能である。

最後に残るのは、決定論的部分が有限状態かである。細胞内部も微分方程式に従うならば、状態数は高々可算個であるということまでは言えそうであるが、それではアルゴリズムということにはならない。とりあえず細胞はブラックボックスと見なそう。もしあなたが、円周率の魔法のランプに似た箱を渡されたとしたら、あなたはその箱の中身がアルゴリズムであるかどうかは判別できるだろうか。「次」のボタンを押し続けると必ず N 回目に、 E という表示が現れ、「終了」ボタンを押さない限り動かなくなる、というものであれば、円周率の魔法のランプに似ているだけで中身はアルゴリズムであると言えよう。しかし「次」ボタンを押して行ってもなかなか停止しない場合はどうだろうか。チューリング機械停止問題を読み替えば、停止しないからと言って、それがアルゴリズムではないとは言えないということである。つまりあなたは、円周率の魔法のランプが本物か偽物（つまりアルゴリズム）かを判別することは出来ない。とどのつまりは細胞の決定論的部分はアルゴリズムか否かは判別が出来ない。

もしどちらかの決着を付けたいと思うならば、ブラックボックスを開け、細胞の中身をくまなく知り尽くす必要がある。しかし進化する細胞では、それを知り尽くすことは出来ない。進化する細胞を追い続けていくことは「次」ボタンを押し続けているのと同じである。無限に進化し続けるのか、進化がどこかで停まるのか、せいぜい百年に満たない生しか与えられていない我々にとって知るすべは無い。

とどのつまり、円周率の魔法のランプの真贋は本質的な問題だろうか。現在の技術では、数十億桁の円周率が求められている。これを記憶したメモリとそれを 1 桁ずつ読み出す装置を備えた小さな箱を作ることくらいは、ちよつとした電子工作で可能だ。この偽魔法のランプをもってしても、1 秒に一回「次」ボタンをおしていったとしても、破綻するまでに百数十年かかる。真贋が判定出来る前にあなたの命が尽きるだろう。

アルゴリズムであるかないかの判定が出来ないということは、どちらを前提

に論を進めても特に矛盾は起きないということである。だが、細胞の決定論的部分がアルゴリズムでないと言うことのメリットはなんだろうか。アルゴリズムであると言い切ってしまうことの方のメリットが大きいのではないか。アルゴリズムであると言い切ることは、細胞の進化がどこかで停まることを意味しているかもしれない。しかし、十分に大きなアルゴリズムを想定すれば、偽の細胞と本物の細胞の差が明らかになるのは、我々が死に絶えたとつとあのことである。少なくとも細胞の動作をアルゴリズムは十分近似できるはずだ。

アルゴリズムであるということは、細胞の動作をなんらか記述できるということの意味している。よく遺伝子はプログラムであると言ったりする。確かに現在の遺伝子は、我々がいまアルゴリズムかどうかを議論しているものの部分を形成するものといえる。しかし、我々が議論しているのは、それがたとえ、何百億年後破綻するとしても、そこまでの細胞の進化をも含めてシミュレートできる記述である。その記述が可能だということだ。しかし、記述できるからといって、その記述が現在の人間に理解可能かどうかは分からない。分子生物学は、この人間に理解不能な記述からある部分を切り出し、理論化しているようなものである。

シュレディンガーは『生命とは何か』において生命を未知なるものとしつつも、いまは解明されていない物理学や化学の知識により説明ができるものであろうと言っている。細胞の状態が有限個か無限個かという議論はともかく、アルゴリズムとなる前提条件、有限個／高々可算個の入出力という前提条件を考えると、細胞の決定論的部分がアルゴリズムでないとする立場は、背後にある微分方程式、つまりは物理学や化学による法則を否定するものである。つまりアルゴリズムを否定することはヴァイタリズムとならざるを得ない。

ここからもう一つ見えてくるものがある。我々は言語を持っている。言語、特に記述される言語は離散的な存在であるがゆえ、連続的な対象である自然界を言語で記述するときどうしてもそこからこぼれ落ちるものが出てくる。ある意味言語のもつ限界としてこのような議論がなされるときがある。しかしおそらくは、それは反対なのだ。

連続的な森羅万象は、おそらく解析学的構造を持っている。解析学的構造は連続的な世界をむやみに連続とさせはせず、先に議論した正則性や L^2 収束、ヒルベルト空間といった骨格構造をがっちりとするものである。自然界が既に

このような骨格構造を内包しているが故に、我々は対象を離散的に認識可能なのだ。もうすこし厳密に言うならば、認識された離散的な記述と連続的な対象の間に、ある種の信憑性を帯びたアナロジーが成り立つようになる、つまりは言語が成立するということだ。我々脊椎動物に骨が無かったら、ぶよぶよと無差別に形を変え動き回る肉のかたまりである。しかし骨格構造があるが故に、動作のメリハリがはっきりし、脊椎動物の体をなすのである。

2.3.10 モンテカルロ法

細胞はアルゴリズムかという前節の議論は、その決定論的部分と確率過程との折り合いの中で、決定論的部分をアルゴリズムとしてもよいだろうという結論になった。ところで、図 2.58 のように、確率過程を用いたアルゴリズムには考え方が二つある。一つは、アルゴリズムによって実現したい問題に本質的に確率過程が含まれているので、このようなスキームが必要である場合である。たとえば、商店街や百貨店におけるお客の動態シミュレーションを行うことによって、立地や商品レイアウトと購買との関係を導き出す、といったとき、客の動態は本質的に確率過程を含んでいて、これに確率過程が使われるのはそんなに不思議なことではない。一方で、解は決定論的に決まるはずであるが、確率過程を導入することで計算時間の短縮や、誤りの軽減といったメリットを引き出す方法である。このような考え方は注目に値する。

まずは一番簡単な例からいこう。図 2.59 は 1m 四方の模造紙に、黒いねじ曲がった図形が描かれているという想定である。この黒い図形の面積を求める方法を考えてみよう。区分求積法を知っている人は、方眼を描き、黒い図形上に乗るマス目を数えることによって、面積を求めるかもしれない。だが、方眼を描くなんてめんどくさい。もっと無精なやり方がある。この模造紙を壁にはり、できるだけ遠くから、目をつぶってダーツを投げる。出来るだけ四方八方に投げる。そして模造紙上の 1m 四方の枠に入ったものと、黒い図形上に入ったものの比率を求めるのである。ダーツがランダムに投げられていると仮定できれば、この比率は面積比率に近づいて行くはずだ。

もう少し技巧的な例を示そう。同じ長さの楊枝を N 本用意する。模造紙に楊枝の長さとも一致する間隔で平行線を引詰める。この模造紙上に千本の楊枝を

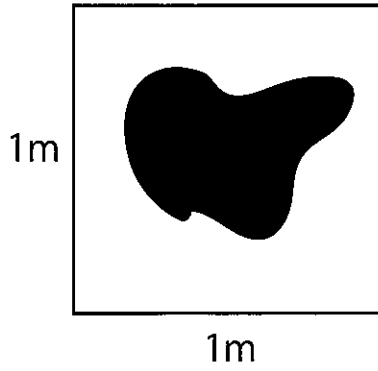


図 2.59: 図形

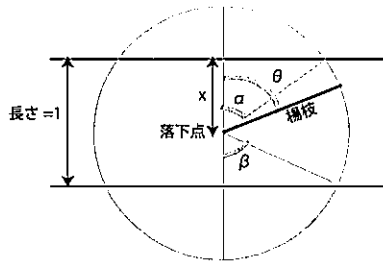


図 2.60: 楊枝と平行線の関係

ばつとまんべんなくばらまく。楊枝と平行線とが交わったもののみの本数を数える。もしランダムに楊枝がばらまかれているならば、その値は $2N/\pi$ に近づく。つまり円周率が求められる²¹。

このように確率過程を用いて何らかの計算を実現する「アルゴリズム+確率過程」をモンテカルロ法という。実に年代を感じさせる命名であるが、おそらく読者のご想像通り、カジノで有名な都市の名前にちなんでいる。モンテカルロ法の魅力は、この二つの例もそうであるが、「緻密な計算」に比べて、非常にアバウトであるにも係わらず、思ったよりも手軽に目的の解に近づけることだ。言ってみれば抜き打ち検査のようなものである。

決定論的な方法でも解ける解が、モンテカルロ法でも得られるというだけではない。決定論的な方法では近づけないかもしれない解を的確に導出することもある。数値計算の分野では最適化問題という分野がある。これは、数学的には、複雑な関数の最小値を求める問題である。有名なものには最小二乗法というものがある。これは、おおよそ直線状にばらついている点の集合から、最もそれに近い直線を求める問題である。直線の位置と角度から各点との隔たりの二乗和、つまり誤差を最小にする問題である。単純な問題であれば解の公式がありそこに必要な値を代入すればよいだけである。ところが複雑な問題ともなってくると、極小値が複数存在し、得られた答えが必ずしも最小とは限らない。

図 2.61 は、2つの極小をもつ関数である。この最小値を求める方法を考えてみよう。一応これは多項式なので、1階微分も3次の多項式で、0になる3点を方程式で解き、得られた3点のうち一番小さい点を答えとすればよい。しかしこの方法は、問題が複雑になれば通用しない。これを急降下法という決定論的方法で最小値を求める方法を考える。これは以下のような方法である。

1. 変数 x を任意の初期値で初期化する。たとえば、2

²¹図 2.60 は、一本の楊枝が落下したときの様子を表している。まず話を簡単にするために、楊枝の長さ=平行線の間隔=1としよう。楊枝の端点が、平行線から x の距離のところへ落下したとする。その確率は Δx とする。そして、平行線と垂直方向に対して角度 θ の方向へ倒れた。この図では平行線と楊枝は交わっていない。平行線と交わるためには、 $0 \leq \theta \leq \alpha$ または、 $\pi - \beta \leq \theta \leq \pi$ である必要がある。従って、端点が落下したところで、平行線と交わる確率は、 $(\alpha + \beta)/\pi$ である。求めるべき全体の確率は、

$$p = \int_0^1 \frac{1}{\pi} (\alpha + \beta) dx = \frac{1}{\pi} \int_0^1 \cos^{-1} x + \cos^{-1}(1-x) dx = \frac{2}{\pi} \quad (2.14)$$

である。

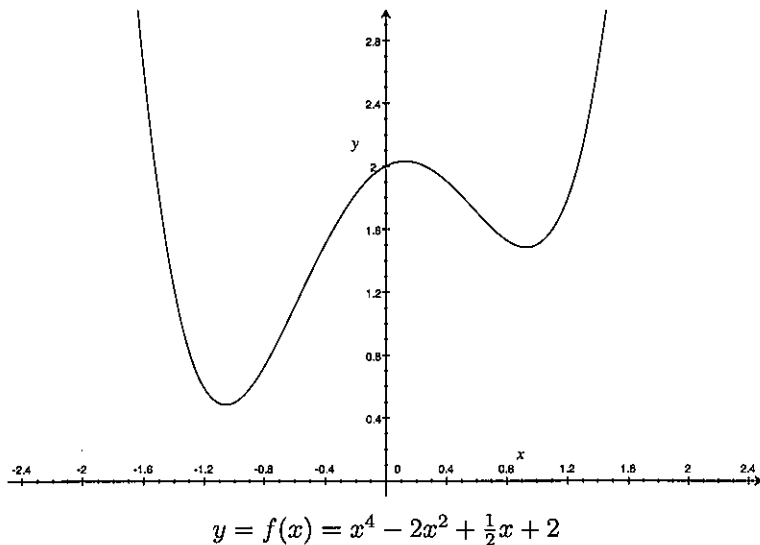


図 2.61: 例

2. 変数 p を 1 とする。
3. 変数 Δ を x の誤差として十分小さい値にセットする。
4. $f(x), f(x-p), f(x+p)$ を求める。
5. $f(x) > f(x-p)$ ならば、 $x ::= x-p$ とする。
6. $f(x) > f(x+p)$ ならば、 $x ::= x+p$ とする。
7. 以上 5,6. のどちらでもない場合は、 $p ::= p/2$ とし、4. から繰り返す。
8. p が Δ より小さくなったところで、そのときの x の値を答えとする。

このプログラムを実際に動かしてみると、初期値によって最終的に収まる解が、二つある極点のいずれかになるのである。つまりこのプログラムを1回試行するだけでは、大きい方の極点に収束し、最小値へたどり着けないかもしれないのである。図 2.62 は、初期値 2 で上記プログラムを実行したところであ

x	p	$f(x)$	$f(x-p)$	$f(x+p)$
2	1	9	-0.5	64.5
1	1	-0.5	0	9
1	0.5	-0.5	-0.1875	1.3125
1	0.25	-0.5	-0.43359375	-0.05859375
1	0.125	-0.5	-0.507588359375	-0.366043359375
0.875	0.125	-0.507568359375	-0.43359375	-0.5
0.875	0.0625	-0.507565359375	-0.478256225585938	-0.5165860303710938
0.9375	0.0625	-0.516586303710938	-0.507588359375	-0.5
0.9375	0.03125	-0.516586303710938	-0.514937400817871	-0.511839866638184
0.9375	0.015625	-0.516586303710938	-0.516518533229828	-0.515055596828461
0.9375	0.0078125	-0.516586303710938	-0.518746874898672	-0.516026135534048
0.9296875	0.0078125	-0.516746874898672	-0.516518533229828	-0.516715869974687
0.9296875	0.00390625	-0.516746874898672	-0.516680653905496	-0.51674350850302
0.9296875	0.001953125	-0.516746874898672	-0.518725834877345	-0.516748290420765
0.9296875	0.0009765625	-0.516746874898672	-0.516739382727792	-0.51674360850302
0.9306640625	0.0009765625	-0.516748290420765	-0.518746874898672	-0.516748290420765
0.9306640625	0.00048828125	-0.516748290420765	-0.516748343539746	-0.516746712942165
0.93017578125	0.00048828125	-0.516748343539746	-0.516746874898672	-0.516748290420765
0.93017578125	0.000244140625	-0.516748343539746	-0.518747799276796	-0.516748507362703
0.930419921875	0.000244140625	-0.516748507362703	-0.516748507362703	-0.516748290420765
0.930419921875	0.0001220703125	-0.516748507362703	-0.518748473026528	-0.516748446507655
0.930419921875	6.103515625e-05	-0.516748507362703	-0.518748502099979	-0.516748488836921
0.930419921875	3.0517578125e-05	-0.516748507362703	-0.516748507701249	-0.516748501074705
0.930389404296875	3.0517578125e-05	-0.516748507701249	-0.516748502099979	-0.516748507362703
0.930389404296875	1.52587890625e-05	-0.516748507701249	-0.518748505639677	-0.516748608275618
0.930404663085938	1.52587890625e-05	-0.516748508275618	-0.516748507701249	-0.516748507362703
0.930404663085938	7.62939453125e-06	-0.518748508275618	-0.516748508174339	-0.516748608005076
0.930404663085938	3.814697265625e-06	-0.516748508275618	-0.516748508271465	-0.516748608186825
0.930404663085938	1.9073486328125e-06	-0.516748508275618	-0.516748508285196	-0.516748608242841
0.930402755737305	1.9073486328125e-06	-0.516748508265156	-0.516748508271465	-0.516748608275618

図 2.62: プログラムの実行結果 1 回の試行 初期値 $x = 2$

初期値	x の収束値
2	0.930402755737305
1	0.930402755737305
0	-1.05745315551758
-1	-1.05745315551758
-2	-1.05745315551758

図 2.63: 初期値を変えての試行結果

る。1 行が、4. の実行後を表示したものと考えると良い。最小値ではなく、極小点 $x = 0.930402755737305$ に収束していることが分かる。図 2.63 は初期値を変えて、試行した結果である。初期値が、0 から小さい時、最小値へ収束することが分かる。

この問題を回避する方法には、初期値をランダムに生成し、実行してみるという方法がある。これも一つのモンテカルロ方である。しかし極点がどれくらいの範囲に広がって存在するのか分からない場合は、初期値をランダムに生成と言えども、どのくらいの範囲で生成すればよいのか分からない。

ここである種の突然変異的な手法がある。1 回の試行のための初期値 I_n を記

初期値	x の収束値	
試行 1	$I_0 = 2$	0.930402755737305
試行 2	$I_1 = -4.41758346557617$	-1.05745315551758
試行 3	$I_2 = 15.7431983947754$	0.930402755737305
試行 4	$I_3 = -73.1335754394531$	-1.05745315551758
試行 5	$I_4 = 359.32315826416$	-1.05745315551758

(a)

初期値	x の収束値
2	-1.05745315551758
1	-1.05745315551758
0	-1.05745315551758
-1	-1.05745315551758
-2	-1.05745315551758

(b)

図 2.64: モンテカルロ法を入れた結果 初期値=2

憶しておく。そして得られた極値 x_n との間で

$$I_{n+1} = x_n + 5(x_n - I_n) \quad (2.15)$$

という計算をする。 I_{n+1} を新しい初期値として新たな試行をする。このようにして何度か試行し得られた極値のうち最小のものを答えとする。定数 5 は 2 より大きければなんでもよいだろう。この式 (2.15) の心は、「初期値から極値へ落ちてきた道筋の方向へ、さらに 5 倍先へ飛ばす」ということだ。この例のように 1 次元の関数の極値ではあまりメリットが感じられないかもしれないが、これが多次元になると思いのほか良い結果をもたらすことが経験上知られている。

図 2.64(a) は、最初の初期値を 2 として、この突然変異的な試行を 5 回繰り返した場合の各試行の初期値と x の収束値を表したものである。初期値をみるといろいろなところへ飛んでいることが分かる。それにともない x の収束値もいろいろ変化している。その中には最小点も含まれていることがわかるだろう。(b) は最初の初期値を 2~-2 と変化させたときの最終結果を示したものであり、最小点が安定的に得られていることがわかる。

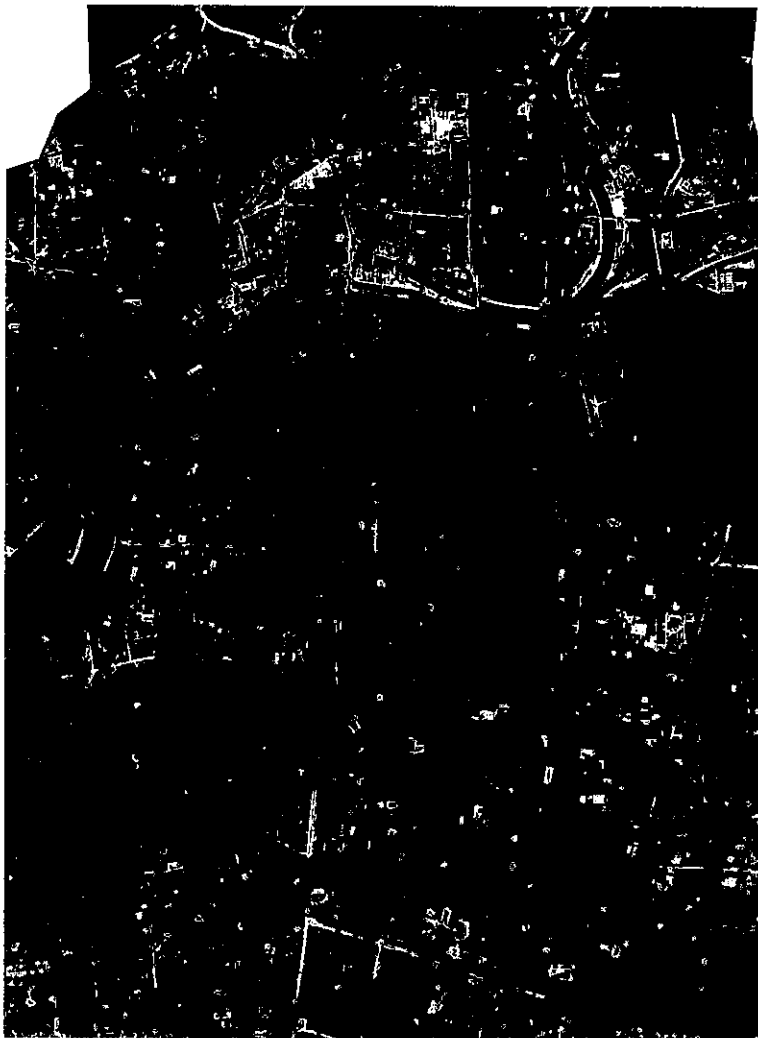


図 2.65: 昭和 17 年 航空写真

図 2.65 は戦前の大阪市、昭和 17 年の航空写真である。実際は数百枚のカットに分かれているものを繋ぎ合わせたものであり、オリジナルデータは拡大して行くと一軒一軒の建家まで判別できるほどの解像度がある。現在の航空写真、いわゆる空中写真測量技術では、航空機による撮影から、全カットを繋ぎ合わせ、一枚の地図を作るまで、ほぼ自動的にできるようになっている。そのためには、撮影地点の位置情報、角度などの情報が、撮影とともに自動的に計測され集積される。戦前の航空写真なので、このように繋ぎ合わせて一枚にするという目的で撮影されたものではない上、傷みも激しく、歪みがひどい。従って、これらを自動的に繋ぎ合わせることは至難の業である。かといって、手動で画像をあわせていくと歪みのために、ある部分を合わせると別の部分が合わなくなり、結局繋ぎ合わせは破綻する。このような時に、モンテカルロ法を含んだ最適化問題は役にたつ。ある程度ラフに手で繋ぎ合わせた後、全てのカットが連続的に接続でき、かつ、最終的な画像の歪みと地理的な位置の狂いが最小になるように、接続点を調節して行く。数百枚の接続点の一座表を入力とし歪みやずれの二乗総和を計算する評価関数を作成し、この評価関数の最小値を与える接続点の位置セットを計算する。こうして作成された図である。

思い起こせば、2.2.10 節でパケットのエラーと再送メカニズムにおいて確率的な議論をした。ネットワークではよく確率的なアルゴリズムが使われる。たとえば、Ethernet や無線 LAN ではパケットをあるノードから別のノードへ送信するとき、パケットは複数のノードがつながっている ethernet ラインや、電波の飛ぶ空間へ放出される。複数のノードが同時に放出する可能性があり、そのときにはパケットは壊れてしまう。このときに良く取られる方法は、送信側ノードは、パケットを送出しながら、同時に受信を行う。送出したパケットと、別のノードが送出したパケットがぶつかった場合、受信されたパケットは破壊されている。このような衝突を検知すると、しばらく間隔をおいて、送信側ノードはパケットをあらためて送信する。しかし、パケットがぶつかった場合、もう一つの送信ノードもしばらく間隔を置いてパケットを送信することになる。もし、この送信メカニズムが決定論的なアルゴリズムで作られていたとすると、再送も再びぶつかる可能性がきわめて大きくなる。そこで、再送間隔はランダムに決めるのが有効である。良く取られる方法としては、パケットが何度もぶつかったときには再送間隔は徐々に長くして行くと同時に、この再送間隔の時

間に 10% ~20% のランダムな揺らぎを載せる。すると何度目かにはおおよそ他ノードに阻害されることなく送信できるタイミングが得られる。

普通山のとっぺんにボールがあるということは不安定な状態であるが、決定論的な方法では、それが、平らなところに正確に置かれていれば起こり得てしまう。山のとっぺんが不安定である、ということは確率過程としてしか説明することが出来ない。いわゆる自発的対称性の破れである。最小値でない極小値、あつてはならない同期といった「招かざる解」を解消するのが確率過程＝乱雑性である。

2.3.8 節では、もう一つ、非正則な場合に生じる、事実上の乱雑性ということを行った。特異点、フラクタルといったことが浮かび上がってくる。このような場合については、章をあらためて議論することにするが、この場では疑似乱数について少々述べておこう。我々がプログラミングをするときに本物の乱数を使うことはあまりない。もちろん本物の乱数を生成する乱数生成器というのは実在する。乱数生成器を実装しているコンピュータはあまりなく、本物の乱数が生成しにくいという問題もある。しかしそれ以上に、本物の乱数でプログラミングされたプログラムはデバッグがしにくいというデメリットがあるからだ。バグをつかまえ修正するためにはバグの再現性は重要なのである。なんども試行することによってバグの性質をつかみ、バグを追い込んで行く。ところが、本物の乱数が組み込まれたプログラムでは、バグの発生もランダムになる。バグの頻度が低い場合は、試行しても現れなくなってしまうことも多々ある。これではバグはとれない。疑似乱数を使っているとバグの再現性が保証できるため、デバッグがしやすいのである。もし、本物の乱数を使うことが要請されている場合、まずは、疑似乱数でプログラムを作り、十分にバグをとってから、本物の乱数に置き換えるといった工夫をしたりする。

2.3.11 細胞のアルゴリズム

細胞を図 2.58 のように決定論的部分と確率過程の協働型のモデルで考えたとき、では、どのような部分が決定論的で、どのような部分が確率過程なのだろうか。ES 細胞や iPS 細胞から、体の各部位の細胞を作り出す研究が盛んになされている。実際に目の網膜の細胞など、実際に可能なことが証明されている

ものもある。これは、人間の手によって、ある種の手順を踏むと、細胞は予定された通りの動作をする、ということを示している。細胞の動作の中にはかなりの決定論的な部分があると言える。一方でDNAはプログラムであるというような言い方をする。しかしまた一方で、DNAは一つの個体の中でも時としてランダムに書き換えられるときがある。どこが決定論的で、どこが確率過程なのか。

今までに述べた分子生物学で明らかになった情報は、ある種の決定論的性質を帯びている。様々な図として書かれる化学反応の連鎖はデータフローそのものである。化学反応の中にはランダムなものもあり得るが、ゴルジ体におけるタンパク質の選別の例として挙げたように、またあまり述べなかったがDNAの転写機構などを見ている、どうも、予期せぬ結果が得られた場合は訂正する仕組みというものも存在するらしい。つまり細胞内の反応の多くは決定論的プロセスとして捉えることが出来そうだ。

2.2.2節で細胞のCPUは化学反応であると言った。そして、DNAとタンパク質の関係を、非活性の情報と活性化された情報という関係で捉えた。上記のように考えるならば、決定論的プロセスを担っているのは、タンパク質=活性状態のプログラムである。しかし化学反応はCPUのようなもので、つまりハードウェアである。これを変更するためには物理科学法則を変更するしかない。ここに論理的な構造を与えるのが、情報の非活性状態を作り出すことのできるDNAではないか。であるならば、確率過程の多くを握るのはDNAであると言える。

実際、DNAには大きくわけて二つの組み替えのメカニズムが備わっている。一つは転移性DNA因子(トランスポゾン)、もうひとつが生殖細胞における減数分裂である。転移性DNA因子はDNA上塩基配列であり、この塩基配列はときどきDNAの別の場所へ転移する。その転移のメカニズムと形態の違いで大きく、DNAトランスポゾンと、レトロトランスポゾンの二つのタイプに分けられる(図2.66)。DNAトランスポゾンは供与部分がすっぽりとぬけ、転移先に移動する、いわばカットアンドペースト型である。そして、レトロトランスポゾンは供与部分を残したまま、転移先に挿入される、いわばコピーアンドペースト型である。

トランスポゾンはある一定の配列構造をもっている。細胞内のDNAの転写

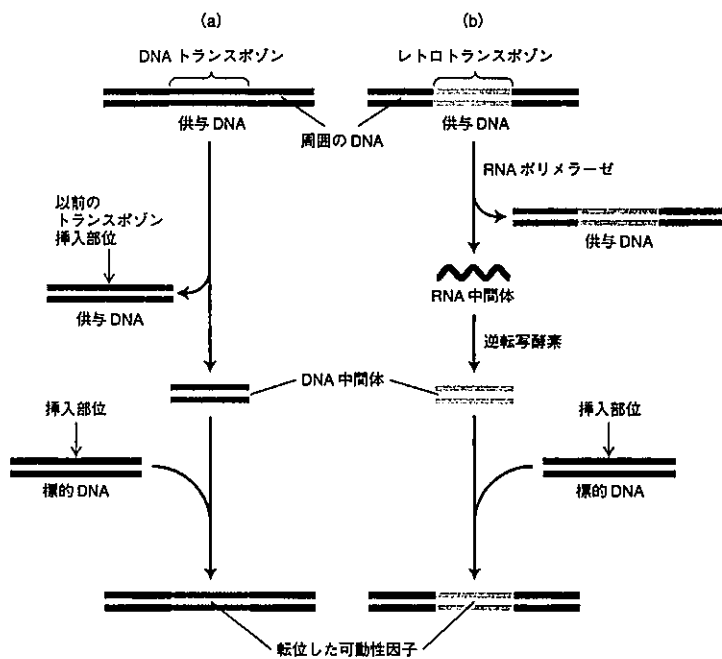


図 2.66: トランスポゾン

のバリエーションがあり、バリエーションを調べることによって進化の歴史がひもとける。

通常トランスポゾン(転写因子)は細胞内においてDNAの上を移動する。しかしレトロトランスポゾンは、その配列の中に、自分を膜で覆う方法をコーディングし、細胞外へ放出する手段を得た。それがレトロウイルスと言われている。レトロウイルスは細胞間で遺伝子を運ぶ。これもトランスポゾンと一緒に、多くの場合は無益か、あるいは疾患をもたらす。しかしごくごくまれに、生物の間を有用な遺伝子を運ぶ役割を果たすかもしれない。特に細胞分裂のみで子孫を増やす原核生物にとってウイルスの存在は大きい。

このようなトランスポゾンの動きを見ていると、イントロンという無意味な領域が重要であると言えよう。我々がもっている98.5%ものイントロンはいわば、Dropbox、あるいは公開領域である。ここへの遺伝子の出入りは自由であり、様々な遺伝子が出入りする。我々は進化の過程で、何食わぬ顔で出入りする「よそのもの」を傍目で監視し、あわよくば有用なものがあればエキソンへ変位させる。このときに、ある瞬間にいままでイントロンだったところを試用してみる仕組みというのがあり得るのかもしれない。イントロンとエキソンの間で配列が交換され、あらたなエキソンが作られた、エキソンシャッフリングと呼ばれる現象を示唆する証拠もあるという(2.68)。[4]にはそのほかにも様々なエキソンシャッフリングが紹介されている。このようなメカニズムが偶然におきるのか、あるいはある種の決定論的なプロセスがあるのかは分からない。[4]では、様々な手法によって、推定遺伝子を25000同定したが、そのうち10000は本当にタンパク質やRNAをコーディングしているという確証は得られていないと言っている。この確証されていないもののうち、いくつかは、イントロンからエキソンへの移行途中段階の試用かもしれない。

ところで、ヒトのDNAの総量はおおよそ3Gbである。「b」とはbase pair(塩基対)である。DNAはA(アデニン)、G(グアニン)、C(シトシン)、T(チミン)という四種類の核酸の配列で出来ている二重螺旋である。二重螺旋の一方の塩基がもう一方の螺旋の塩基と手を結んでいる。これが塩基対である。手の結び方には決まりがあり、GとC、AとTが手を組む。3Gbとは、この対が30億個あるという意味である。4種類の塩基は2bitの情報量である。上記の規則があるために、二重螺旋の一方のコーディングが決まれば他方のコーディ

ングは自動的に決まる。しかし、DNA はその他方のコーディングも別のタンパク質のコーディングとして利用されることもある。したがって別情報とも捉えることが出来る。ヒトの DNA は一本鎖で 6Gbit、両方あわせると 12Gbit の情報量をもつ。とりあえず DVD 一枚に全部収納できてあまりある量である。しかもこの中で実際に動いているのは 1.5 % である。量としては、180Mbit = 22.5Mbyte である。

さて 22.5Mbyte という遺伝子の量を大きいと見るか小さいと見るか、みなさんはどう思われるだろうか。1 行 22.5 文字のプログラムを百万行書いたならば、プログラムを記述するテキストデータは 1 文字 1byte なので丁度 22.5Mbyte となる。さて、百万行のプログラムコードを書いたことのあるプログラマというのはそういない。私が若かったころ、1ヶ月に 1 万行くらいのコーディングをした。一週間かかって 1 万行を書き、3 週間デバッグするというペースで、考えて、1 年かけるとおおよそ 10 万行書ける。まあ、12 万行書くのは、他の仕事などが入ったりして、なかなかそこまでは行かないときが多かった。そして 10 年かければおおよそ百万行のプログラムが出来上がる。そう考えると、その気になれば書けないことはない量とも言えるが、十年間同じプログラムを作り続けるプログラマもまれであるから、このヒトゲノムの量は尋常ではないと言える。

では百万行クラスのコード量²²のプログラムとは具体的にどんなものがあるのだろうか。実際にコードを数えたわけではないが、プログラマとしての感覚からすると、マルチウィンドウが備わったちよつとしたオペレーティング・システムである。Macintosh や Linux のコア部分、あるいは、Firefox などのブラウザ・ソフトウェアなどかと思われる。主にグラフィカルなユーザインタフェースが備わったものはコード量が増える傾向にある。ウィンドウが重なりあった様子を、内部データ構造と整合性を保ちながらマニピュレーションする仕組みなどにコードがかさむ。グラフィカルなユーザインタフェースはコンピュータに多大な負担をかけるために、高速化のチューニングを行うとコードはすぐさま二倍三倍となる。

ヒトの顔にディスプレイがありマルチウィンドウ・インタフェースが表示されるわけではない、一方で、コンピュータはヒトゲノムと同じコード量を使いながら、ヒトがするような思考が出来る訳ではない。だがひよつとするとヒト

²²プログラムの量のことをコード量と言うことがある。

のもつ知能の実現には、せいぜいオペレーティング・システム程度のコード量で十分なのかもしれない。

ここでももう一つの事実がある。DNAの大きさや遺伝子（つまり実際にコードとして機能している部分）の量は見かけの生物の複雑さとは相関がないと言われている。たとえば、ヒトは線虫の1.5倍弱ほどの遺伝子量であるという。ヒトが少ないのか、線虫が多いのか。プログラムも似たようなことがある。コード量の大きいプログラムが必ずしも高機能であるとは言えない。コード量は様々な方法で減らすことが出来る。その一つの方法が制御用言語を作ることだ。巨大なアプリケーションを開発するとき、与えられた仕様を直接実現するプログラムをいきなり書き出すことはしない。まず、その仕様全体を眺め、何が求められているかはもちろんのこと、様々な事項の間で共通事項はないか、同一のパラダイムとして捉えられる事項はないかを洗い出す。そして、まず最初にやることは、アプリケーションを記述するのに適したスクリプト言語を作ることだ。まずアプリケーションが求めている機能は必要ない、そのような機能は思い切ってスクリプト言語からは省く。また、アプリケーションのパラダイムに適した言語パラダイムを選ぶ。オートマトンか、データフローか、関数型か、マークアップか、スクリプト言語は複数あってもよい。既存の言語と一部を新作スクリプト言語とすることもあり得る。既存のプログラミング言語でアプリケーションの全てを記述した場合と、既存の言語ではスクリプト言語のインタプリタを記述し、そのスクリプト言語で目的のアプリケーションを記述した場合では、後者の方がコード量を削減することが出来るほか、プログラム全体の見通しが良くなり、開発時間の削減、管理効率の向上など様々なメリットが見込めるのである。

世界初のマイクロプロセッサ Intel 4004 の開発秘話に同様な例を見いだせる。当時の計算機メーカーのビジコン社は、新しいプログラマブルな電卓の開発のため、インテルに10個のLSIチップの制作を依頼した。しかし設計コストや納期を考えると10個ものチップを制作する余裕はなかった。そこでインテル側から提案があったのは、機能を厳選し、小型化したマイクロプロセッサを1つだけ開発し、10個のLSIチップに課せられる様々な機能をすべてこのマイクロプロセッサ上で動くプログラムで実現する方法が提案された。そして誕生したのが Intel 4004 である。当初はビジコン社への専売であったが、後にマイクロプ

ロセッサの重要性に気づいたインテルはビジコン社から販売権を買い取り 1971 年に Intel 4004 として販売された。この場合、ハードウェアの実現の話であるが、マイクロプロセッサのプログラムが上述のスクリプト言語の役割を果たしている。このような階層化構造がコードの削減、見通しの良さを生み出す。

細胞というシステムも同じように考えることが出来るだろう。たとえば、ヒトの知能について考えてみよう。ヒトの思考を直接作り出しているのはニューラル・ネットワークである。ニューロンやニューロン・シナプスには活動電位を伝達する機能をもっている。ニューラル・ネットワークのシミュレーションなどで分かってきていることとして、抑制型のシナプスと活動型のシナプスがあり、さらに、頻繁に信号を受けるとそれぞれの型によって増強、抑制が働く。

DNA 上には確かに、これらの機能を実現するのに必要なタンパク質や RNA のコードが記載されているはずだ。あるいは、ニューラル・ネットワークが初期状態としてどの細胞群とどの細胞群が結合すべきかという情報に当たるタンパク質を生産するプログラムも DNA 上には記載されているかもしれない。しかし、ニューラル・ネットワークがどういう情報を採取し蓄え、出力するのかという情報は DNA に直接的にコーディングされているわけではない。しかしヒトは明らかにニューラル・ネットワークを利用し、情報を採取し蓄え、出力する。

遺伝子が生産するタンパク質や RNA は細胞という環境あるいは生物個体という環境へ放出され、環境にすでにあるモノを化学反応によって結びつけ、変換し、操作する。遺伝子はいわばその結合、変換、操作が巧みなのだ。DNA が変化すれば、その結合、変換、操作は変化を受ける。従って異なったアウトプットが得られる。DNA は環境中のモノを利用して様々な操作をする。環境中のモノを CPU として見なすならば、DNA はその上で動くスクリプトと言えよう。しかし一方で、DNA が変わらなくとも、環境中のモノが変わればアウトプットが変わる。深海の未知なる生物をつり上げてきて水族館で育てることは至難の業を極める。多くは育たず死亡してしまう。つまり生物にとって環境がいかに大切か分かる。同じ遺伝子もっていても異なる環境に暮らす生物は徐々に形態が異なってくる。環境中にあるモノが変化するとアウトプットが変わる良い例である。環境中に投入されるモノは遺伝子による結合、変換、操作を受け、結果を出力する。こう見れば環境に投入されるモノは遺伝子という既存プログラ

ミング言語上で動くスクリプトと言えよう。

少なくとも、細胞内の環境、あるいは個体内の環境は遺伝子によってよくコントロールされているように見える。だが、コントロールは一方方向ではない。遺伝子と環境内のモノの関係は循環関係になっている。生命の形として、相互に一方が片方のスクリプトとなって動いているという形が見えてくる。

また、このような循環は進化において重要である。この循環によって進化に対して抑制と促進の両方の圧がかかる。この相互バランスによって進化が暴走し、生命が破綻しないように制御されているのではないか。環境というスクリプト、DNA というスクリプトが、進化の過程で、より洗練されて行けば遺伝子のコード量は必ずしも大きくなって行くとは限らない、むしろ逆に小さくなることもあり得るかもしれない。

第3章 機械は生命化する

3.1 人工知能

チャールズ・バベッジは、自身の開発していた機械式計算機に、計算をする機械以上の意味を見いだしていただろうか。ペンシルベニア大学における真空管コンピュータの開発チームをはじめとしたその当時のコンピュータ開発者たちは新たな時代を感じ始めていた。コンピュータがプログラムを自分に読み込み、プログラム自身を書き換え、計算を進めるようになっていた。コンピュータの対象とするものが少しだけ「計算」から「プログラム」へ幅を広げ始めたのである。「計算機」を超えたなにかが実現可能になってくるだろう、という期待が膨らんだ。

コンピュータに計算以外のなにをさせることが出来るか、計算機学者たちは考え始めた。一つの考え方は、人工知能に代表されるような、人間の思考を代行する、という考え方だ。この考え方の一方に、思考そのものというよりは人間の思考の入出力の代行者として、思考の道具としての機能の追求がある。まずは人工知能に目を向けよう。人工知能のレベルも様々な考え方があるだろう。人間の思考に完全に取って代わられるような”完全な”人工知能という考え方もあれば、現在だったら笑顔を認識してそこにピントを合わせることでできるカメラ、といったように、様々な有用な知能を模擬することによって、あらたに思考の道具とすることも考えられる。

丁度、人工知能研究が盛んだった1980年代のころだったろうか。かつてこんな逸話を読んだことがある。世界中の天才的な計算機学者や脳科学者があつめられ、世界最高の人工知能マシンを作るプロジェクトが始まった。幾年かの歳月を経てその人工知能は完成し、多くのメディアが集まり、世界が注目するか、哲学的最終問題が入力された。「神は存在するか。」人工知能はおもむろに

計算を開始し、やがて答えを出した。「いまこそ神は存在せり。」

人工知能学会などの立場を見ると、人工知能は強い AI と弱い AI という二つの立場に分けることが出来るとされる。人工知能学会のホームページ¹では、

人工知能 (AI) とは知能のある機械のことです。しかし、実際の AI の研究ではこのような機械を作る研究は行われていません。AI は、本当に知能のある機械である強い AI と、知能があるようにも見える機械、つまり、人間の知的な活動の一部と同じようなことをする弱い AI とがあります。AI 研究のほとんどはこの弱い AI で、図のような研究分野があります。

と書かれている。情報科学は実際に動作するものを作ろうとするので、作れないものは作れない、わけである。しかし、別の考え方として、作れなかったとしても、「知能とはなにか」という問いは成立するはずだ。上記の立場に立てば、「強い AI に相当する、あるいは強い AI が模倣する対象としての知能とはなにか」、同様に、「弱い AI に相当する、あるいは弱い AI が模倣する対象としての知能とはなにか」という問いが成立するはずだ。前者の問いを言い換えると「人間の知能とはなにか」という問いになる。

人工知能の黎明期である 1950 年代はこのような人工知能の住み分けは出来ていなかった。知能が人間的であるか否かを判別する、注目すべき主張として、アラン・チューリングのチューリングテストという考え方がある [6]。チューリング・テストでは、2 台のディスプレイとキーボードを備えた端末を用意する。一方の端末にはテストする人 (テスター)、他方には、テスターの質問に答える解答者が座り、会話を行う。解答者の方をテストする機械に置き換えた場合に、テスターがその相手が機械なのか人間なのか見分けがつかないとき、その機械はある種の人間的知能をもっていると断定してよいだろうという主張である。このテストの目的からすると、解答者が被験者もしくは検査対象機械である。

これに対する反論は様々あるが、その中でも良く取り上げられる有名なものが「中国人の部屋」というサールの反論である [7]。これは、テスターが中国人であり、一方、被験者の部分に中国語を全く理解することのできない英国人と、中国語の文章をその解答へ変換するための用意周到なマニュアルを用意するも

¹<http://www.ai-gakkai.or.jp/>

のである。もちろんマニュアルは、被験者の母国語で書かれている。被験者はテスターから送られてくる中国語の質問をこのマニュアルに基づき変換しテスターへ送る。テスターはかえってくる答えが用意周到であれば、応答しているのは人間だと思うが、果たして被験者英国人は中国語を理解していないではないか、というものである。

「中国人の部屋」に対する反論として「システム論」がある。私が日本語を理解しているが、私の臓器やニューロンやシナプスの一つ一つが日本語を理解している、というような言い方はしない。私を構成するすべての部品が一体となった私が日本語を理解しているのである。従って、英国人とマニュアルを一体として考えなければならない。英国人が中国語を理解していないからといって強い AI に対する、あるいはチューリングテストに対する批判にはなっていないというものである。

これに対してサールは、データの入出力作業やその他マニュアルを自分の頭の中にすべて押し込めたとしても、被験者は中国語を理解したとは言えないと言って再批判している。これに対して、Levesque らによる計算論的な反論がある。中国語の問題を複雑な足し算の問題に置き換え、足し算のやり方を知っているテスターと知らない被験者とマニュアルの組み合わせとする。被験者が足し算の問題を全部頭に押し込めたとしたら、その被験者は足し算のやり方を理解していないと言うことが出来るだろうか、というものである。結局我々が小学校で習う足し算のやり方は、先生に与えられた教科書というマニュアルを頭に叩き込むことではないか。

しかしこの Levesque の反論 [8] はなにか弱いところがある。それは、中国語を理解するという、いまの人工知能をもってしても難しい問題を、足し算という計算機にとってはかねてから得意な問題、あるいは弱い AI の問題へ置き換えてしまったことだ。たしかに我々は俗に「計算機は足し算を理解している」というような言い方もする。しかしそれは、言葉のレトリックに過ぎないという主張も成り立ち得る。

サールの再批判に対する、強い AI は強い AI の問題での、再々批判としては、服部裕幸が述べている、クオリアの問題が私にはしっくりくる。中国語のマニュアルを全部頭に叩き入れた被験者英国人は、中国語を理解しないまま、つまり、中国語を理解するというクオリアをもたないまま、テスターとコミュニ

ケーションをやっているということになるというものだ。つまり、サールはクオリア不要論を主張しているということになるというのである。これはかえって、クオリア不在の哲学的ゾンビをコンピュータは実現しうる可能性と、その有用性を示唆してしまっている。

服部も論文中でしばしば主張しているように、サールの批判は当時の黎明期の技術に対する批判で、現在の様々なシステムには当てはまらない点もある、という主張も可能である。たとえば、自動翻訳の研究はいわば「中国人の部屋」実現計画である。そしてそれらの中には、それなりに有用で便利なものも登場していることは事実である。AI というものにこだわる必要はない。インターネットに沢山存在する様々なショッピング・サイトでは商品の購入、カートへの投入、決済という一連の手続きが自動化されている。その様々な状況で確認メールが届いたり、お礼メールが届く。これらのメールとやり取りをすることによって、手続きが進んで行くことになる。そのメールの送り主は人間なのか、それともコンピュータなのか、時々混乱してしまうわけで、これもまさしく「中国人の部屋」である。こう話すとき情報科学者ですらも「それはショッピングサイトという限定的なシチュエーションだから可能なのだ」と反論するだろう。しかし人間も限定的なシチュエーションでしか行動できないと主張することも可能である。人間同士のコミュニケーションにおいても、自分の理解可能範囲へいかに状況を落とし込んで行くか、ということが重要なファクターとなっているといえる。一方、自分の理解可能な範囲から外れてしまうと、会話を放棄するか、あるいは ELIZA のように聞き役に徹するか、しかなくなる。最も危機的な状況では、たとえば、沈没船と一緒に沈んだ溺死者を捜索すると、しばしば船の欄干に堅くしがみついたまま溺死している人がいると聞く。自分が遭遇したことの無い危機的な状況に陥ったとき、人が頼りにするのは、それが先入観であっても、自分のもっている過去の知識であると言える。結局これはフレーム問題である。

道具としてのコンピュータのもつメリットは、ショッピングサイトのように、購入者が予期せぬ行動をとらないように、予測可能な範囲へ追い込むことができるということ、と考えることも出来る。顧客相手の様々な商売では、顧客のクレームへの対応はコスト増大させるので、いかに顧客のクレームを少なくするかということは重要課題である。そのため多くのビジネスではフレーム問題

を予測可能な範疇へ押し込める方法が真剣に検討される。この観点からすると、人的応対をあえてインターネットショッピングサイトへ移行するという選択肢もおおいにあり得る。コンピュータが人間的な知能をもたないことが逆にメリットなのである。このような黎明期の批判合戦に、ある意味情報科学者は飽きたのだらう。それよりも、弱いAIの研究や、道具としてのコンピュータの道の方が、実際に動作する快感が味わえるし、社会的にも有用である。研究しがいがある。人間の知能の核心的な部分は哲学者へ任せてしまった。

情報科学者どういう道を選択したかについては次節以降にあらためるとして、ここではもう少し哲学的な議論を検討しよう。強いAIが人の知能を目指すものである以上、心の哲学と密接な関係をもつ。「中国人の部屋」の問題で服部はクオリアの問題を提示した。心の哲学としての関係を示唆するものである。クオリアの問題に関連して、ここで、物理主義的な立場を取ったとしても、クオリアは消えないことを示唆する思考実験を提示しておこう。

クオリアの問題で良く取り上げる視覚に関する思考実験を設定しよう。名付けて「超未来MRI」とでもしよう。物理主義を以下のように設定する。視覚が与える心的影響を私秘的なものも含めて、物理量としてとりだせるという立場である。被験者からとりだされた物理量は被験者がこれは違う感覚だという場合、物理量も異なる値を示す。被験者がこれと同じ感覚だと思う場合は物理量も同じ値を示すものとする。このような対応関係が得られるだけの化学的知識が得られたとする。また、見せる色についても、現在あるような三原色ではたりない。おおよそ、スペクトルやテクスチャに関する情報その他すべてが人間の認識可能なまでの細部の情報まで再現できるディスプレイ装置が開発されたとする。ここで言う物理量は実数で表されるだけでなく、ベクトル、関数、いかなる数学的な量であってもかまわないが、あくまでも数学的の量として記述できるということである。

このような仮定に立った時、被験者AとBの二人の認識の違いについて実験をする。被験者Aに上述の高精細ディスプレイを眺めてもらう。同時にプローブを差し込み色覚の情報をとりだす。ディスプレイを調節すると、色覚情報に変化することが読み取れるだろう。被験者Aの読み取られた色覚情報を被験者Aに見せても、被験者Aはそのことを理解できるかどうか分からない。それは高度な化学的知識がないと理解できないかもしれない。しかし、その情報を保

存しておき、ディスプレイにフィードバックすることによって、同じ色覚を被験者 A に再現することは可能はずだ。同じ装置を被験者 B にも装着する。同様にとりだされた色覚情報をディスプレイにフィードバックすれば、同じ色覚を被験者 B に再現することが可能である。

もう一つ、得られる知見として、被験者 A から得られる色覚情報と被験者 B から得られる色覚情報が同じか違うかを比較することが出来るはずだ。もし一致させることに成功すれば、A と B は同じ感覚になっていると判断できる。ここまで、被験者 A にある色をディスプレイに表示し、そのときの色覚情報をとりだす。これをリファレンスとして、被験者 B の色覚情報がリファレンスと一致するようにディスプレイを調節することが可能かということである。おそらく、全ての場合において一致させることが出来るという事はあり得ないだろう。ある程度譲歩できても、一致させることが出来る場合もあるが、一致させることが出来ない場合もある、というところまでだ。一致させることが出来ない、つまり、被験者 A の感じていることを被験者 B には再現できない、つまり被験者 A にとっての私秘的な現象がある、と言える。

もともと、このような超未来的な思考実験でなくても類似したことは再現できる。かつて所属していた研究室で、マルチメディア通信なるものを研究していた。音声や画像だけではなく、こちらの環境全体を先方に送って、同じ環境で話ができるシステムという話があった。話を簡単にするために、部屋の温度を送信するという事だけを考えてみよう。両方を同じ温度にして「あついね」「さむいね」という感覚を共有するという事を考える。一見簡単そうである。しかしここで、「涼しいね」とか「暖かいね」という感覚を再現できるだろうか。もう少し状況設定を具体的にしよう。話者 A は札幌在住であり、いまは真冬で外は猛吹雪、零下 10 度。時刻は 15 時、しかしこんな悪天候にも係わらず、どうしても必要な用事から帰ってきたところだった。家へ入って早々、着替えもままならぬ状態で、マレーシア在住の話者 B からマルチメディア電話がかかってきた。話者 A はなんとか簡単な着替えをすまし、端末の前に座る。話者 B は友人たちとのテニスの交流試合が終わり家に帰ってきて、冷房の効いた部屋で、友人にマルチメディア電話を掛けたところだった。話者 B は通信が接続されるなり「ああ、この部屋は涼しい。」と言った。さてこの状況を、話者 A の環境に再現できるだろうか。「涼しい」という感覚は、温度が高い環境で十分に汗腺

が開ききったところで、温度の低い環境へ移り、体の温度が下がって行く状態のときに感じる感覚である。従って、既に冷えきった体の話者 A には体験することが出来ない感覚である。

もし、無理矢理話者 A に「涼しい」という感覚を経験させようと思ったら、しばらくサウナにでも入ってもらい、そのあと、タオル一枚と牛乳瓶片手に、扇風機にあたりながら、マルチメディア電話に向かってもらうしかない。つまり感覚には、時間経過が係わる場合がおおい。ある種のことを体験することによってもち得るのである。その時間経過のスケールは様々である。「涼しい」と感じるくらいの程度であれば十数分サウナに入ってもらえればいいたろう。だが、生涯、あるいは進化論的な時間経過が必要なものもあり得る。

この思考実験は、マス・ネーゲルが論文「コウモリであるとはどのようなことか？」での議論に近い。コウモリの超音波で感じる空間認識を人間が感じる事が出来ない、ということである。だが、一方で、同じ体験に帰属しているもの同士であれば、理解し合うことも可能性としては残る。また、「マリーの部屋」の問題の考え方として、事実に関する知識である命題知と、実際に体験して得られる知識、技能知は異なるという考え方にも通じる。

マリーの部屋も色覚に関する思考実験である。聡明な科学者であるマリーは色覚に関する研究を行っている。なんらかの事情によりあるときから、白黒の部屋に閉じこもり、外界の情報も白黒テレビを通してのみ接するようになった。マリーの研究は、熟したトマトや晴れた空を見るときに感じる色彩についての情報を物理的、神経生理学的に捉えることである。このマリーが白黒の部屋から開放され、あるいは、テレビがカラーになったときになにがおこるだろうか。なにか新しい知識を得るだろうか。というものだ。

これに対する私の答えはこうだ。マリーが白黒の部屋で研究していたものは、上述超未来 MRI の思考実験の、被験者 A からプローブされる色覚情報と同じである。マリーは自分の色覚情報を研究していた訳ではない。また、白黒の部屋に閉じこもってしまっているので、自分の色覚情報を研究しようにも研究できない。したがって、得られる情報は全て他者の情報である。したがってマリーは被験者 B の立場なのである。マリーが白黒の部屋から開放されたという状況は、被験者 A の色覚情報をリファレンスとして、被験者 B の色覚情報を調節することと等価である。結果は上述通りである。つまり、マリーはなにも感じる

ことは出来ず、白黒の部屋で得た知識は知識の域を脱することが出来ない可能性は歴然として残る。

この思考実験をより現実的な、しかしあり得ない実験に置き換えると、二原色色覚者がある日三原色色覚者になったらどうなるか、あるいはその逆、または、我々のほとんどであろう三原色色覚者が四原色色覚者にある日突然になったらどうなるか、と比較できる。あるいはこんな仮定もあり得るかもしれない。幾何学者は、多様体理論によって4次元、5次元といった高次元の立体物の性質を議論する。さて、かれらの感覚器が突然変異をおこし、超紐理論で言うところの多次元空間が実際に知覚できるようになったらどうなるか。マリーは上述の文章から「なんらかの事情により」ということなので、先天的に白黒しか見えないという状態ではなかったようだ。カラーが見えていた状態から白黒の部屋へ入り、そして再びカラーの世界へ出てきたということである。つまり、カラーを見たらどういう状況かすでに経験して知っていたということにならないか。マリーが先天的とするならば、先天的な失明者が手術などで見る事が出来るようになったならば、それまで触覚によって認識していた、立方体や球などの形が、まだ触れる前に目で認識可能か、という、これはモリヌークス問題の変形ととれる。ティドロがこれに対し、回復した失明者の目は不完全であり、認識そのものがままたず、時間的経験を経てようやく「見る」という行為が可能となることを指摘した。

たいしたことではないが、私は生まれつきの上斜位だった。ロンパリということである。急激な視力低下を招いたので高校時代に手術をしこれを矯正した。しかし矯正当時静止した立体物を両目の視差だけで認識することは出来なかった。つまり静止物立体認識のクオリアは無かった。当然、目は見えていて、それまでも、動く物体の動的視差によって立体物や距離感を認識することも出来ていたので、立体とか距離とかのクオリアや知識はあったわけだ。矯正からすでに三十年たつが、静止物体の立体認識は未だに不得意である。だが、矯正当時よりはまして、静止立体感のクオリアというのも多少は体験できるようにはなっている。

マリーの部屋は、マリーがカラーの国へ復帰した瞬間を議論しているが、おそらく復帰直後は全くなが起きたか認識できないだろうが、経験を積み重ねて行くにつれて徐々に経験知を理解できるようになる。あるいは徐々にクオリ

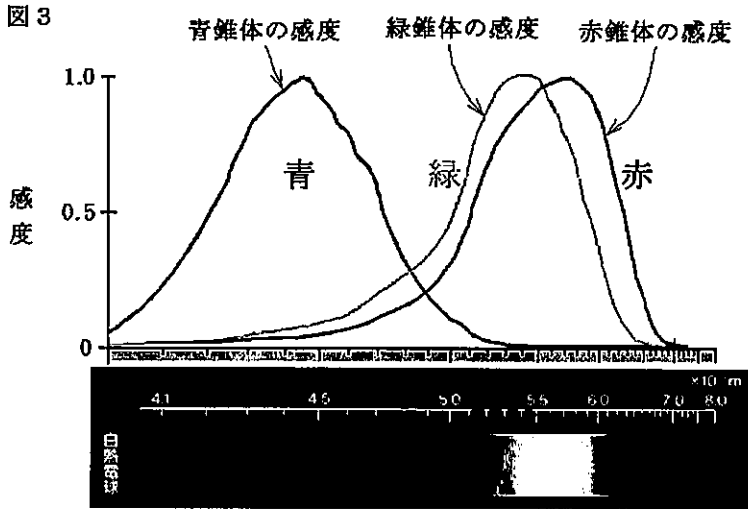


図 3.1: 色

アを獲得できるようになってくる。そういう経験は誰でもするだろう。お酒は二十歳からだが、そのあと酒の味をだんだん覚えて行くということは、酒好きにとっては「あるある」といったところだろう。

コンピュータのディスプレイは Red, Green, Blue の三原色によって自然な色を再現している。我々の多くは三原色色覚者なのでこれで満足である。だが、この問題は思うほど単純ではない。色は光の波長に対応している。可視光の波長は 400nm から 800nm 程度までの間の波長に分布している。波長が短くなるほど青、長くなるほど赤色となる。並べると赤橙黄緑青藍紫と呼ばれる虹色になる。残念ながら白黒であるが、図 3.1 下段がそれであり、光のスペクトルである。ここでまず第一の疑問は、たとえば、450nm の光と 550nm の光を混ぜ合わせても、物理的には、500nm の光の波形になるはずがない。しかし色覚の混色は、まさにそういうことだ。どうしてだろうか。

眼底には、4種類の視細胞で埋め尽くされている。そのうち3つは色を認識するもので錐体細胞と呼ばれるものである。残りの一つは明暗を認識するもので桿体細胞と呼ばれるものである。それぞれの錐体細胞はそれぞれの感光特性

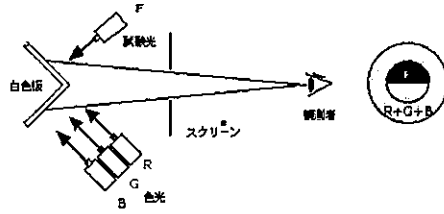
をもっている。それが図 3.1 の上段のグラフである。それぞれの錐体細胞がどの程度興奮するかによって三原色の混合の割合が決まると考えられる。

だが事態はそんなに自明なことではない。図 3.1 を見ると、410nm くらいの光になると、青色の錐体細胞だけが興奮する領域といえそうである。しかし、緑色の錐体細胞だけ、あるいは赤色の錐体細胞だけが興奮する領域というのはほんのわずかしきが無い。あるいは皆無と言って良い。それで、赤とか緑という色は感じる事が出来ないではないか。これが第二の疑問である。この問題を解決するために、各錐体細胞の正確な特性曲線を割出す等色実験がある (図 3.2(a))。これは波長が可変できる短波長の光源 F と、可視光からとりあえず選んだ三つの色の光源 R,G,B の混合とを比較する実験である。R,G,B の波長は固定であるが、強さをそれぞれ独立に調整できる。図では、F を上側のスクリーンに、R,G,B の混合色を画面の下側のスクリーンに照射したものを被験者に見てもらい、上と下の見分けがつかなくなるように R,G,B の光を調整してもらう。

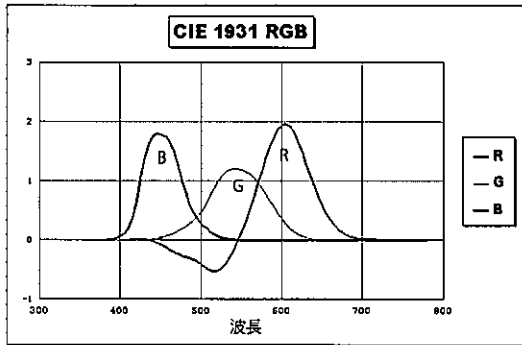
F の波長をいろいろ変えて実験すると、おおよその場合上下を合わせることが出来る。しかしある波長の領域においては、R,G,B の色の調整では作り出せない領域が生じる。そこで、R の光源をもう一つ用意し、それを F に重ねると、上下を一緒にすることが出来る。F 側に R を足したということは、F を変えずに、R の強さをマイナスにしたということと等価であると解釈できる。このようにして F の波長と、R,G,B の強さの関係をグラフにすると、図 3.3(b) となる。このようにマイナスの部分が出現するのは錐体細胞の特性がかぶっているからである。しかもここでは適当に選んだ R,G,B であるので錐体細胞の特性には至っていない。そこで錐体細胞そのものの特性を、記号を変えて X,Y,Z としよう。そして、X,Y,Z と R,G,B には以下のような線形結合の関係があると仮定する。

$$\begin{aligned} X &= a_{11}R + a_{12}G + a_{13}B \\ Y &= a_{21}R + a_{22}G + a_{23}B \\ Z &= a_{31}R + a_{32}G + a_{33}B \end{aligned} \quad (3.1)$$

さらには、錐体細胞の特性は負にはならないということで、

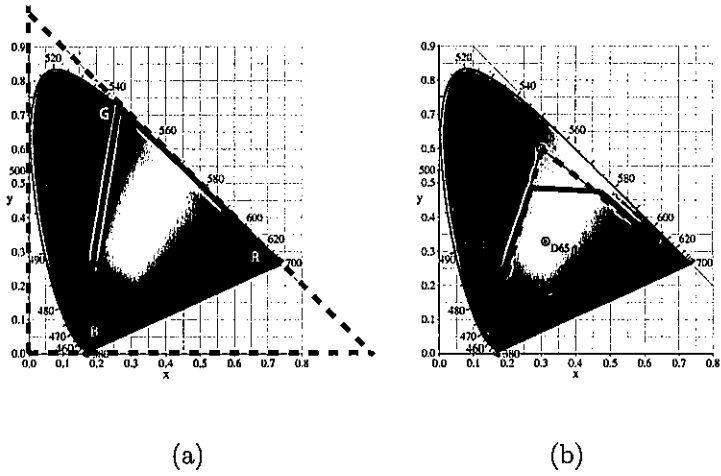


(a)



(b)

図 3.2: 三原色を特定する実験



図中に、色の領域を示すために R,G,B の記号が記されている。

図 3.3: 色彩空間

$$X \geq 0, \quad Y \geq 0, \quad Z \geq 0 \quad (3.2)$$

という条件を付ける。R,G,Bが図3.2(b)のグラフを実現し、上記非負の条件に当てはまるよう a_{ij} を決めると、等色実験から、X,Y,Zを求める計算式が得られる。国際標準規格CIE(Commission internationale de l'clairage)表色系においては、RGBのそれぞれの波長を、R=700nm,G=546.1nm,B=435.8nmととり、以下のように定めている。

$$\begin{aligned} X &= 2.7689 R + 1.7517 G + 1.1302 B \\ Y &= \quad \quad R + 4.5907 G + 0.0601 B \\ Z &= \quad \quad \quad 0.0565 G + 5.5943 B \end{aligned} \quad (3.3)$$

このX,Y,Zを平面上に色相として展開するため、全体の光の強さを $X+Y+Z$ を一定にし色相のみをとりだす方法として、

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z} \quad (3.4)$$

とし、 x, y を平面に展開する。

$$x + y + z = 1 \quad (3.5)$$

が成り立ち、 $x \geq 0, y \geq 0$ である。 x, y が最大になるのは、 $z = 0$ のときであり、 $x + y = 1$ という、右下がりの直線になる。このようにして色をプロットすると、図3.3(a)のようになる。破線で囲われた三角形の領域は、 x, y, z の範囲XYZで表すことのできる色の範囲である。光のスペクトル、各波長の色はこの図の上では上部に湾曲して出来ている円弧の部分の色に相当する。人間の目が知覚できる色は、この半円形の内側となる。破線の三角形の内側、半円形の外側は、XYZまたは xyz の値としては存在するが、実際には存在しない色である。特に破線の三角形の頂点は、錐体細胞が個別に一つだけ反応した場合の値ということになる。あくまでも仮想的な話で、そのような色は人間は知覚することができない。この図の中に、上記R=700nm,G=546.1nm,B=435.8nmをプロットしてみると、太い線の三角形RGBのようになる。この三角形のRGBは

いわゆる等色実験で RGB 光源によって作り出すことのできる色の範囲である。この RGB 光源の作り出すことのできない範囲として、線 GB の外側に大きく緑の領域がある。ここが、R の信号を F に足すことで表現される領域である。

しかし、等色実験に使われた RGB 各光源は実験用の高精度のものである。おおよそ家庭用テレビやその他多くのカラーディスプレイはこのような高精度な発光素子を使うにはあまりにも大掛かりでコストに合わない。そこで、実用性を考慮した sRGB などの規格が存在する。図 3.3(b) の破線は広く普及している表色系である sRGB の範囲であり、実線は印刷に使われる代表的なインキの発色範囲である。

このように人間の色覚は複雑な構造をもっている。また、実は三原色では表すことのできない色領域というものもあることが分かる。さらにはこの等色実験は民族差があると言われている。また、個人差のばらつきも大きいと言われている。また色は配色、光の増減、様々な動的な環境変化の中で大きく感じ方が変化する。そういった複雑な感覚の機構はいまだ解明されているわけではない。

昨今、都会人たちは、テレビやコンピュータ・ディスプレイ、スマートフォンといった sRGB 系列の様々な機器を絶え間なく使うようになった。さらには持続性社会、CO₂ 排出量削減と称し、また色味が変えられるといった宣伝も相まって LED 照明が普及し始めている。だが、これらに色領域が実際に人間が感じられる色領域から比べると、如何に貧弱かということも分かるだろう。このような急速な人工的な発色の普及は、色の多様性を破壊して行く傾向を生み出すのではないだろうか。通販サイトにおいて、RGB ディスプレイに映し出されたときに発色のわるい商品は売りにくくなり、そういった商品は自然と姿を消して行くのではないか。人間の目は人工色にならされた人が多くなり、そういった色を渴望するようになり、逆に外れる色味のクオリアが消えて行く。

言ってみれば、現在世界中で「マリーの部屋大実験」が進行中なのである。テクノロジーは世界中の都会人たちを人工的な RGB マリーの部屋に閉じ込めている。マリーはいつしかその部屋から開放されたが、都会人たちは開放されるのだろうか。サールは「中国人の部屋」を拡張し「中国人のジム」という思考実験をしている。英語のみを話す多くの人が居るホールを想定し中国語解釈を行うのである。サールはコネクショニストへの批判としてこのような思考実験を持ち出したが、単なる人工知能研究者への批判というよりも、現代社会の

構造的な問題への批判にも受け取れる。考えてみればこれはマンハッタン計画である。誰も原爆を作っているということを知らされず部分的なマニュアルだけで行動する。全体は原爆を理解しているか。「マリーの部屋大実験」も同じである。このようなRGB化がなにを意味するのか、誰も理解していない。サールの批判はこのように見ると的を得ていないとも言えない。

人工知能における黎明期の批判合戦に、ある意味情報科学者は飽きたのだろう。それよりも、弱いAIの研究や、道具としてのコンピュータの道の方が、実際に動作する快感が味わえるし、社会的にも有用である。研究しがいがある。人間の知能の核心的な部分は哲学者へ任せてしまった…。

3.2 道具としてのコンピュータ

人間の作るものはそもそも道具である、と考えるならば「道具としてのコンピュータ」という言い方はトートロジーである。しかし、コンピュータの歴史のある時期、人工知能という、もしかすると人間と同等、あるいはそれ以上の知性をもち得る機械が登場するかもしれないという時代があったためにこのような言説が持ち場を得るのだろう。

コンピュータの道具としての役割とはなんだろうか。バベッジの時代からノイマン前夜くらいまではコンピュータは、それまで手で行われた数値計算を代行する機械であった。ノイマン型アーキテクチャ、さらにはマイクロプロセッサと技術革新が続くにつれて、その役割が劇的に変わった。と言っても、その延長線上の帰結から見ている我々からすると実感がないだろう。身の回りにコンピュータを使ったシステムが氾濫する現代、特に若い世代にとってみれば、コンピュータが道具かどうか以前にはじめからそのような形でそこにある。

まずPCとよばれる、パーソナル・コンピュータからいこう。「パーソナル」と呼ばれるからには、つまり「パーソナル」でないコンピュータがある。現代でいうならばスーパーコンピュータであったり、昔のタイム・シェアリング型汎用コンピュータがパーソナルでないコンピュータの代表である。これらは、まず購入価格が高価であり、個人が買えるようなものではない。ある機関が購入し、設置し、管理する。多くのユーザはこの機関にユーザ登録をし、様々な計算や処理を依頼する。依頼された処理は、いわゆる「パーソナル」でないコンピュー

タへ送り込まれ、仕上がったものから返却される。依頼から結果の返却までは現代であれば、インターネット経由で自動化されている。昔であれば、パンチカードをもって行き、プリントアウトをもらってくる、など、である。「パーソナル」でないコンピュータとは、いわば計算の工場である。

これに対して現在のパーソナル・コンピュータはディスプレイとキーボード、マウスの三点セットがそろったコンピュータである。ディスプレイはウィンドウシステムを備えている。初期のパーソナル・コンピュータにおいて、一番最初に入出力装置として登場したのは、タイプライターを模したキーボードであった。そのときの出力装置はプリンタであった。その名残が、文字を表示するときのコンピュータ言語の命令 `print` に残っている。キーボードとプリンタはまさにタイプライターである。デニス・リッチーとケン・トンプソンはキーボードとタイプライターを備えた PDP-7 上で UNIX を開発した。当時、おそらくコンピュータにキーボードを付けたということでも入力装置としては進化であったろう。しかし同時代、もっと革新的なインタフェースの開発が行われていた。マウスの開発は 1960 年代エンゲルバートによるものだった。エンゲルバートは、有名な論文、“Augmenting human intellect: A conceptual framework” [9] 『人間の知性を増強する：概念的枠組み』において人間と機械のコラボレーションによる新しい形の思考作業の形態を提案している (図 3.4)。マウスとディスプレイは、このような人間の思考を助ける道具としてのコンピュータという発想から生まれてきた。これだけではない。この論文に ARPA より予算がつき、エンゲルバートはスタンフォード大学 SRI インターナショナル内に、Augmentation Research Center (ARC) が開設された。ここで様々な先進的なユーザインタフェースのアイデアが生まれる。たとえば、ビットマップ・ディスプレイやハイパーテキスト、グループウェアといった概念も、この研究所から生まれてきた。

このような思考としてのユーザ・インタフェースのマイルストーンが Alto である。Alto は後にスティーブ・ジョブズに影響を与え、Lisa とそれに続く Macintosh の開発の着想の元となった。ゼロックスのパロアルト研究所 (PARC) で制作された Alto の中心的コンセプトは `What You See Is What You Get = WYSIWYG` (ウィズィウィグと読む) というものだった。Alto はグラフィカル・ユーザ・インタフェースを備え、今で言うワードプロセッサが動き、画面に、印刷出力と同じレイアウトの文書が表示され、編集できる。編集された文章は、

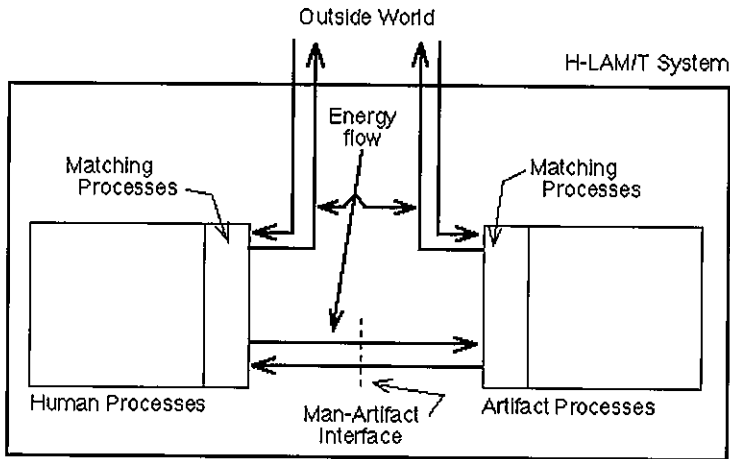


図 3.4: [9] The Two Sides of the H-LAM/T System

ネットワークで接続されたプリンターで出力される。1970年代半ばにはウィンドウ・システム、アイコン付きパレット、あるいは様々な機器をネットワークで接続するといった、現代のコンピュータ、あるいはスマートフォンなどが備えている基本的な概念が確立されていた。図 3.5 は Alto の写真である。ディスプレイが縦長なのは書類や論文のページは縦長であるところに由来しているからだ。かつて Macintosh のディスプレイも多種多様なものがあつた。あまり知られていないことだが、MacPlus や SE など小型かつディスプレイ一体型は、ケースに入れて持ち運びやすいように作られていた。世界初のモバイル・コンピューティングと言っても良いかもしれない。また、Iicx などの時代には縦型ディスプレイも存在した。コンピュータのディスプレイはテレビや映像モニタではなく、思考の道具にとって最適な形があるはずだ、というのも一つのユーザ・インタフェースへのこだわりである。

だが、面白いことに現代は横長の書類が増えている。それはコンピュータで作りやすいからに違いない。ではなぜ現代のコンピュータは横長なのだろうか、その答えは簡単だ。映画やドラマを見るためである。つまり 1970 年代に議論された、思考の道具としてのコンピュータ、とは異なる視点が見えてくる。1980

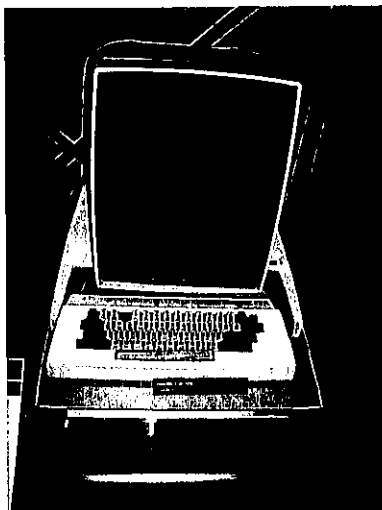


図 3.5: Alto (Photo of the Xerox Alto, taken by Martin Pittenauer)

年代ごろにはやった言葉「マルチメディア」という言葉を聞いたことがあるだろうか。これもメディアで溢れている今では死語かもしれない。その当時、その定義は曖昧で言葉だけが一人歩きをしていた感が否めない。情報処理学会などでのシンポジウム・タイトルにも使われたりしたが、会場からは決まって「マルチメディアとはなにをさすのか」とか「メディアが集まっただけでなにが研究なんだ」という陰口に似た質問が舞った。映像技術や音声技術の発展に伴い、特に高度成長期の日本のメーカーによって、メディア技術の小型化、家庭への浸透、といった現象がおこっていた。VHS, Beta などの記録技術や、ウォークマンといった小型携帯型プレーヤ、そういった懐かしい品々の名前が並ぶ。小型化すれば、それらを組み合わせることも考えられよう。また、コンピュータに取り込むという方法もあり得るはずだ。マルチメディア時代はどこからくるか、という問いに対して、様々な説があった。コンピュータ屋さんの解答は、コンピュータが発展してマルチメディア時代がくるとなるが、それだけではない。テレビが多機能化するのだろう、あるいは、電話が多機能化するのだ、という、それぞれの業界の思惑があった。さていまから振り返ってみると、皆さんはど

う感じられるだろうか。

こういったメディア技術の革新はメディアの開発史独自のものではなく、背景に Alto から始まる思考の道具としてのコンピュータの役割も見え隠れしている。そもそも Alto の提示した WYSIWYG は、その後 DTP(Desk Top Publishing) と呼ばれるものである。つまり机の上で出版する、という意味である。活版やオフセット印刷の時代、出版は出版社、印刷所へ頼まないと実現しないものであったが、DTP によれば、WYSIWYG を備えたコンピュータ一式があれば手元で可能である、ということだ。実際、現在では出版社や新聞社が DTP を導入している。これは一種のコンピュータを利用したメディア技術である。当然のことながら、DTP の考え方は他のメディア制作現場へも転用できるはずだ。同種の言葉に、DTM(Desk Top Music) というのがある。音楽制作をコンピュータで行うことを指す。DTM の先駆けとして、有名な事件は NeXT コンピュータによる音楽のデモンストレーションであった。アップル・コンピュータ社をやめたスティーブ・ジョブスが 1985 年に創業したのが NeXT Inc. であり、カーネギーメロン大学の OS Mach や、アドビ・システムズの PostScript といった、多くの技術的マイルストーンを結集した NeXTSTEP オペレーティング・システムが開発された。これを搭載したコンピュータが NeXT コンピュータである。NeXT コンピュータは商売としてはあえなく失敗となるが、そのコンセプト、技術は MacOSX へと受け継がれている。現在では、パーソナル・コンピュータ上で、本物と見まがう音色のオーケストラが合成できるシンセサイザーと様々な DTM ソフトウェアが遜色無く動く²。多くのドラマや CM で DTM を使ったバックミュージックが使われる時代となった。

DTP, DTM とくれば、Desk Top Movie とか、Desk Top Video といった言葉がありそうであるが、どうもないようだ。それは、そういった分野がないからではない。映像技術はコンピュータの固まりであり、コンピュータ技術とほぼ同等と言っても過言ではない。おそらく一つの言葉で包含しきれないのだろう。こうしてどのようなメディアもコンピュータから出てくるのが無くなった。かつては印刷といえば印刷所という非コンピュータ的プロセスを通り、音楽であれば、演奏という人間的なプロセスを通った。しかし、今ではほぼ全てのプロセスがオフィスの机の前で済んでしまう。「出てくるのが無くなった」

²聞き慣れてくると差が分る物もある。

「全てがオフィス」というのは言い過ぎかもしれないが、少なくともそういう方向を確実に世の中が目指していることは否めない。

エンゲルバートからの道具としてのコンピュータに貫かれている一つの思想はメタファーである。デスクトップ・メタファーあるいは、メタファーとマジックと呼ばれる考え方である。デスクトップ・メタファーはいわゆる机の上のメタファーである。コンピュータのディスプレイは机の上であり、そこに書類が積み重なっているのがウィンドウである。書類はファイルへ保存したり、またとりだしたりすることが出来る。いらぬ物はゴミ箱へ捨てる、といったメタファーである。本来メタファーとは、コンピュータのインタフェースが現実からのメタファーによって設計されているという意味であるが、実際のユーザにとってはその逆の方向に思考が働く。おそらく、メタファーによって、ユーザはコンピュータの世界へ入って行きやすくなる。初めてのことで、またなれないことでも、現実の世界からメタファーを求めることによって、マニュアルに頼ること無くコンピュータの操作を行うことが出来る。

だが、机の上と全く同じことしか出来なければ、コンピュータはいらない。コンピュータを導入する理由は、本物の机の上では出来ないマジックがあるからだ。例えば、最近の大学期末試験では厄介者のコピーアンドペーストは、右の書類から左の書類へ、文書の一部を選択するだけで、文章を移動することが出来る。通常の書類では出来ないことだ。現在では表計算ソフトウェアとして知られ、1979年にダン・ブリックリン考案、ボブ・フランクストン設計によるソフトウェア VisiCalc は、数字を並べて表を作ることが出来るソフトウェアである。だがそれだけであれば、紙の上に表を作るのとなんら変わらない。小計、合計といった表には付き物の計算が、入力処理にともなってリアルタイムかつ自動的に行われる。これもメタファーとマジックの一つの形である。ユーザ・インタフェースを設計するとき、メタファーにウェイトをおきすぎると、確かに分かりやすいが、利用価値の低いシステムが出来上がる。一方、マジックにウェイトをおきすぎると、便利なのかもしれないが、利用方法の分かりにくいシステムになる。メタファーとマジックのちょうど良い兼ね合いが重要だ。

DTP, DTM、あるいは様々な映像技術の制作現場では様々なメタファーに覆われている。DTP は出版前に、出版と同じ状態をコンピュータ上で作り出す。DTM は音楽の録音前に音楽を聴くことが出来る。それは制作物のメタファー

であると同時に、それを制作前に見られる、聴くことが出来るというマジックでもある。これらは一種のシミュレーションである。シミュレーションとは現実世界のメタファーである、と同時にそれは未来を予測したり、過去を再現したりすることのできるマジックである。そうなれば、道具としてのコンピュータの究極は、全てのことをコンピュータ上でシミュレーションできればよいのか。その考え方もまた面白いかもしれない。机の上のメタファーだけでなく、書斎のメタファー、住居のメタファー.....、都市や国のメタファーと、全てをコンピュータの中に入れてしまう。その間で動くメディアは今までは実世界に出てきて交換されていたが、未来はコンピュータの中で、全てが重量もなくコストも最小限の電子情報で交換される。究極の姿はヒラリー・バットナムの「水槽の脳」である。我々が眺めている外界の姿はコンピュータが作り出すCGに過ぎないのかもしれない。

しかし、そのようにはなっていないようだ。Second Lifeに見られるように、ネットワークでつながった仮想世界をCGで再現し、その中で暮らすシステム、というのが一時脚光を浴びた。Second Lifeの中に店出するいくつかの衣料品メーカーが現れた時期があった。しかし、実際のところそれほど発展していない。実際に仮想的な三次元空間は人間にとって違和感がある。これは三次元化技術がまだ未熟だからかもしれないが、そもそも、コンピュータの世界は現実より情報が欠落しているということもマジックの一つなのだ。かつて、それこそマルチメディアに夢があった時代、テレビ電話や、先にも少し触れたが環境を転送する技術、立体テレビといった様々な技術が提案された。現在、技術的に可能なものも少なくないはずであるが、実際に実用化されていなかったり、ごく限定的な利用にとどまっている。たとえば、2D映像は2Dであるが故に視聴者に映像のスケールを想像させることが出来る。数センチ角の画面のスマホの画面でも、大映画館に行かずとも、それなりに感動できるのはそのためである。もし小さなスマホが完全な立体ディスプレイを備えていたとしよう。おそらく完全に立体的な富士山は、ホログラムシールのような小さな富士山であろう。確かに立体映像は臨場感があるが、それ以上の想像がリセットされてしまうのだ。それゆえ、3D映画が作られつつも2D映画が無くならない所以である。これは、メタファーのウェイトが大きくなりすぎた例と考えることが出来る。

現実以上の現実はない、ということと、このようなメタファーの限界を考え

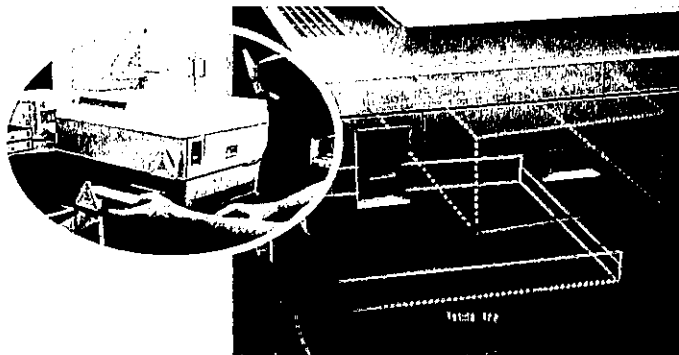
合わせれば、もう一つのアプローチがマジックを現実を持ち込んでくる方法である。これが強化現実感 (augmented environment)、あるいは、ユビキタス・コンピューティングと呼ばれる技術である。

3.3 ユビキタス・コンピューティング

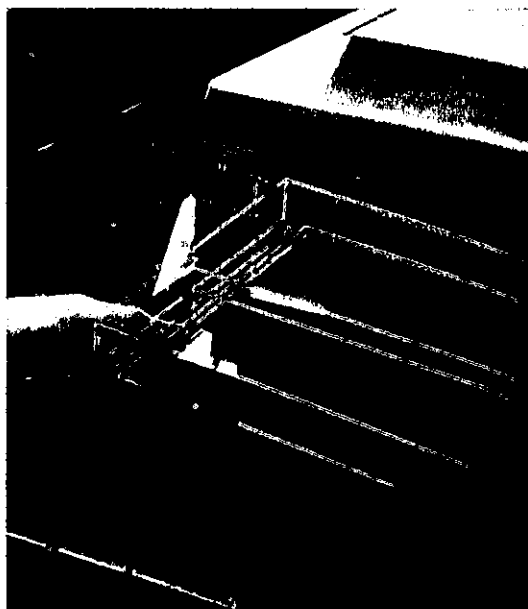
ユビキタス・コンピューティングにおける空間モデルについて眺めてみよう。ユビキタス・コンピューティングの起源については文献 [10][11] に 1980 年代から 1990 年代初頭にかけての先進的な研究がまとめられている。ユビキタス・コンピューティングは現実空間で生活している人=ユーザの居る位置やユーザの行動や状態に応じて、現実空間の環境に埋め込まれた多数の組み込みコンピュータによって、様々な電子的サービスを提供するメカニズム、システム、アーキテクチャ、概念の総称を言う。一方、ユビキタス・コンピューティングによってサービスが受けられる空間のことを脳強化環境という。究極の脳強化環境ではユーザはコンピュータを意識することがない。

脳強化環境では、ユーザの位置や行動が現実空間とどのように関わっているかをシステムが把握している。このためシステムの中には現実空間のモデルが存在する。現実空間の中にいるユーザは一方で、システム中の空間モデル上でのアバタ=化身として認識される。ユーザの行動はアバタの動作や属性としてモデル中のアバタに反映される。一方、システムがアバタに対して行う各種のサービスは、現実空間に埋め込まれた様々なコンピュータ=アクチュエータから実際の人=ユーザに対するサービスとなる。

このような発想は結構古くからあったが、まだ実現されていない様々な先進的な実験がなされている。図 3.6 はオフィス機器のマネジメントを目的とした実験システムである [10][12]。透過型ヘッドマウントディスプレイをかぶり、オフィス機器を眺めるとそこにオフィス機器の内部が映し出される。プリンタのトナーやトレイの位置を示し、どのようにして取り出したら良いかなどメンテナンスの方法を指示する。図 3.7 は、建築と景観シミュレーションのために開発されたシステムであり、I/O bulb と呼ばれる新しい「電球」に照らし出された机の上の模型に、影や風のシミュレーション結果が映し出される [13]。I/O bulb は机の上の都市設計から、「建築空間の何百、何千という電球を I/O



(a) 透過型ヘッドマウントディスプレイで
レーザープリンタの簡単な保守説明を行う。



(b) ユーザの動作とともにその結果の状態を表示。

図 3.6: KARMA

3.4 位置情報システム

ここに挙げた例は、かなり先進的で未来的なものであるが、実際に簡易なヘッドマウントディスプレイなどが開発され、現在、そこへの第一歩となりそうな様々な実用品が現れている。たとえば、Google Glass は町の中にいろいろな情報を埋め込む技術である。コンピュータを身にまとうかどうかは別として、カーナビやスマートフォンによるリアルタイム・ナビゲーションの数々の技術は、環境にコンピュータを埋め込む技術である。町の中を歩いている、あるいは書齋に居ても、空間を情報が飛び交い、通信路で結びついている、そういうユビキタス・コンピューティングへの一つの布陣なのだ。

LBS(Location Based Service) という言葉がある。これは ISO/TC211 で以下のように定義されている。

Location based service (LBS) Service whose return or other property is dependent on the location of the client requesting the service or of some other thing, object or person.

Note. The client is referred to the vehicle and the other things are hereafter collectively referred to as the targets.

つまり、サービスの形態がサービスを受ける側クライアントの位置に関するものである。広く言えば、ユビキタス・コンピューティングで扱った様々なサービスは LBS が基礎にある。その LBS の中心的なサービスがナビゲーションである。コンピュータや GPS によるナビゲーションで最初に普及したのがカー・ナビゲーション、あるいは ITS(Intelligent Transport Systems) と呼ばれる自動車を取り巻く様々な情報システムであろう。

プローブ情報システム [23] は、自動車をインターネットと接続し、様々な情報を自動車から吸い上げることによって、運転者のみならず歩行者や環境監視などに様々なサービスを提供していこうという実験である (図 3.9)。現在の自動車は走行制御系だけでもかなりのインテリジェント化が進んでおり、既に様々なセンサが搭載されている。これらのセンサからくる情報を走行制御のみならず、インターネットを通じてサーバへ集め、データベース化しここから様々な

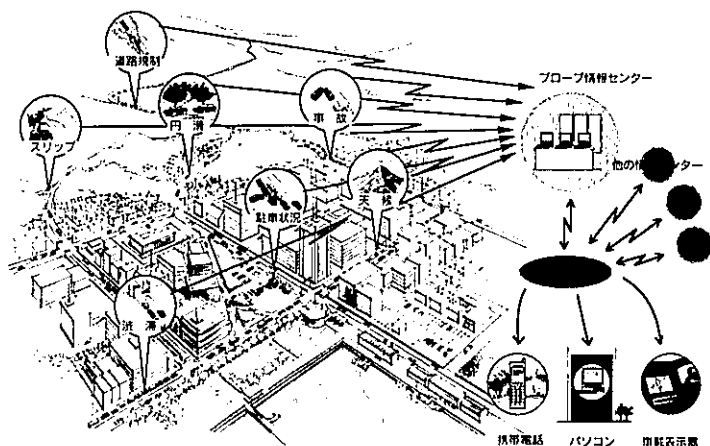


図 3.9: プローブ情報システム

環境情報を抽出することができ、渋滞の情報、降雨、路面凍結の情報の取得に有効活用する。

インターネット ITS プロジェクト [24][25] ではプローブ情報サービスのほかに、一般ドライバーを主なターゲットとした様々なサービスとしてガソリンスタンドにおけるガイダンス、駐車場における決済、コンテンツ配信等のサービスを位置情報を利用し、タイムリーに行う実験を行っている。

プローブ情報システムは町中に豊富に存在する移動体である自動車を利用し、情報を集めることだった。その情報の中に、気象情報などの環境情報も含まれる。このように都市の環境情報を集めることは、地球環境保全においても重要なことである。都市環境の中に様々な環境センサ・ステーションを配置し環境情報を収集、可視化するプロジェクトとして NTT 環境エネルギー研究所の取り組みがある [26]。これらのセンサにより環境中の二酸化窒素、オゾン、浮遊粒子状物質 (SPM) の量が位置情報とともに取得可能である。得られたセンサの点情報を面的広がり拡大し、街角レベルでの詳細な環境情報を得ることができる。この情報をもとに建物や地形による環境の影響を評価、解析、環境の将来の状況を三次元的に予測する (図 3.10)。

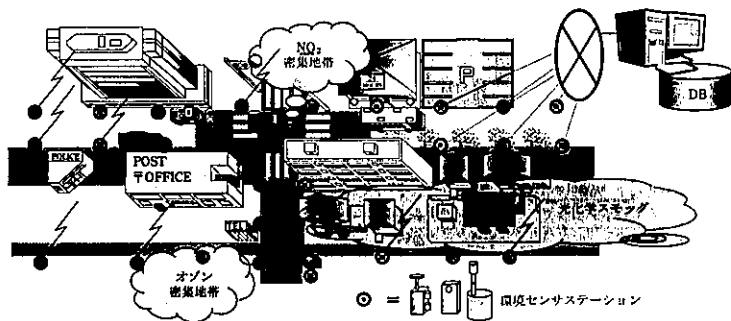


図 3.10: ユビキタス環境センシング

GPS 付き携帯やスマートフォンが普及するに従って、マン・ナビゲーションも実用化されつつある。人は道路以外にも屋内や地下街といった道路よりも複雑な径路を通る。その意味で人のナビゲーションは自動車のナビゲーションより難しいのである。特に屋内のナビゲーションや地下鉄のナビゲーションと言ったように GPS の電波の届かないところを行動する場合もある。屋内測位が重要な技術となってくる。屋内測位には GPS の信号をリレーする方式や端末と屋内に設けられた基地局との間の電波の強度から基地局からの相対的な位置を割り出す方法など多数提案されている。二次元または三次元の座標値を得るものではなく、赤外線タグ等によって人があるエリアに入ったことを感知する方式もある。また、様々な印刷媒体その他にある文字やバーコードから位置を割り出し、適切な情報サービスを行う方法も提案されている [27]。

マン・ナビゲーションの研究においては、歩行者の年齢や趣向、性格や目的にあわせて動線を変更するサービスなどがある。特に身体障害者や老人の支援といったテーマが取り上げられている [28]。個人の ID カードにあらかじめ登録された情報と周辺の様々なセンサやアクチュエータからくる情報をつきあわせることによって最適なサービスを提供することが可能である。

3.5 情報は誰のためにあるのか

このようなナビゲーションの技術は現在既に実現しているものも多い。いわば我々はユビキタス・コンピューティングの入り口まで来ている。エンゲルバートやユビキタス・コンピューティングなど、すべてはユーザのため、道具としてのコンピュータである。今まで引用してきた論文、研究を眺めると、それらはユーザの視線で書かれている。ユーザが最も快適と考える環境を提供する技術たちである。だが、こうして眺めてみると、研究者のまなざしと、実際に実現されてくるコンピュータ社会の間の温度差もある。DTP は迅速かつエレガントな出版を実現したかもしれない。しかし、現在のデザイナーのオフィスは、エンゲルバートが思い描いたオフィスだったのだろうか。いままでは役割分担で作成されていた出版物が DTP の前に座ったデザイナーに一挙に集中する。同じことが、CM ソングを作る DTM アーティストにもおこる。彼らはフィードバックされるあらゆるクレームやコメントと昼夜問わず戦い、睡眠不足で仕事に疲れ果ててしまう。

いつでもどこでも求める情報がとりだせるスマートフォンはどうだろうか。様々なサービスにはユーザ登録が付き物であり、時として有料である。また便利な LBS では常に、私の位置情報が様々な企業や組織のデータベースに送信される。確かに、いまどこに居ますよと案内してくれ、近くのレストランを紹介してくれるのは便利である。アマゾンで本を購入すると、関連する書籍も選んでくれるのは、時々便利である。様々なネットショッピングも家に居ながらにして便利である。しかし、時がたつにつれてダイレクトメールが次第に増えて行く。カーナビゲーションにおける渋滞情報は、一方で、渋滞に巻き込まれ憂鬱な時間を過ごしている他のユーザから送信される情報たちではある。また、便利さにかまけていると時々フィッシング詐欺の引っかかる。ネットワークには危険もいっぱいあるのである。

研究者のまなざしは常にユーザ目線であるが、技術が現実化すると実際はバータである。されど、バータなのだ。我々の情報を吸い上げる何者かは、我々の息の根を止めることはしない。常に持ちつ持たれつであり、我々は自分の個人情報に少しづつ切り売りすることによって、先方もそこそこのサービスを提供する。

最先端の研究であっても、あるいは往年の研究であっても、研究者のまなざしは常に「古き良き時代」である。かつて研究室の仲間たちと六本木にある TRON ハウスという、ユビキタス・コンピューティングのモデルハウスへ見学に行ったことがあった。出来たばかりの屋内に入ると、窓、電灯、ガス、水道、すべてがコンピュータ制御されている。清潔でウッディーなシステム・キッチンのある戸棚は地下に設置された巨大な可動式冷蔵庫の入り口だという。制御画面から「オレンジジュース」と入力すると、床下から鈍いモータの音が響き、待つこと数分、戸棚が自動的に開いた。中には、オレンジジュースのパックが一つ入っていた。役に立たないものにはなにか未来的なオーラがある。役に立たないものが役に立つようになるにつれ、次第にオーラは消え、社会の中に潜むバータに絡めとられて行く。

技術が実用化された社会においては、研究者によってユーザと呼ばれる多くの人々は、自分の情報がどこでどのように利用されているのか、確実に知ることは出来ない。確かに、個人情報保護法があり、我々の個人情報は守られているように見える。しかし、自分の情報を開示請求したところで、世界中に広がったネットワークのどこかへ埋もれてしまった自分のデータをすべて追いかけることは出来ない。

スマートフォンから上がってくる人々の行動情報や、ショッピング・サイトにアクセスされる多くの購買情報は集計化され様々な顧客行動分析に利用される。それらは新しい商品やショッピングサイトの開発、顧客の誘導に使われる。いわゆるデータ・マイニング、あるいは、最近ではビックデータと呼ばれる技術が生まれつつある。このような分析は顧客や市民を守ることもある [29]。しかし悪用され害することもある。すべてはバータなのだ。

西垣によれば、機械は生命ではない [1]。アロ・ポエティック・システムであって、オート・ポエティック・システムではない。情報はヒトが生み出すものであって機械ではない。しかしここでヒトとは誰のことだろうか。もともと分類学上の定義が明確な訳ではなく、また脳死や中絶、つまり、ヒトが誕生し、死んで行く、その境界線の問題を考えても、どこからがヒトでどこからがヒトでないのか曖昧である。だが、そういった曖昧以上に、私にはヒトとはコンピュータ科学者、エンジニアのことと聞こえてくる。コンピュータ・エンジニアや研究者にとってみれば、世の中のコンピュータ・システムが如何に複雑になり得て

も、それらは彼らの知識、手中にあるからだ。インターネットがどのように情報通信しており、GPS がどのようなメカニズムで位置測位を行っているかは、たとえ、自分自身が直接その知識をもっていなくても、誰かに問い合わせれば分かることであるし、その気になって勉強しようと思えば、どこかにドキュメントが存在する。情報を作り出しているのはそういったヒトたちであり、そうでない人たちにとっては、コンピュータはオート・ポエティック・システムとして厳然として現れてくる。

3.6 歴史的地図の多様性

3.6.1 Google Map

いま GPS によるナビゲーションが急速に発展している中、ネットワークにおける、地理情報、平たく言えば地図の需要は大きく伸びている。だが、ネットワーク上の地図は Google を筆頭にある種の画一化を見せている。この画一化は、インターネットの普及に伴うものとも捉えることが出来るが、一方で、経緯度という問題にまつわる地図の技術史的な背景も存在する。この技術史的背景とその周りに存在する地図の多様性を見てみよう。

3.6.2 経緯度の系譜と地球説

大地が球形であることを最初に唱えたのは幾何学者として有名なピタゴラス（紀元前 6 世紀）であった。大地が神の創造物である以上幾何学的に完全な形である球でなければならないと考えたからだ。その後、月食の際の大地の影が常に円形であることを根拠として、その節が正しいことを裏付けたのはアリストテレス（紀元前 384～322 年）である [30][31]。

ストラボンの『地理学』の記載によれば、地球に経緯度を入れて図化した最初の人物は、エラトステネス (Eratosthenes, 紀元前 275 年～紀元前 194 年) であるとされている。『地理学』の記載によりしばしば想像図が描かれている (図 3.11)。

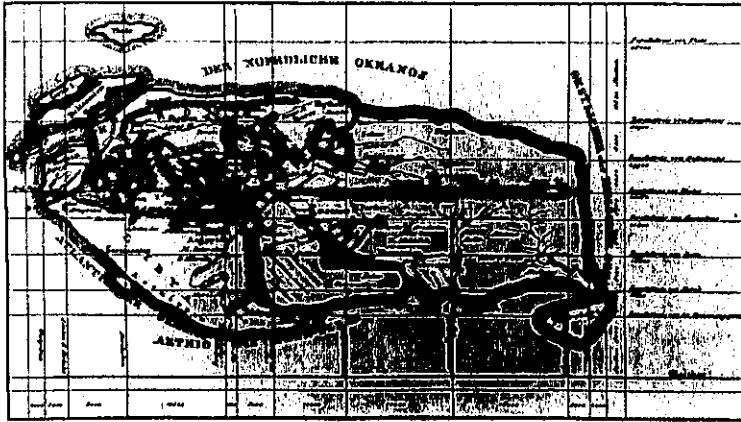


図 3.11: エラトステネスの世界地図 フォルビガーによる推定 1842年 (文献 [30] より抜粋)

エラトステネスは夏至の日、シエネ（現在のアスワン）にある深い井戸の底へ陽の光が到達することを聞き、シエネの夏至の日の南中高度が 90 度であることを知った。アレキサンドリアの太陽の南中高度との差と両市の距離の差を合わせることで地球の全周が計算できると唱えた。彼の計算した全周距離は各説あるが、250000 スタディア (46,250 km) となったとされる。地球の全周距離の計算により彼は経緯度という概念に到達した。

エラトステネスの地図においては、経緯度は著名な場所を通る不等間隔の直線であることが特徴的である。エラトステネスの地図において興味深い点は経緯度が不等間隔であるということである。ロードス島、ヘラクレスの柱（ジブラルタル海峡）シチリア（メッシナ）海峡、アレクサンドリア、シエネが書き込まれた経緯度上にあるとされている。

古代の地図学の水準を高めたのは紀元 2 世紀のアレクサンドリアの天文学者プトレマイオス (Ptolemaios) であった。プトレマイオスは正距円錐図法、プトレマイオス第二図法を考案し、約八千地点に及ぶ世界各地の経緯度数値を使って地図の精密化をはかった (図 3.12)。

プトレマイオスの生きた時代はローマ帝国最盛期ではある。ローマ人社会に



図 3.12: プトレマイオスの世界図 正距円錐図法 (15 世紀末写本、概略) (文献 [30] より抜粋)

においては関心が科学の理論ではなくて、応用面に向けられていたこともあり、その優れた地図学の進化は認められることがなく、ルネッサンスを迎えるまでのラテン世界とは無縁であった。一方、イスラーム社会ではいち早くプトレマイオスの地図学を吸収し展開していく。ギリシア学術書の翻訳を奨励したバグダッドのカリフ（教王）アル・マムーン（19 世紀）は子午弧 1 度の長さを測定させた。その結果は約 113km であり、きわめて現在の値に近いものであった。

イスラーム地図学完成期の代表とされるのは、シチリアのノルマン王宮で活躍したアル・イドゥーリーシーによって作られた世界分域図であると言われる（図 3.13, 図 3.14）。特徴的なことは南が上になっていることである。一般にイスラーム圏では南を正面と考え東を左、西を右と呼ぶ内陸アジア諸民族の風習と関連していると考えられている。一方で、各分域図には経緯線はなく、投影法もはっきりしない。しかし当時モンゴル等のイスラーム周辺地域への波及を見ていくと、図 3.15 のように経緯線が記入されているものもあり、プトレマイオスの地図学のイスラームへの浸透の深さをうかがうことが出来る。

イスラーム世界の科学は十字軍によるキリスト教徒とイスラーム教徒の接触などにより、ラテン世界へ知れるようになった。ロジャー・ベーコンはプトレ



図 3.13: アル・イドゥーリーシーの円形世界図 (1154年 オックスフォード
ボードリアン図書館) (文献 [30] より抜粋)

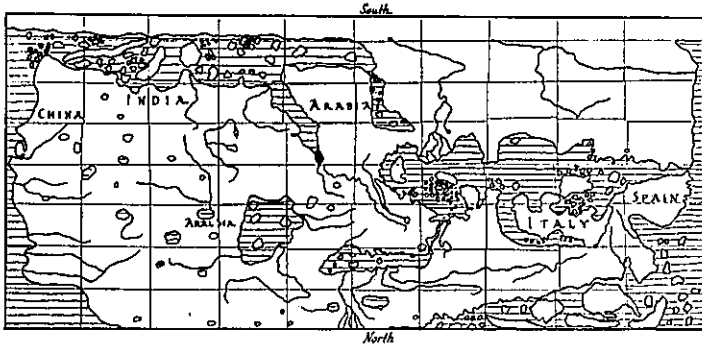


図 3.14: アル・イドゥーリーシーの世界分域図 接合したもの (1154 年) (文献 [30] より抜粋)

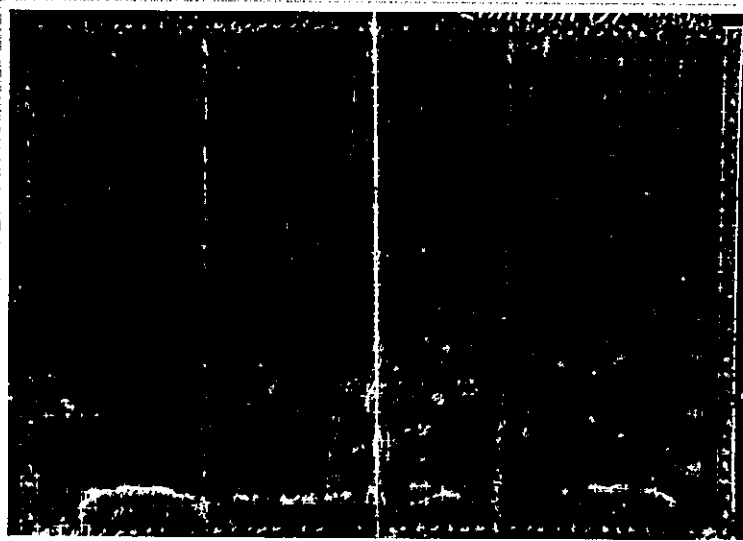


図 3.15: ハムダッラー・ムスタウフィのイラン図 (1330 年頃 手書) (文献 [30] より抜粋)

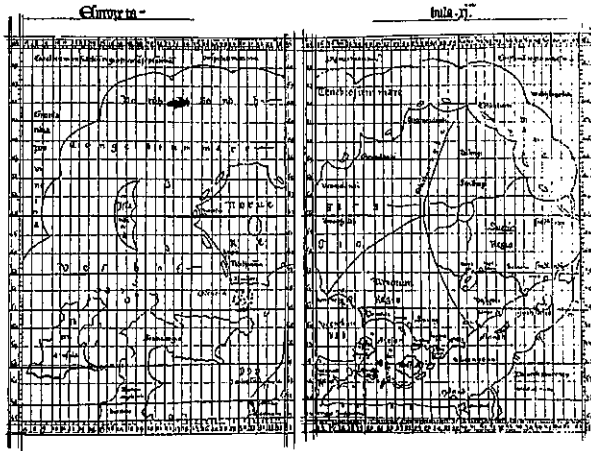


図 3.16: クラヴスの北欧図（プトレマイオス「地理学」1427年写本所収）（文献 [30] より抜粋）

マイオスや、イスラーム科学を引用しつつ、経緯度に基づく地図の必要性を説いた。1427年クラヴスによってプトレマイオスにはなかった北欧の入った経緯度地図を作っている（図 3.16）。これよりルネッサンスにおける地図学が花開く。ルネッサンス期における地図はプトレマイオスの地図学の継承が中心であった。この時期はラテン世界がその外側へと海路を切り開く大航海時代であり、これによってもたらされる世界の情報を盛り込み図化するのにふさわしい投影法が模索されたが、先に述べた容易にプトレマイオスの投影法を超えることが出来なかった。

16世紀中期から17世紀中期に至る約一世紀間は、ヨーロッパ地図学の華がフランドル地方を中心に開いた時期である。そのフランドル学派が生み出した地図学者がメルカトルである。彼が生み出した正角円筒図法こそルネッサンス地図学への決別を意味する。この図法は、数学的な投影法を持たない海図ポルトラーノとプトレマイオスの図法との合体であり、海図としてふさわしい地図である（図 3.17）。しかしこのメルカトルのアイデアが定着するまでに

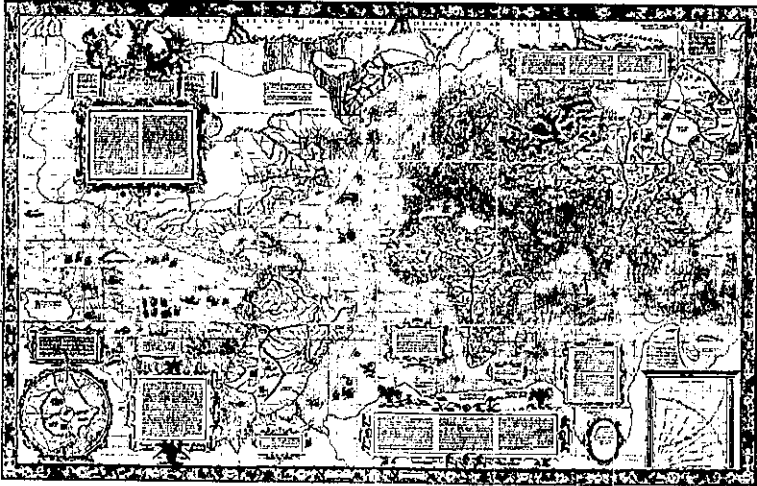


図 3.17: メルカートル正角円筒図法世界図 概略 (1569 年刊 銅版)

http://en.wikipedia.org/wiki/Mercator_projection より抜粋

はまだしばしの時間が必要であった。というのも、この図法の計算方法が難しかったからである。

1700年代に入ると、フランスおよびイギリスの世界の測量と地図製作の全盛期となる。フランスの天文学者カッシーニ一門の業績は大きい。ジーン・ドミニク・カッシーニ（カッシーニI世）は天文学者であった。ガリレオの木星の衛星の研究を通して正確な時間が測定できることを応用し、正確な経度の測定を行った。ヨーロッパ中の天文学者に手紙を出し、この方法を用いて測定された各地の天文台の経緯度を送ってもらうようにした。得られたデータを直径約10メートルの丸い地球図の上にプロットして行き、徐々に正確なヨーロッパ図が浮かび上がってきた。さらにフランス学士院は天文学者や天文学の教育を受けた宣教師を、エジプト、南アメリカ、西インド諸島、中国やシャムへと派遣し、やがて彼らからの情報も届きはじめ、地球全体の正確な形状が次第に明らかになってきた。これをカッシーニは、プラニスフェール・テレストルと名付けた。

カッシーニのもとで、測量を行ったピカールのグループはカッシーニの天文学の方法と三角測量を組み合わせ、子午弧1度の精密な測量を行った。この仕

線の集中が感じられるものとなっている。

エラトステネスの次の時代に現れたプトレマイオスは投影法とセットで、等間隔に二次元の座標値として地球上の位置をくまなく表すことのできる経緯度という概念を確立し、これが今日まで発展してきた。今日の世界測地系においても重要な役割を果たしていることは言うまでもない。

しかしプトレマイオスの経緯度概念ははじめから世界中に広まったわけではなかった。特にローマ文化圏では広がらず、後にイスラーム社会の力によって広まった理由はなんだろうか。またヨーロッパにおいてはルネッサンス期以降に広まったのはなぜだろうか。推測するに地球規模の商業、経済の発展にともなう航海術、いわば地球規模のロジスティックスからの要請があつて始めて意味を持ち始めた概念であつたのではないだろうか。交通網が整備された一国の範疇を行動するためには、経緯度というグローバルな指標の必要性はあまりなかったと考えられる。人工的な標識の存在しない海や砂漠における、航行に経緯度の威力が発揮される。

3.6.3 地誌としての地図

経緯度の概念が日本に伝わったのはいつであろうか。江戸時代の日本の庶民は鎖国の時代であつたにも関わらず「地図」へのアクセスは容易であつた。一部の地図や絵図を除いて幕府はことさら取り締まったりはしなかつた。海外へ渡航することはかなわないはずなのだが、いやそれだからむしろ、世界地図についても庶民の間で享受されていた。図3.19は日本にも入ってきたマテオ・リッチの世界地図である。また図3.20は長久保赤水の日本地図であり、経緯度が入っていることが分かる。

経緯度に関する定義を書き記した例は我々が知る限りでは、宝暦5年(1755年)森幸安[32]によって手書き彩色された「大地世界 両卵合掌渾円図」(図3.23.)がある。この図の識語には経度について「それ大地広大といえども、その里数極むるところ有りて著明なり。今の世界と号するは、その周囲三百六十度にして、方五千里にすぎることなし。」と説明している。はっきりした説明はないが、1752年に同氏によって描かれた「渾円天度合体図」については「天度」という語で経緯度を表している。



図 3.19: マテオ・リッチの世界地図「坤輿万国全図」
(万暦 30 年 1602 年刊) (文献 [30] より抜粋)

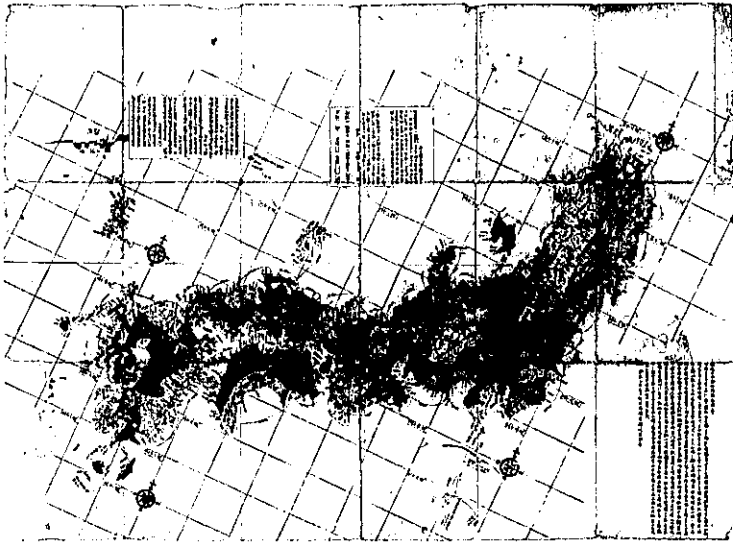


図 3.20: 新刻日本輿地路程全圖 長久保玄珠 (赤水)
安永 4 年序 1775 年 (国際日本文化研究センター蔵)

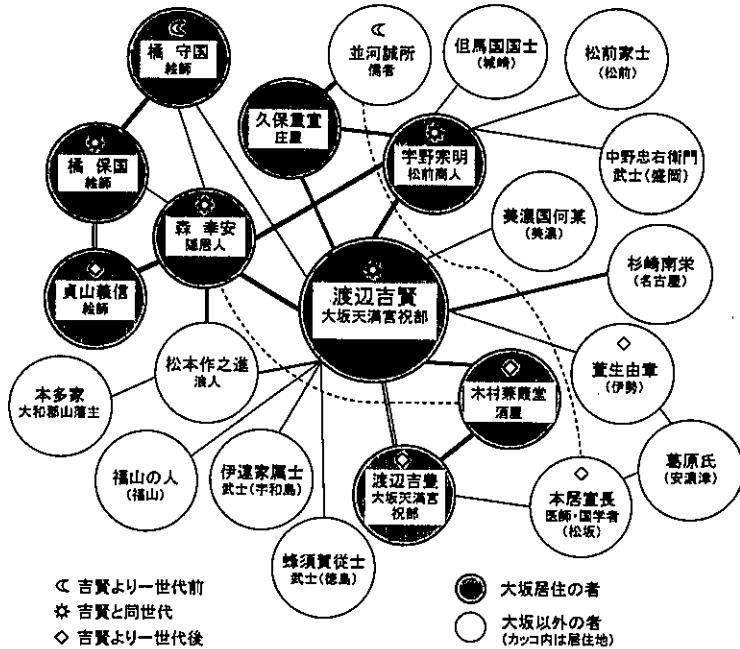
森幸安は日本の地図のみならず外国の地図をも模写し残している。どのようにして外国の地図を手に入れたのか興味深い。この問題について上杉は近世の知のネットワークという点から一つの解答を与えている(図 3.21)[33]。この図によれば、森幸安のみならず、江戸時代の身分制度を超えて様々な人同士が交流し、地図を交換、模写しあったことが分かる。本論文には朝鮮通信使来日について論じられており「日朝間の『知』の交流の場において地図が交流の道具として一定の役割を果たしていた」とされる。このように地図が外国をも含めた人的ネットワークの中で交換されていたことがうかがえる。森幸安の膨大な地図コレクションはいわばこのような知的ネットワークの人々の所蔵地図のデータベースであるとも言える。

このような人脈を駆使し地図を集めていくなかで、森幸安が求めていたものは「地の理」であった。「地の理」が何を意味しているのかは彼自身ははっきりと言及していない。しかし彼の記述を総合するに、縮尺、測量といった幾何学的な客観性と根拠を保証する技術など、現在で言えば地理学や天文学に相当する学問のことを言っているのではないかと思われる。文献 [32]p.12 の記述を引用すると、

幸安「従来の地誌研究者による絵図は、日本図や山城図が作られた程度でしかも精度が悪い。「世にある地図」(国絵図か)も自国を賞して土地を実際より大きく見せているため地理を全うしていない。」

とある。

森幸安の地図を眺めていると識語の量が多いことに気づく。森幸安は元禄十四年(1701年)に京都で生まれたが、その活動時期は享保末年頃(1729年)から宝暦十一年頃(1761年)までであるとされる。この活動期の中で前半は地誌作成を中心とした活動で、後半は地図作成時代と大きく分けることが出来る [36]。地図作成時代においても、地誌という視点から地図を捉える姿勢を貫いたのだろうと考えられる。地誌においては地図は文章に対する挿絵である。これを地誌が図を説明する立場に逆転させたのである。これにより識語自身も自由を得たと言えよう。地誌における地域の風物や名勝などに関する記述などに加え、先にも挙げたように経度や天度に関する説明など、様々な新しい知識に関する説明や地図の由来など、その説明範囲は大きく広がったと言える。また地図と



実線は本文中で指摘した地図の貸借関係（関係が深いことがわかる場合は太く表示）、二重線は親子ないし師弟関係、点線は有坂 [34]、上杉 [35] に基づいた地図の貸借関係を示す。

(文献 [33] より抜粋)

図 3.21: 江戸時代の地図収集ネットワーク

いう世界共通語によって外国の風物や知識が入ってくるようになったのである。

森幸安によれば識語にも客観性と根拠が求められる。彼は地図を作るにあたって、一部の対象地域は巡視した。また巡視できない遠くの地域や考証により作られる昔の地図に関しては様々な文献により情報を確かめることを怠らない。こうして作成される図は「絵図ではなく地図である」と彼は主張する。この地の理に関する森幸安の態度は、近世における地理情報の必要性を説いていると言えよう。森幸安の地誌と地図の関係について [37] に詳しい分析があるので参照されたい。

森幸安の活動は 1751 年ごろから飛躍的な展開を遂げる。それまでの彼の創作は日本国内の地図が中心であった。この時期を境に彼の興味は外国へと広がる。寛延 4 年 (1751 年) 正月 5 日、日本志興地部「天文之図」を皮切りに日本志シリーズの制作を開始する。世界地図として、「渾地図識」「大地円球天合 三大界五大洲萬国図」「大地世界 両卵合掌渾円図」(図 3.23)「三大界五大洲 万国月旋潮候之図」の 4 つの地図が描かれる。続いて東アジア 3 図、辰旦 2 図、朝鮮 2 図 (図 3.24)、蝦夷 3 図が描かれる。そして日本全域の地図 9 図へと続く。その日本全域の地図のうちの一つ、「日本分形図」(図 3.25)には次のように書かれている。

かつて、この国において、日本六十八箇国、ようやく始めて形を分くる。固めて分形図と名ずるなり。三大界五大洲のうち、その亜細亜大洲に属するの嶋なり。伝え聞く、大海の中、嶋州の大なるもの八つ。そのうち、日本の国をもつて第一の大嶋となす。四時中正の国なり。方位、亜細亜中央の地よりして、これを定むるなれば、すなわち正東卯の位にあり。一世界赤道の上よりして、これを窮むるなれば、すなわち辰の位にあり。

つまり、この識語において日本を世界およびアジアの中で位置づけている。その位置づけの方法として、ヨーロッパ、アフリカ、アジア、オーストラリア、アメリカをさす「五大洲」と「三大界」を並べている。また「伝え聞く」とあり「古事記」[38]「日本書紀」[39]に由来する日本六十八箇国の定まった経緯などを述べ、日本の国の成り立ちを示す。最後に、

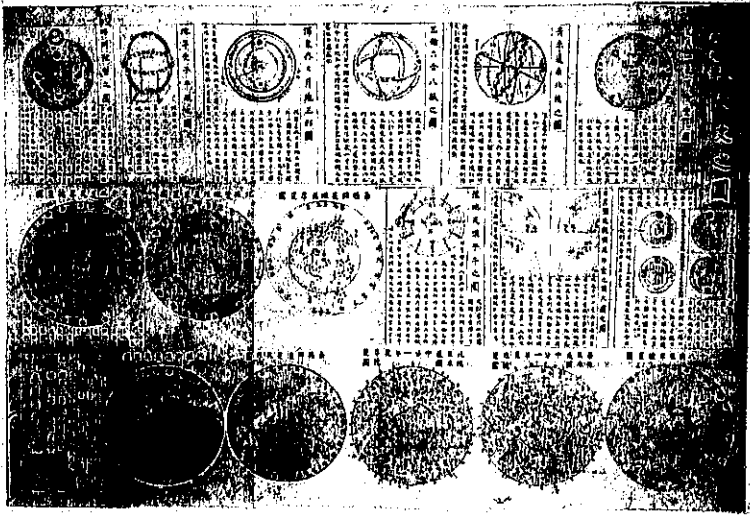


図 3.22: 日本志 輿地部 天文之図 寛延 4 年 (1751 年) 国立公文書館

この図の体たるやなり。六十八州の国々を分くるのみ。つまびらかなることは、次々の日本の図に載す。

と締めくくり、日本六十八箇国各地の地図へと続く。日本志における森幸安の視線は、地球の外側から徐々に降りてきている。これらのうちの山城（おおおよそ現在の京都府）の例を図 3.26 に示す。さらに各国に属する諸地域についても、地域による情報量の差はあるものの、他の成果等とも合わせると図 3.27 のように詳しい図面をそろえることが出来る。

このような視線の動きを見ていくと、森幸安はその数は数百点におよぶ地図を残しているが、ただただバラバラに地図を作り続けていたのではない。実際図面においても、すべての地図は小縮尺の地図から大縮尺の地図まで重なり合い、つながり合う仕組みとなっている。大縮尺の地図の地域は小縮尺の地図の地域のどこかに位置づけられており、また、隣り合う地図同士の境界線はかみ合うように書かれている（文献 [32] p.10. 参照）。

森幸安の日本志以前の地図とも合わせれば、その拡大率は、日本六十八箇国

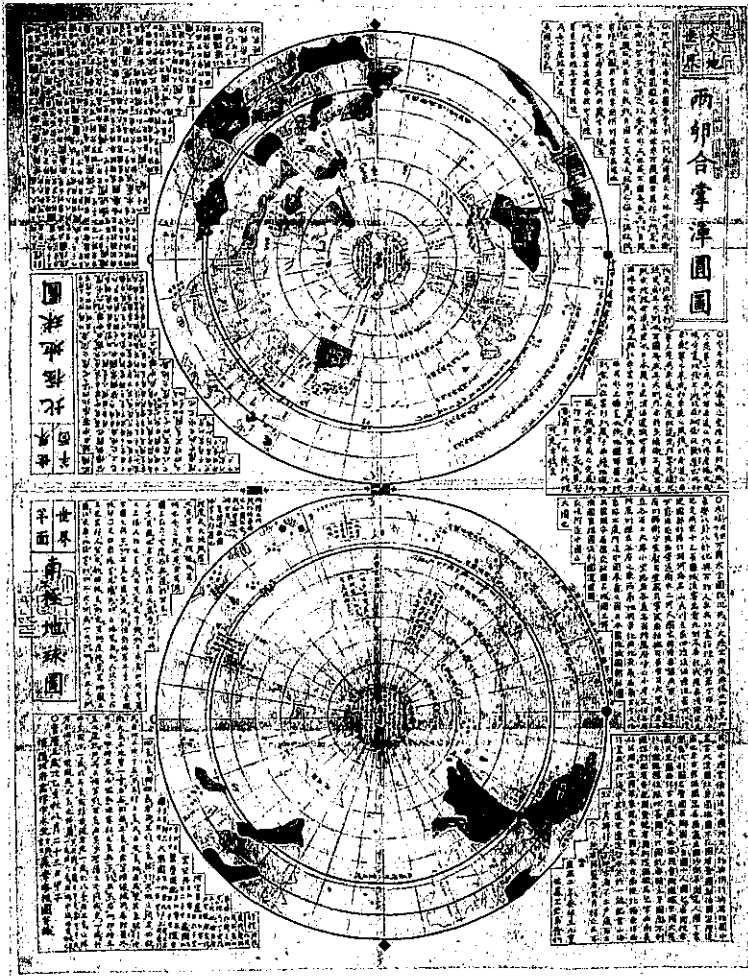


图 3.23: 大地世界 兩卵合掌渾円圖 宝曆 5 年 (1755 年) 国立公文書館

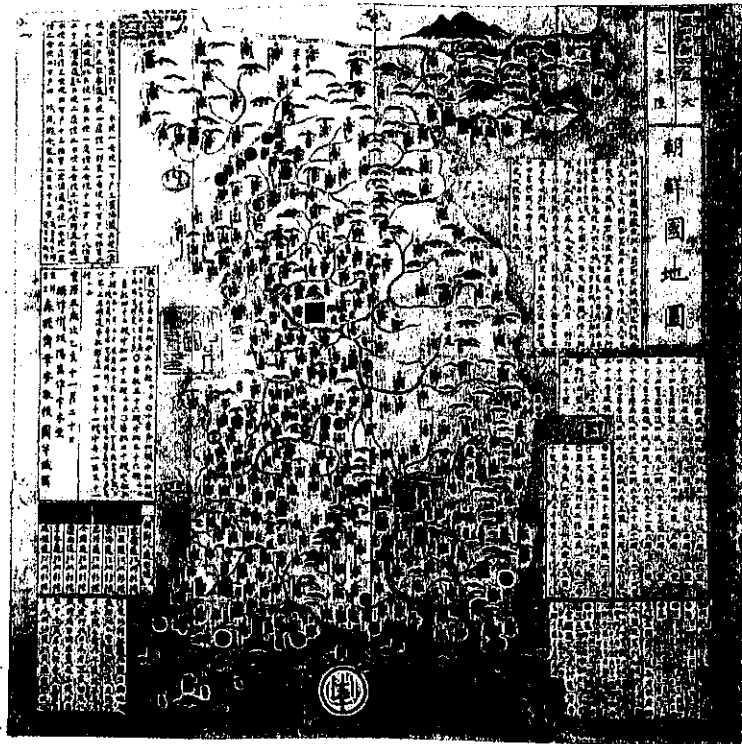


図 3.24: 亜細亞大洲之裏陸 朝鮮國地圖 宝暦 5 年 (1755 年) 国立公文書館

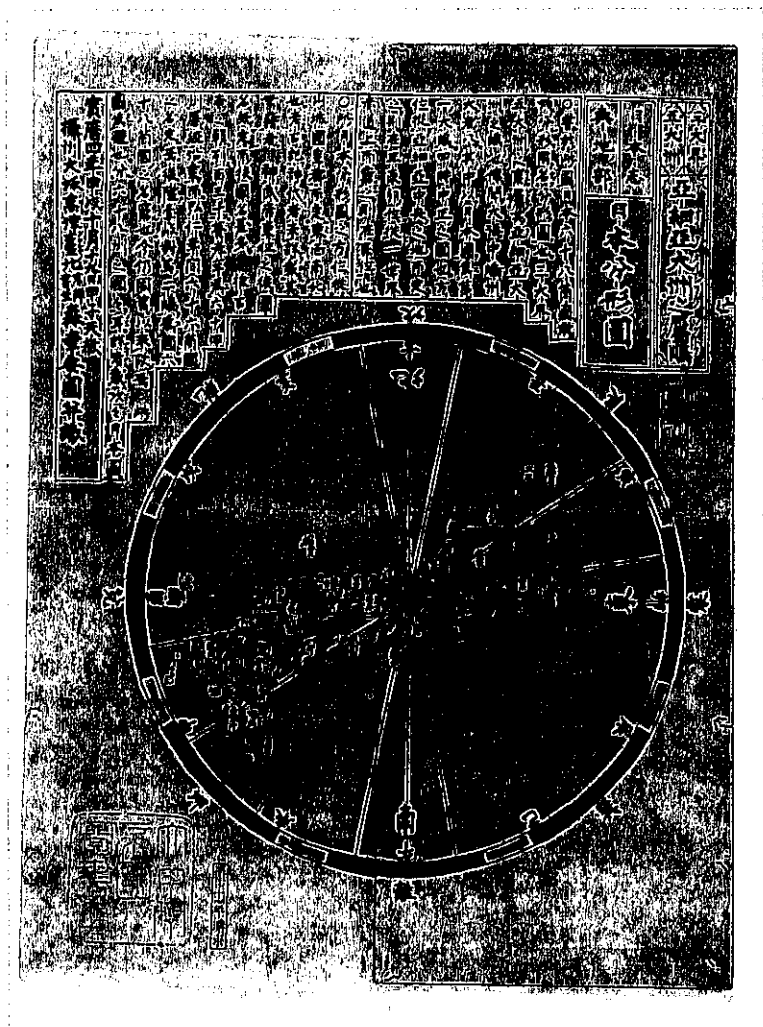


図 3.25: 三大界五大洲 亜細亞大洲之属嶋 日本志 與地部 日本分形圖 森幸安
宝曆 4 年 (1754 年) 国立公文書館

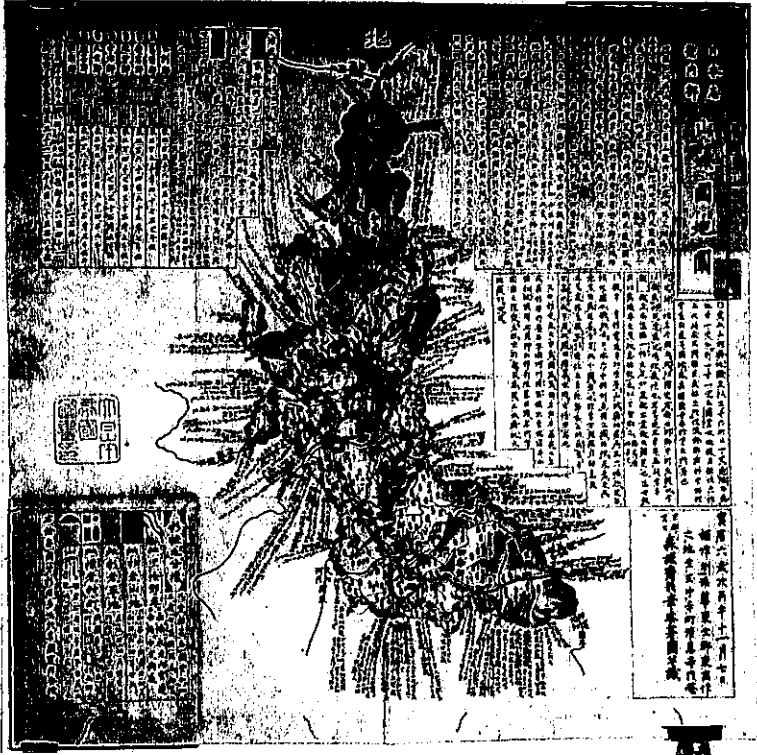


図 3.26: 日本志 畿内部 山城國地図 宝暦 6 年 (1756 年) 国立公文書館

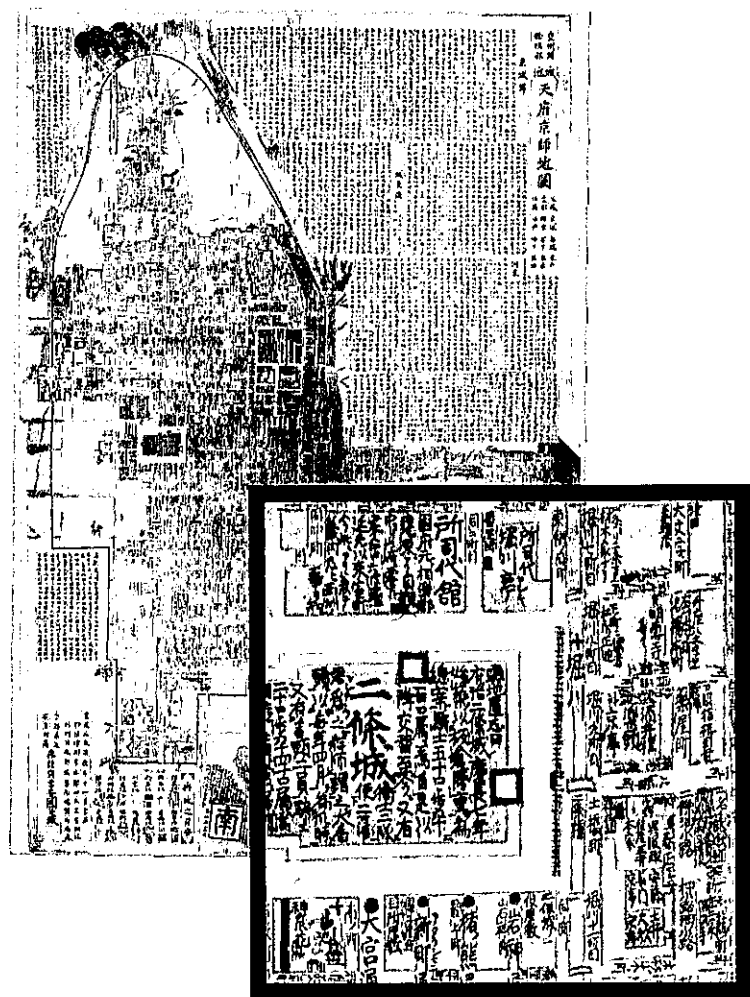


図 3.27: 皇州諸餘撰部 城池天府京師地図 寛延3年(1750年) 国立公文書館



図 3.28: 「The New Charleston」プロジェクトにおけるインスタレーションによる地図

各地の地図に続いて各地の神社仏閣の境内図へと入っていく。そこには地図の領域を超えて、いまここに居る自分の居場所についても、全球的な天度の一部として捉えることが出来ると言っているかのようである。この意味で言えば、森幸安の地図の概念は地球説を極限まで追求した結果であるとも考えられる。

しかしこれは現代の Google Earth [40] に見られるような完全に連続的な空間上に幅のない POI(Point Of Interest=関心地点) に対して説明を与えていくのではなく、そのズームインの過程でところどころ立ち止まり、識語という形で対応する空間区域に説明を与えるという地誌的作業が常に行われている。

3.6.4 非地球説の地図の数々

このように歴史的な中でも様々な地図の考え方がある。インターネット時代になってからも、人間の心の中には様々な地図があり、それを展開する人たちがいる。

図 3.28 は、ザ・ニュー・チャールストンと呼ばれるプロジェクトで、都市における 3 世紀以上に渡るアフリカ系アメリカ人の空間の歴史を俯瞰してみようとするプロジェクトで、アフリカ系アメリカ人にとって重要な歴史的な場所の

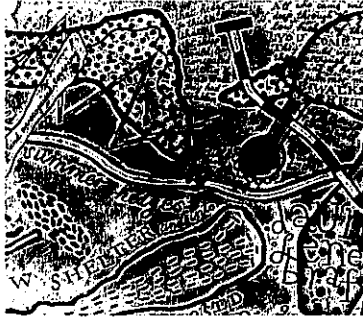


図 3.29: 1987 年コモン・グランドによる教会区地図づくりプロジェクト

詳細な地図を作成するという過程を通し、住人たちのアイデンティティーを復活させるプロジェクトである。

図 3.29 は、環境保全の運動家アンジェラ・キングと地理学者でありプランナーであるスー・クリフォードによって 1983 年にロンドンで設立されたコモン・グランドというプロジェクトであり、自然環境の発見と保護を場所に関する議論と関係づけて捉えてきた。人々にありふれた場所が持つ豊かさや日常生活の価値を思い起こしてもらふこと、自然に授けられた象徴性を感じ取ってもらふこと、場所に対して抱く感情的な関わりや場所が意味するものすべてを再評価してもらふことを通して、場所の維持管理に組むプロジェクトである。

ギー・エルンスト・ドゥボールが発表した、「心理地理学的パリ・ガイドー恋愛の情熱についてのディスクール」(Guide Psychogéographique de Paris: discours sur les passions de l'amour) [41](図 3.30) と「ネイキッド・シティ」(The Naked City)(1957 年)(図 3.31) は、都市を見る既存の視点を破壊する挑戦である。その後ギー・エルンスト・ドゥボールはシチュアニスト・インターナショナルという運動を展開する。街路から町を見てそれを自分なりに記録するこの方法は、様々な人々、運動に飛び火し現在に至っている(図 3.32)。

彼らシチュアニストたちはこれらの地図を心理地理学的地図と呼んでいる。彼らによれば、一般的に地図として知られている図面は、政治的、組織的、あるいは論理的、科学的に形成された都市のイメージであり、これらの既成的



図 3.30: Guide Psychogéographique de Paris: discours sur les passions de l'amour

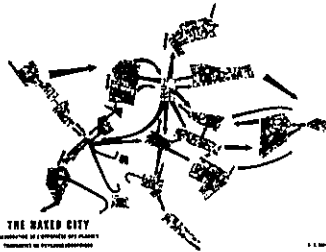


図 3.31: Naked city

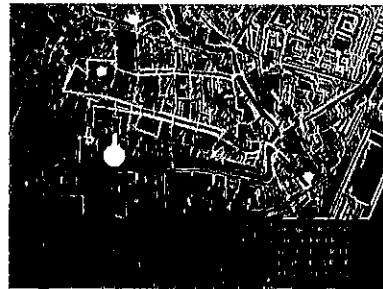


図 3.32: 現在あるなかで.... インターネットから

な空間イメージを個人の感覚だけをたよりに消し去り、新しい空間イメージを作り出す運動である。

このネイキッド・シティーは現在もいろいろな人々が実践している。言うならばフラッシュ・モブズとも言えよう [42]。フラッシュ・モブとは、FaceBookなどで呼びかけられ、駅前広場など特定の場所に集まった多数の人々によって繰り広げられるダンスなどのパフォーマンスである。一種デモ活動にも似ているが、政治的な意味合いを含まない場合が多い。ネットワークでつながることの、ある種の短期的な盛り上がり高揚感があるのだろう。これらも、駅前広場という公共空間の固定化された空間イメージを打ち砕くものである。

Googleの地図も、衛星写真も、このような数ある様々な地図のなかの一つである。これを図にしてみると、図 3.33 のようになる。かつては社会において様々なパラダイムの地図が流通していた。現代になるにつれて、測量図に重きが置かれるようになる。だが、それは他の地図が人間社会の中から無くなった訳ではない。幹の部分とその外側の色分けがなされただけだ。

ユビキタス・コンピューティングのメカニズムの中では幹の外側の地図は統一的ではなく流通しにくい。従って徐々に排除されて行く。ネットワークによって、あるいはビックデータによって多くの情報を一挙に結びつけることは、ある一つのことがフォーカスされ見えやすくなる。だが一方で多様性が犠牲となる。

3.7 ユビキタス・コンピューティングのゆくえ

コンピュータが無数に世界に広がり、人間とコンピュータが結びつき、更新し合う時代が今後促進されて行くとうなるのだろうか。いくつかの点で機械と生命の差が無くなる、のではないか。

まず、フラッシュモブズのような、あるいは、ネットワークで集められるビックデータから生まれる様々な知識である。こうして生まれる社会という生命の存在である。これも機械から生まれる生命と言ってよいのではないだろうか。「中国人のジム」が人工知能かという議論があった。まさに社会というのが「中国人のジム」である中国人のジムの構成員と全体は異なるという議論があった。我々も社会に翻弄されながら生きているが、私は社会の一員でありながら、その社会と私は異なる存在である。

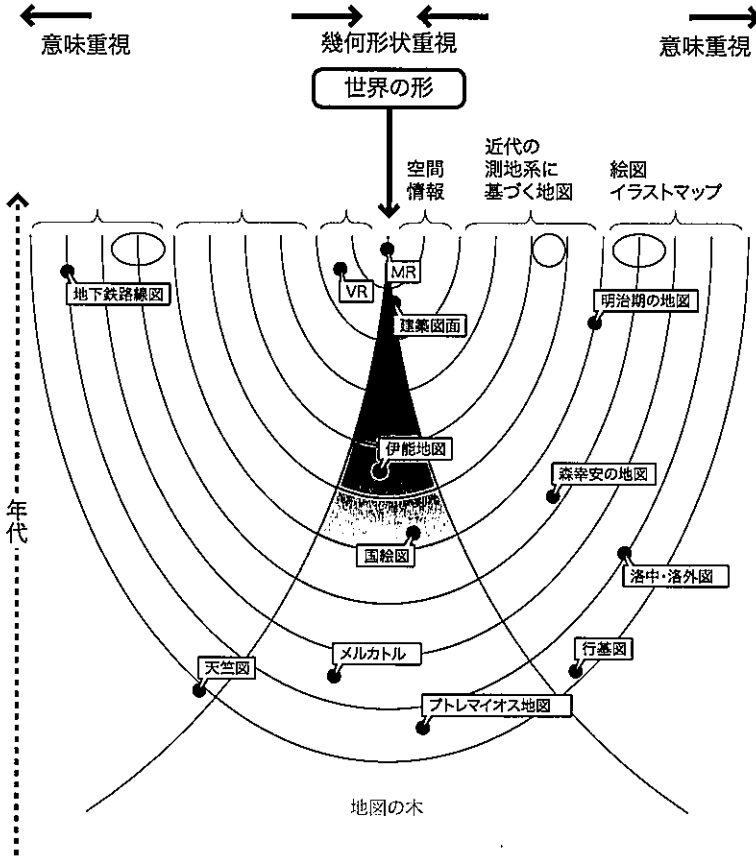


図 3.33: 地図の木

これは新しくて古い問題である。『湾岸戦争は起こらなかった』に見て取れるような、巨大なネットワークの中を流動する情報はやがて空虚なりアリティーを作り上げる。地球説の地図も同じだろう。ある種の画一化によって「中国人のジム」の構成員が忘れ去られてしまうのである。

次に、デバイスとして、現在我々はコンピュータと言えばシリコンチップで構成された電子的なシステムを想像する。だが、いま遺伝子とタンパク質から出来た細胞から構成される、生命由来のシステムとこのシリコンチップのハイブリッド型のデバイスが現れ始めている [43]。例えば、細胞の機能や、昆虫の嗅覚などを利用したセンサであったり、人間の神経に直接接続し、脳からの指令を直接動きに変える BMI(Brain Machine Interface) という技術により、新しい義肢や、重労働を補助するパワードスーツなどが開発されている。この細胞からなる部分とシリコンからなる部分、合わせた総体としては機械なのか生命なのか、区別する意味は無い。特に iPS 細胞などをつかった、再生医療が発達してくると、これが情報科学の世界に影響を与えない訳が無い。ある種のパーセプトロンを実際の神経細胞をつかって組織化し、ハイブリッドデバイスを構成するといった新しい形の人工知能も可能となってくるだろう。なにか倫理的な問題がありそうであるが、だが、どこまで良くてどこからがいけないのか、本当に線引きが出来るのだろうか。

第4章 連続と離散

4.1 乱雑性

ランダムというのは本当にランダムなのだろうか。いま私はここに0から9までの数字が出るルーレットを用意した。1から10までの数字が出るルーレットでもよい。出た結果から1を引き算すれば良い。このルーレットを使って、以下のような7つのランダムな数を得たところで、

1,6,4,8,9,0,6

Pi-Search (<http://www.angio.net/pi/bigpi.cgi>) というサイトを発見した。上記の数列が円周率 2 億桁の中で出現するかどうかを検索してくれるサイトだ。早速入力してみると、2,921,388 桁目から 7 桁が上記の数列に相当するという。その周囲の数は、

3.1415926535 61208857262385180467 1648906 80209319525601512169
...

円周率は誰が計算しても正しい計算結果は同一になるランダムではない数である。従ってこの7つの数はランダムではない。円周率は同じ数列が反復することのない無理数である。また、0から9までの数が、かなり均等に出現することによく知られている。となると、2億桁くらいあると7つの数の組み合わせが存在する確率が高い。十分大きな桁数を準備すれば、7つの数の組み合わせは確実に存在すると言える。つまり7つくらいの数をルーレットで得たとしてもそれは円周率の一部なのでランダムとはならない。ではルーレットで14個の数字を得た場合はどうだろうか。おそらく同じことが言えるだろう。

確かに、与えた任意の n 桁の数が円周率の何桁目に存在するか、ということ
を計算することは n が十分小さければ可能であるし、その結果は確定する。 n
がどんなに大きくても確定できるかどうかは証明されていない。しかしここで
「確定できる」と仮定し、かつ、数の出現が一様である¹と仮定するならば、 n
個の数の組み合わせが、円周率 N 桁中に出現する確率 $1 - \epsilon$ を計算すると、以
下のようなになる。

$$1 - \epsilon = 1 - \left(1 - \frac{1}{10^n}\right)^{N-n+1} \quad (4.1)$$

逆に、 n 個の与えられた数が確率 $1 - \epsilon$ で出現するためには、円周率が何桁
必要かと言えば、

$$N = \frac{\log \epsilon}{\log \left(1 - \frac{1}{10^n}\right)} \quad (4.2)$$

である。ある意味高い確率で出現する長さというものがある。

どうもおかしい。ランダムとはなんだろう。ランダムとは、どうも、ルー
レットなどで得られた数そのもののことをさすのではなさそうだ。むしろ「ルー
レットなどで数を得る」行為のことを言っているのではないだろうか。そう考
えたとしてもまだ懸念が残る。「ルーレット」だ。ルーレットがアルゴリズムに
よって動いていたならば、それはランダムとは言えない。ルーレットがアルゴ
リズムで動いていたとするならば、長いこと使い続けていると、あるところか
ら最初と同じ数列が現れてくる。実際にそのようなアルゴリズムは存在する。
疑似乱数である。ではルーレットの動作がアルゴリズムでなければその出力は
乱数となるだろうか。

ここで読者は 2.3.9 節に登場した円周率の魔法のランプを思い出すだろう。そ
のときのおさらいをすると、円周率の 2 億桁、あるいは 1 京桁にしろ、途方も
なく大きな桁にしても、それが有限桁の近似値を計算するものであれば、アル
ゴリズムは存在する。しかし、円周率全体を計算する、あるいは計算し続ける
機械は、有限時間内に停止することが出来ないのでアルゴリズムではない。実
際そんな機械を我々は作ることが出来ない。だが、円周率の計算であるからに

¹注意深く、ランダムである、もしくは、乱雑である。とは言っていない。

は、そこには計算式が存在する。

では、全く法則性のない数列を生成し続ける機械、つまり真のルーレットと
いうのはこの世に存在するのであろうか。物理乱数生成器という電子回路がある。
ダイオードの熱変動など、量子力学的な確率過程を利用し、乱数を生成する
回路である。おそらくこの回路が無限に動けば、円周率のどの部分にも相当
しない数列が得られるはずである。だが、所詮は電子回路である。いつかは壊
れて動かなくなる。壊れぬまでも、お役御免になり捨てられる運命にある。つ
まりその時点で生成できた数列は有限であり、その時点で円周率の途方もない
数列のどこかに取り込まれてしまう。

無限に動き続ける物理乱数生成器、そのような数学的空想世界の機械こそが
真の意味でのランダムなのだ。だが我々有限世界の住民としては、真の数学的
ランダムが必要になることはないと言言できる。せいぜい、耐用年数を過ぎれ
ばお役御免の物理乱数生成器で十分である。物理乱数生成器もあまり出番はな
い。一番使いやすいのはアルゴリズムで生成できる疑似乱数である。疑似乱数
の使いやすさはなんとといっても、はじめからやり直すと同じ数列が出てくるこ
とである。乱数を使ったシステムを開発仕様とした場合、悩ましいことは、同
じ状態が再現できないと、おこるバグもおこらなかつたりする。物理乱数生成
器をつかった暁には、たまたま発見したバグが次に再現するのは数百年後、と
いった場合もあり得る。疑似乱数を使っていれば、システムをリセットすれば
同じ状況が再現できる。これは便利。

すこし見方を変えよう。最後に私がぼやいたように、実際には疑似乱数がラン
ダムの代替として利用価値がある。また円周率も大変質のよい乱数として使
われることがある。これで十分ことが足りる場合が多いということはどういう
ことであろうか。円周率を乱数として使えるのは0から9までの各数の出現頻
度が一様であるということだけではない。ある桁のあと続く、その次の桁がな
んであるかを予測するのが大変難しいという点にある。つまり次の桁が「わか
らない」ということなのだ。五分五分という言葉がある。つまり二つの起こり
得る事象が確率的に0.5ずつである、という意味であるが、等確率ということ
はなにがおきるか「分からない」ということなのだ。

「分からない」ということは主観的な言葉だ。量子力学的な確率過程であ
れば、だれも予測できる人はいない。その意味で「分からない」ことである。し

かし多くのことは誰かが知っているがある人には「分からない」ことであつたり、あるいは調べれば分かることであるが、調べていないから「分からない」ことだったりする。円周率を計算する力量やその時間のある人にとっては次の桁がなにかということは分かりきったことである。しかし、そのような力量や時間がない場合、次にくる数字は「分からない」、分かりきった人から与えられたときに初めて「分かる」のである。

確率は、事情はどうであれ「分からない」状況をいかにモデル化するか、ということである。完全にランダム＝法則性のない数列、つまり「分からない」数列なのである。先に述べたランダムのパラドックスは、言い換えれば、「分からない」ことを「分かったように」語るからおきるパラドックスである。

4.2 情報エントロピー

情報理論ではシャノンの情報エントロピーの話がしばしばお目見えする。熱力学エントロピーとの間で性質がよく似ているという点から、物理学的な要素を情報理論に持ち込もうとする理論家も少なくない。また、社会的なコミュニケーションを論ずるときに情報理論的な発想を持ち込むこともしばしばである。だが、それらの多くを見ていると、ある種の混乱がおきている。

典型的な例として「あるいは外交は情報戦だ。情報量が多いほどイニシアチブを握ることが出来る。」と言われる。一方で、「現代における情報社会では情報が氾濫し、ひとつひとつの情報の価値が下がっている。」とも言われる。一体、情報量が多いほどいいのか、少ない方がいいのか....。「情報エントロピー」は別名「情報量」とも呼ばれる。まずはこの二つの意味の差がくせ者である。

「情報エントロピー」の概念を始めて示したのはシャノンである。しかしシャノンはそれを「情報量」と定義していた。シャノンの論文を読み「情報エントロピー」という名が良いのではないかという提案は、フォン・ノイマンだったと言われる。こうして「情報量」と「情報エントロピー」の混同が始まった。まずは、私なりに「情報エントロピー」を整理してみよう。

「カラスは黒い」という文章はまるでからすが黒いことが正しいかのようなイメージを喚起する。だが厳密に言えば（しかし、当たり前なことでもあるが）、この文章（＝コードとしての情報）が間違っている可能性も含んでいる。「カ

「カラスは黒い」という命題をもう少し詳しく吟味するならば、背後に「カラスは何色だ?」という疑問文があり、これに対する解答として「カラスは黒い」というコードが提示されている。この疑問文に対する答えとしては、「カラスは白い」「カラスは赤い」「カラスは緑だ」...といった同種の異なる命題が想定され、その想定される幾種類かの命題の中で「カラスは黒い」のみが正しいということを主張しているのが「カラスは黒い」という文章なのだ。「疑問文に対する解答の可能性」あるいは、「疑問文に対する解答=情報の揺らぎ」が情報エントロピーである。

解答の可能性、「カラスは黒い」「カラスは白い」「カラスは赤い」「カラスは緑だ」という4種のパターンが想定されたとしよう。この4種のうちどれが優位か、あなたの持つ知識からは決め難い場合、それぞれの命題の確からしさは $1/4$ である。

からすが何色か、あなたが友達にメールで解答を報告する必要があった場合、上記4つの可能性を考えて、何ビットを送信することが可能なメールがあれば、報告に備えることが出来るだろうか。答えは2ビットである。

「カラスは黒い」 = 00

「カラスは白い」 = 01

「カラスは赤い」 = 10

「カラスは緑だ」 = 11

という4つのコードを友達と申し合わせておけば、カラスが実際に何色だったかを報告するには十分である。このときの2ビットというのがこの場合の情報エントロピーである。

もし、あなたが事前にカラスの色についてもうすこし情報を持ち合わせており、どうも「カラスは黒い」というのがおおよそ9割がた確からしいと分かっていたとしよう。他の3つの命題の可能性については、 $1/30$ の確率としよう。このような場合、コーディングのビット長に偏りを入れて、こんなコーディングはどうであろう。

「カラスは黒い」 = 0

「カラスは白い」 = 100

「カラスは赤い」 = 101

「カラスは緑だ」 = 110

「カラスは黒い」という可能性が高いのだから、メールで用意するビット長はさつきより短くなる可能性が高い。まあ、運悪く3ビットになってしまう可能性もあるが....。では平均的に期待できるビット長（ビット長期待値）はどれくらいであろうか。

$$1.2bit = \frac{9}{10} \times 1bit + \frac{1}{30} \times 3bit + \frac{1}{30} \times 3bit + \frac{1}{30} \times 3bit \quad (4.3)$$

となる。この場合の情報エントロピーは1.2bitということになる。一つ確度の高い命題があると情報エントロピーは減少することがわかる。別の言い方をすれば情報としての揺らぎが減少する。当然のことながら、最初から「カラスは黒い」と分かっているならば、メールを送る必然性はない。つまり情報エントロピーは0である。（=全く揺らぎが無い。）

ここで、情報エントロピーをもう少し形式的に定義してみよう。ある疑問文 $P = \{p, p_n\}$ とは、ひとつの疑問 p に対する解答を表す N 個の命題 $p_n (n = 0 \dots N-1)$ のことであり、それぞれの確からしさを p_n とする。そのとき各解答に割り振るコードのコード量を $-p_n \log_2 p_n$ とし、情報エントロピーは

$$H(p_n) = - \sum_{n=0}^{N-1} p_n \log_2 p_n \quad \text{ただし、} \quad \sum_{n=0}^{N-1} p_n = 1 \quad (4.4)$$

と定義できる。この定義に従って、2番目のカラスの例を計算し直すと、 $H = 0.627491844bit$ となる。

このように「情報エントロピー」はあくまでも疑問文とそれに対する解答のヴァリエーション（揺らぎ）の対のなかで意味を持つてくる。これに対して、「カラスは黒い」という「文章」は「カ」「ラ」「ス」「は」「黒」「い」とう6文字のコードである。これは解答の一つを意味しているかもしれないし、あるいは、確定した答えを意味しているかもしれないが、とにかく、コード量=6文字の文章である。

この私の「情報エントロピー」は一見シャノンの情報エントロピーとかけ離れているようにも見える。次に、実はそうではないことを示そう。同時に、シャノンの「情報量」がこの「情報エントロピー」と「コード量」を混同していることも示される。

シャノンの情報理論では転送されるコードの圧縮を目的としている。たとえば画像を転送する場合を考えよう。現代の情報システムでは画像は、長方形の全体をピクセルとよぶ小さな区画からなるグリッドに切り刻む。各ピクセルに相当する色彩を三原色に分解し、各色 8bit の合計 24bit の「色彩コード」でコーディングする。たとえば、一つの画像を 1000×1000 の格子に分解したとすれば、合計 $S = 24,000,000 \text{ bit} = 1000 \times 1000 \times 24 \text{ bit}$ の「コード量」が必要である。シャノンはいくつかの圧縮できると主張した。

どんな画像もそれぞれの色が同一の割合で含まれている訳ではない。使われていない色もあればよく使われている色もある。色彩コード n の色が使われている割合を p_n とすれば、 n の色を 24bit ではなく、 $-p_n \log_2 p_n \text{ bit}$ にコーディングすれば、全体のコード量を

$$S_{\text{after}} = -S \sum_{n=0}^{N-1} p_n \log_2 p_n \quad \text{ただし、} \quad \sum_{n=0}^{N-1} p_n = 1 \quad (4.5)$$

とすることが出来る。一般に $S_{\text{after}} \leq S$ である。 S_{after} は式 (4.4) に画像の元のコード量 S を掛けたものである。シャノンはこれを元の画像の「情報量」と呼んだ。だがこの場合、画像全体については、最初から与えられたものであるから、これ自体に揺らぎはない。つまり画像自体に情報エントロピーはなく、コード量があるのみであると考えべきである。乱雑性を含んでいるのはピクセルである。任意のピクセルを選び、「このピクセルは何色か」という問いに対する解答の揺らぎ、つまりピクセルの情報エントロピーは式 (4.4) となる。これでシャノンの言う情報エントロピーはこれの S 倍であり、私の定義した情報エントロピーに相関があることが示された。

4.3 熱力学エントロピー

ここまでの議論をふまえて、熱力学エントロピーと情報エントロピーの違いについて述べておこう。熱力学エントロピーは情報エントロピーの一種である。ある分子の集団に対して「注目する分子の持っているエネルギーはいかほどか」という疑問に対する答えのバリエーションがつまり熱力学エントロピーである。よく、熱力学第二法則を引き合いに出し、情報エントロピーも全体で増大し続けるのではないかという言及がなされたりする。しかし、熱力学エントロピーは情報エントロピーの特殊な場合であるので、熱力学エントロピーに成り立つことを情報エントロピーに拡大することは出来ない。

そもそも熱力学第二法則が成り立つためには、エントロピーという概念の一方でエネルギーという概念があることが前提条件である。そのエネルギーは熱力学第一法則によって保存されていることが保証されている。さらに、熱エネルギーは二つ以上の熱源が無いと、他のエネルギーに変換することが出来ないというクラジウス・トンプソンの原理があつて初めて熱力学第二法則が示される。

2010年に中央大学と東京大学のチームが、マックスウェルの悪魔を実験によって実現したという報道があつた²。報道によれば、マックスウェルの悪魔が、分子の情報を得ることによって、一方の熱源からもう一方の熱源へ熱エネルギーをポンプアップすることが出来る。これによって熱力学エントロピーは下がる。しかし、マックスウェルの悪魔が情報を消去するときに、情報エントロピーが増大し相殺される、ということで、熱力学エントロピーと情報エントロピーは一見同じものであるように見えるが、この情報エントロピーはあらゆる情報エントロピーではなく、「注目する分子の持っているエネルギーはいかほどか」という情報であり、熱力学エントロピーである。つまり、熱力学エントロピーの問題を情報エントロピー全般に拡大できないということには変わりない。

4.4 情報を集めると情報エントロピーは下がるか

最初に挙げた二つの問題に戻ろう。「あるいは外交は情報戦だ。情報量が多いほどイニシアチブを握ることが出来る。」というのはどういう意味だろうか。

²<http://news.mynavi.jp/news/2010/11/19/113/>

外交官は確かな情報を集めなければならない。確かだと分かっている情報を出来るだけ多く集めれば、情報戦に勝てる。つまり確かな情報の「コード量」のことを言っている。「現代における情報社会では情報が氾濫し、ひとつひとつの情報の価値が下がっている。」はどうだろうか。「京都の美味しいイタめし屋はどこか」探そうとして、ぐるなびを引けばレストラン A がヒットし、友達に聞くとレストラン B だと言う。さらには、町歩きガイドブックではレストラン C... ということであるから、情報を集めれば集めるほど「情報エントロピー」が増大して行くことに悩める文章である。

「コード量」はとてもシンプルな性質をもっている。情報 A のコード量を $S(A)$ として情報 B のコード量を $S(B)$ としよう。情報 A と B をあわせたときのコード量は単純な加算 $S(A) + S(B)$ である。だが、「情報エントロピー」はそうはいかない。情報 A の情報エントロピーを $H(A)$ としよう。情報 B は情報 A のコピーであったとする。すると情報 B 単独ではこの情報エントロピーは情報 A と同じ $H(A)$ であるはずである。このとき、情報 A と情報 B をあわせても情報エントロピーは倍にも半分にもならず、 $H(A)$ のままである。情報 B は情報 A があればコピーとして確定した情報である。つまり情報 A の元では、情報エントロピー 0 なのである。これは情報 A, B のそれぞれの確率過程が相互に依存しているかどうかということに起因する。依存の如何によって情報エントロピーは複雑に変化する。

複数の情報をあわせた場合のエントロピーがどうなるか、という議論をするためには、情報をあわせるということはどういうことかを考えなければならない。全く依存しない疑問文とその解答、たとえば $P = \{「カラスは何色か」, p_n\}_{0 \leq n \leq N}$ と疑問文と、あわせて、 $Q = \{「鳩は何色か」, q_n\}_{0 \leq n \leq M}$ という疑問文に両方答える場合を考えよう。それぞれの答えは全く関係ないので、二つの答えの組み合わせパターンは NM 個である。そしてその情報エントロピーはそのまま両者を加算した、以下となる³。

$$H(\{p_i\} + \{q_j\}) = H(\{p_i\}) + H(\{q_j\}) \quad (4.9)$$

³詳しくは以下のような展開となる。

$$H(\{p_i\} + \{q_j\}) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} p_i q_j \log_2 p_i q_j \quad (4.6)$$

次に考えられる「あわせ」として、同じ疑問文に対する、異なった揺らぎをもった解答である。先のイタめしレストランの例で考えてみよう。レストランはA,B,Cの3店あるとして、ぐるなびの答え、友達への答え、町歩きガイドブックの答えと3つの答えがある、ということである。だがこれだけでは、情報をあわせることは出来ない。これに対してもう一つ疑問文が必要である。つまり「どの答えが正しいか」という疑問文である。その答えとしては、「ぐるなびの答えが正しい」「友達への答えが正しい」「町歩きガイドブックの答えが正しい」という3つがあることは言うまでもない。この合計4つの疑問文と答えをあわせることによって、あなた自身の情報が得られる。

話を簡単にするために、同じ疑問文 π による2つの情報 $P = \{\pi, p_n\}$ と $Q = \{\pi, q_n\}$ があり、どちらの情報か正しいかの判断を $R = \{\rho, r, 1-r\}$ としよう。判断の結果を、 $P \oplus Q|_R = \{\pi + \rho, r p_n + (1-r) q_n\}$ とあらわすことが出来る。このときのエントロピーの関係は、

$$\min\{H(P), H(Q)\} \leq H(P \oplus Q|_R) \quad (4.10)$$

である⁴。

$$= \sum_{i=0}^{M-1} p_i \left(\sum_{j=0}^{N-1} q_j \right) \log_2 p_i + \sum_{j=0}^{N-1} q_j \left(\sum_{i=0}^{M-1} p_i \right) \log_2 q_j \quad (4.7)$$

$$= H(\{p_i\}) + H(\{q_j\}) \quad (4.8)$$

⁴この導出は少々ややこしい。P, Q, RをそれぞれN次元空間上の点と考えると、RはPとQを結ぶ直線上、 $1-r : r$ に分割する点である。Rは、 $r=0$ のときQと、 $r=1$ のときPと一致する。 r を0から1まで変化させる時、情報エントロピー $h = H(P \oplus Q|_R)$ は r の関数となっている。この関数は、(1) 右上がり、(2) 右下がり、あるいは、(3) 1か所最大値をもつかのいずれかである。(1)~(3)をグラフ化してみれば分かることであるが、式(4.10)が証明される。

r の関数 h が、(1)~(3)のいずれかであることを示すには、 h の導関数を求めればよい。

$$\begin{aligned} h'(r) &= - \sum_{n=0}^{N-1} \frac{d \{(p_n - q_n)r + q_n\} \log_2 \{(p_n - q_n)r + q_n\}}{dr} \\ &= - \sum_{n=0}^{N-1} (p_n - q_n) \log_2 \{(p_n - q_n)r + q_n\} \end{aligned} \quad (4.11)$$

この式の各項の構造を見てみよう。

$p_n - q_n = 0$ の場合の項 項全体も0である。

$p_n - q_n > 0$ の場合の項 r の変化に対して右上がりとなる。

この式は、 P, Q から判断する限り、双方の情報よりも確実な情報は得られないと言っている。また、 $H(P \oplus Q | R)$ が $H(P)$ と一致するのは、 $r = 1$ のとき、 $H(Q)$ と一致するのは、 $r = 0$ のときである。これは、 P, Q のより確実な方を一つ積極的に選び他方を捨てる、という判断が最適であることを言っている。 R という最終判断が、情報エントロピーを増大させるか否かということは、大組織の決められない会議を思わせる結果である。

4.5 推論

このようにしてみると、ただ情報を集めてくるだけでは情報エントロピーは減少しない様子である。熱力学におけるエントロピーとは大きく異なるが、ある意味情報理論におけるエントロピー増大の法則である。だがそうになると、「外交は情報戦だ。情報量が多いほどイニシアチブを握ることが出来る。」ということはどういうことなのか、あらためて疑問が生じる。

情報を集めると情報エントロピーが増大して行くなれば、不要な情報を捨て去るしか無い。実はこれこそが推論の根幹をなす概念である。たとえば「 A ならば B かつ B ならば C 、ゆえに、 A ならば C 」という古典的三段論法を考えてみよう。これは、2つの情報「 A ならば B 」と「 B ならば C 」から1つの情報「 A ならば C 」が導出できると主張している。三つの命題 A, B, C が提示されているため、三段論法を考えなければ、以下のような三つの疑問文が独立に考えられることになる。

1. A は正しいか

$p_n - q_n < 0$ の場合の項 \log_2 の中身は、 q_n から p_n にかけて右下がりとなり、 $\log_2 \{(p_n - q_n)r + q_n\}$ 全体としても右下がりとなるが、定数が負であるために、項全体としては右上がりとなる。

従って、0 以外の項はすべて右上がりであり、式 (4.11) 全体も右上がりとなる。ということは、 $h'(0) \geq 0$ ならば、 h' は $r \in (0, 1)$ のすべてにおいて正であり、 h は右上がりの関数となる。 $h'(0) < 0$ かつ $h'(1) \leq 0$ のとき、 h' は $r \in (0, 1)$ のすべてにおいて負であり、右下がりの関数となる。

$h'(0) < 0$ かつ $h'(1) > 0$ のとき、ある一か所 $r_0 \in (0, 1)$ において、 $h'(r_0) = 0$ なる点が存在する。 h はここを最大とした関数となる。

h の形状は以上ですべてである。

2. B は正しいか

3. C は正しいか

それぞれ正しい確立が $1/2$ とするならば (いま A, B, C に関して全く情報がないので、こう判断するのが妥当だろう。) 一つ一つの情報エントロピーは 1bit で、この3つの合計の情報エントロピーは 3bit である。ここで、一つ新しい情報として「 A ならば B 」という情報が得られたとしよう。問題を簡単にするために、この情報は確実である (情報エントロピー 0) と考える。すると、 B の確からしさは A に制約されるわけであるから、2番目の疑問「 B は正しいか」のエントロピーが変化するはずである。その場合の数を数えてみよう。

「 A ならば B 」という推論は、 A が正しい場合は B も正しい、ということである。 A が正しい確率は $1/2$ なので、この場合の B が正しい確率も $1/2$ である。一方、 A が間違っている場合について「 A ならば B 」という推論は、なにも言及していない。いわば話を避けている。つまり、 A が間違っている場合 B の正しい確率は五分五分なのだ、 A が間違っている確率が $1/2$ でさらに B が正しい確率はさらにその半分で、あわせて $1/4$ となる。 A が正しい場合とあわせると、合計確率は $3/4$ ということで、最初の $1/2$ より確率がやや大きくなる。「 B は正しいか」の情報エントロピーは、 0.81^5 となる。3つの問いの合計の情報エントロピーは、 2.81 bit であると言いたいところであるが、「 A は正しいか」と「 B は正しいか」という二つの問いの相関をもう少し考慮しなければならぬ。表 4.1(a) は今の議論を表にしたもので、 A, B の考えられる正誤の4パターンがどのような確率で起こり得るかを表している。たとえば、3行目は A が正しく、 B が間違っているということはある得ない。つまり確率 0 であると考えられる。 A が正しい確率は合計で $1/2$ であるが、その場合すべて B は正しいわけで最後の行が確率 $1/2$ となる。これによれば、「 A は正しいか、および、 B は正しいか」この問いに対する情報エントロピーは、

$$1.5\text{ bit} = -\frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) \quad (4.12)$$

⁵ $-\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)$

表 4.1: 命題のあいだの相関と確率

(a) A と B の場合

A	B	確率
誤	誤	1/4
誤	正	1/4
正	誤	0
正	正	1/2

(b) A と B および C の場合

A	B	C	確率
誤	誤	誤	1/8
誤	誤	正	1/8
誤	正	誤	0
誤	正	正	1/4
正	誤	誤	0
正	誤	正	0
正	正	誤	0
正	正	正	1/2

と計算される。まだ、 A や B と相関はまだ考えていない C の分も合わせたものは 2.5 となる。

さらにもう一つの推論「 B ならば C 」を導入してみよう。これも確実な推論と仮定しよう。 B が正しいかという疑問の確からしさ $3/4$ から、 C の確からしさを計算すると、 $7/8$ となる。表 4.1(b) のように相関をまとめると、全体の情報エントロピーは、

$$1.75 \text{ bit} = -\frac{1}{8} \log_2 \left(\frac{1}{8} \right) - \frac{1}{8} \log_2 \left(\frac{1}{8} \right) - \frac{1}{4} \log_2 \left(\frac{1}{4} \right) - \frac{1}{2} \log_2 \left(\frac{1}{2} \right) \quad (4.13)$$

となる。推論による関連性を増やして行くにつれて、情報エントロピーは、3 bit, 2.5 bit, 1.75 bit というように確かに押さえられていく。集めた情報がバラバラでは意味をなさない。それらを関連づけることによって、一方の情報で他方の情報に制約を与えることが推論である。制約が与えられれば、それだけ情報／答えの揺らぎが現象し、情報エントロピーが小さくなる。

ここで興味深いことがある。このような推論をいくら繰り返しても、情報エ

ントロピーは1より小さくなることはない。上記の例で言うならば、推論の根っこにあたる「Aは正しいか」という問いに対する答えにはいつまでたっても答えることが出来ない。つまりAの確からしきはいつまでたっても五分五分である。推論の根っこには必ず公理が存在し、これは信じるしかない。数学、物理学、あらゆる学問がどんなに論理的であっても、超えることのできない情報エントロピーである。

4.6 平均化

推論により情報エントロピーを圧縮する方法を見てきた。しかし、それ以外にも情報エントロピーを圧縮する方法がある。これから代表的な方法の一つずつみていこう。まず最初が平均化という方法である。

平均化は最もなじみの深い方法ではないだろうか。言ってみれば「木を見ずに森を見る」ということである。一本一本の木を見ると千差万別であるが、森全体を見るとある傾向が見えてくるということである。そもそも我々がいま目の前にしているあらゆる物質は千差万別に振る舞う原子や分子の平均的な振る舞いである。原子や分子が無秩序な運動をしているからといって、我々が見ている日常世界が無秩序なわけではない。目の前の様々なものが「そのもの」として認識できるのはこの平均化の原理によるものである。

平均化をもっと詳しく見て行こう。例として、あまり信用のない人々に対するローンにおいて、いかに確実にお金を回収するかという話をしよう。お金を貸し付けた場合、10年後に返済してくれる確率が70%という人を仮定しよう。同じ人が10人いれば10回に3回は貸し倒れるという意味であり、信頼度70%の人という言い方をしよう。さて、この人に「10年後は2倍にして返してほしい」と言って、100万円をあなたは貸せるでしょうか。結構リスクである。しかし、もし、同じ信頼度の人々が2人いて、それぞれに50万円合計100万円貸すと想定した場合はどうだろうか(表4.2)。

完全に損するのは1割に満たない。おおよそ貸したお金がそのまま戻ってくる。あわよくば49%の確率で(これもそんなに悪くない値だ。)儲かる。この数値を見てあなたも少し心が揺らぐのではないだろうか。1人より2人の平均的な振る舞いの方が全体的な信頼度が上がるという効果である。91%の確

表 4.2: 二人へ貸した場合の返済額期待値

	返済額	確率	計算方法
二人ともに踏み倒される場合	0円	9%	$((1 - 0.7) \cdot (1 - 0.7))$
どちらか一方に踏み倒される場合	100万円	42%	$(2 \cdot 0.7 \cdot (1 - 0.7))$
どちらも返済する場合	200万円	49%	$(0.7 \cdot 0.7)$

率で100万円以上が戻ってくる。このとき「100万円以上」というものを91%信頼区間と言う。50%信頼区間といえば、「200万円以上」ということになる。信頼区間は狭い方が良いが、確率を上げ、より確実にしようとする信頼区間は広くなる。95%の信頼性を確保しようと思うと、一円も戻ってこない可能性が大きい、つまり、2人ではたりないことが分かる。3人では、4人では...と人数を増やして行くと現実的な、95%信頼区間が確保できるようになっていくのだろうか。

図 4.1 は貸し付け対象人数を、3人、10人、100人、1000人と増やして行ったときに、貸付額の何倍(横軸)がどれくらいの確率で返ってくるかをグラフにしたものである。このグラフにより、95%信頼区間を計算すると、3人のときは、0.6倍以上、10人のときは1倍以上、100人のときは、1.24倍以上、1000人のときは、1.352倍以上となる。グラフは人数が増えるにつれて、なだらかな形から急順に切り立った山型になっていく。原理的には、期待値の1.4倍に近づいていく。こうして見ると、100人、1000人では、ほぼ確実に儲けができることになる。

ローンの考え方は、やや信頼のおけない対象であっても多くの人に貸付けることによって利益を出すことができるというものである。個々の人々の振る舞いとその平均値の振る舞いは異なるということでもある。それをうまく利用すれば、安定した結果が得られる。この考え方はローンだけでなく、多くのウェブサイトの集客や購買行動パターン分析などに利用される。いわゆるビッグデータと呼ばれる情報処理の一種である。ここで重要となってくるのは、「一人

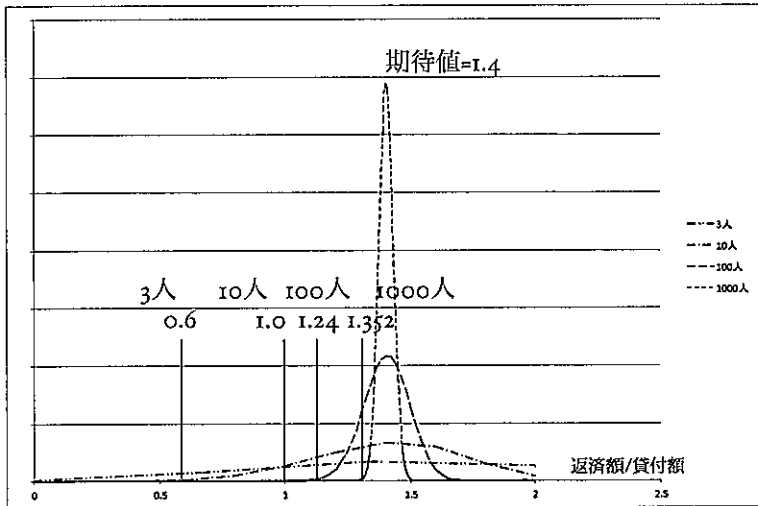


図 4.1: 対象人数と 95% 信頼区間

一人の信頼度が 70%」というのを確実に保証しなければならないということである。貸付け時の審査をおろそかにし、信頼度の査定が狂い始めると、いわばサブプライムローン問題となるのである。

4.7 共鳴・量子飛躍

通常、乱雑な世界では、生命は安定的に存在し得ないと考えるだろう。しかし乱雑な世界であっても、先の平均化という現象を考えれば、つまり、乱雑な無数の原子や分子は、ある一定の「平均的な」物理法則によって秩序ある運動をする。たとえば、どんな物質も一定の温度を保っている。あるいは、無数の分子が集まった個体の重心はひとつで、その運動はニュートン力学に従う。だが、それだけでは、秩序は長続きしない。たとえば、振り子のような一定の秩序だった運動するおもりも、やがて空気の抵抗によって停止してしまう。運動エネルギーは空気の乱雑な分子運動の中にかき消されて行く。これでは生命も長時間の秩序を維持して生きることなどできないはずだ。進化などまして、お

や、である。乱雑な分子運動の蔓延するこの世においては、秩序だった運動など、一過性の現象に過ぎない。

ところが、世の中には実は長時間秩序を保ち続ける現象が存在する。生命もその一つだが、原子も一つである。原子は、中心に陽子と中性子からなる原子核、この周りを電子が回っている。電子と陽子は電荷をもち、引き合っている。電子は回り続ける。しかしそのままであれば、周囲からの電磁気的な攪乱によって、電子はやがて軌道を外れ、原子核に落ち込んでしまうか、あるいは原子から飛び出してしまうはずで、物質は崩壊する。しかし、実際には違う。自然界に存在する原子の安定度は確実なものである。原子の安定性、また、原子の組み合わせによる分子の安定性はどこからくるのであろうか。平均化の原理や古典物理学では説明できない。まして分子をベースとして成り立っており、さらには「高機能」な仕組みを限界まで小さい「個体」に詰め込んだ生命が安定的に存在することを説明することは出来ない。

シュレディンガーは「生命とは何か」の中で、量子飛躍によってこれが説明できると言っている。さきの原子内の電子の例で示すならば、次のようになる。電子のエネルギーは連続的に変化することが出来ず、とびとびな値しか持ち得ない。あるエネルギーを持っている電子がエネルギーを得ようにも、一定以下のエネルギー量では、電子の状態は変化し得ず、ある一定以上のエネルギーを与えないと、エネルギーが増えることが出来ない。電子のエネルギーを変化させることのできるこの最小値が原子の崩壊を防ぐバリアの役割を果たす。このとびとびのエネルギーが量子飛躍である。同様な原理が分子の安定性にも寄与している。当然のことながら、生命を構成する遺伝子やタンパク質といった高分子がなぜ高分子ながら存在し得るのかといえば、この量子飛躍があるからだというわけである。

量子飛躍を情報エントロピーとして解釈するならば、「電子のもつエネルギーはどれくらいか」という問いに対する答えの揺らぎとして解釈できる。エネルギーに関する問いなので、これは熱力学エントロピーでもある。もし、古典力学的に解釈するならば電子のエネルギーの取りうる値は、0以上の実数で連続的に無数に存在する。これに対して、エネルギーがとびとびであるならば、しかも、原子核が電子を捕捉できる最大エネルギーがあるので、電子のエネルギーの取りうる値の場合の数は有限個でしかない。答えの揺らぎがこれほどまでに

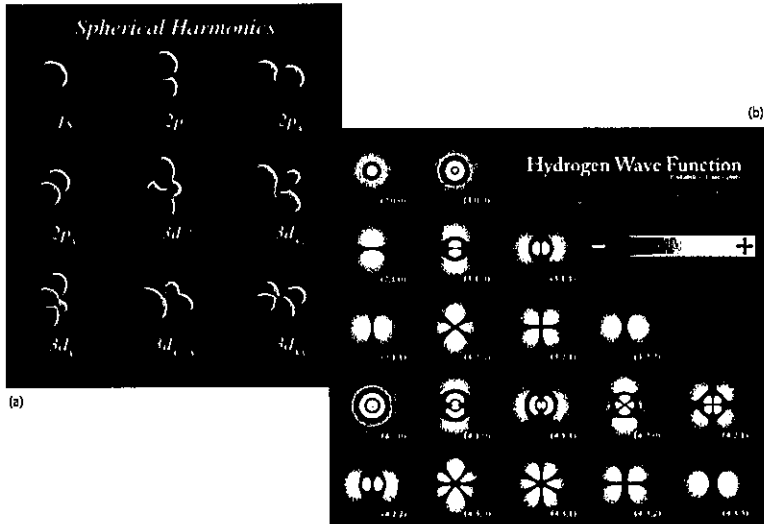


図 4.2: 水素原子の電子軌道

出典 <http://www.ss.scphys.kyoto-u.ac.jp/person/yonezawa/contents/program/gnuplot/gallery/spherical-harmonics-list.html>

絞られるということである。

原子の安定性は量子飛躍で説明できるが、生命現象が量子飛躍だけで説明できるとはおそらくシュレディンガーも思っていないだろうが、連続的な対象から、情報エントロピーを押しえ込む一つの手法として量子飛躍を一般化することが出来ることを見て行こう。

量子飛躍がどのような原理でおこなのかというのをまともに説明しようとする量子力学を熟知しなければならないが、一言で言うならばこれは電子の共鳴現象である。共鳴といえば、音の共鳴現象というものもある。実は量子飛躍と音の共鳴現象はよく似ている。

図 4.2 は、水素原子の電子軌道の図である。(a) は模式的に表したものであり、(b) は電子の存在分布確率を色の濃淡であらわしたものである。これに対

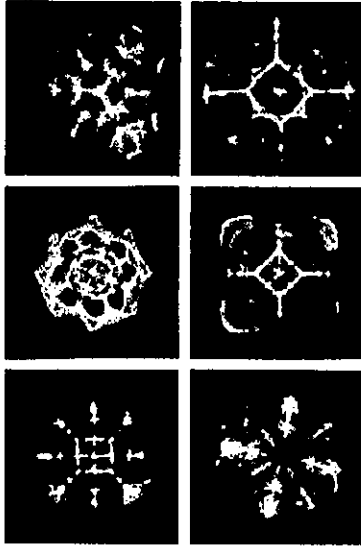


図 4.3: 音の波紋

出典 http://www.world-mysteries.com/sci_cymatics.htm

して、図 4.3 は、ティンパニーの上に胡麻をまき、いろいろな強さで叩いたときに胡麻が作る紋様であり、つまり音の波紋である。この二つの図を見比べて、どことなく似ているのは偶然ではない。両方とも同じ共鳴現象だからなのだ。音が共鳴現象を起こすことは経験することだ。量子力学では、電子や陽子は、いわゆる我々が想像するような粒子ではない。粒子として出現する確率はシュレディンガーの波動方程式に従う波動関数としてふるまう。世の中には波動関数が満ちあふれていて、その波動が干渉し合い、また、共鳴し合っているいろいろなことがおこっている。電子軌道も同じである。電子が原子核に引き寄せられる時、丁度原子核を底とした球形の壺のような空間に電子がたまっていると考えれば良い。電子はこの原子核の作り出す閉鎖空間の中で共鳴をおこすのだ。ティンパニーの音も同じである。ティンパニーも円形の膜という閉鎖空間で音が共鳴しているのである。

球形や円形の空間の中では波動は四方八方に飛び散り、複雑な共鳴を起こす。もう少し話を簡単にするためには、笛のような細長い空間で、波動は右か左にしか移動できないような状況を考えれば良い。音波は空気の分子の振動である。音は管の太さが変化するところで反射する。笛の吹き口側の端は壁となっていて、また、開口部も太さの変わる場所である。この二カ所で音は反射する。つまり音は管の中で何度も往復するのである。図 4.4 は音の成分のうち、波長が笛の長さの $4/5$ となる音の往復を表したものである。この (a) はその音の往路、つまり笛の吹き口から開口部へ向かう音を表している。音の波形が時間を追って (1)(2)(3)(4) と変化して行く。同時間における、同じ波長の音の復路を (b) に示している。図中の数字 (1)(2)(3)(4) は (a) の場合の同時間を表している。この往路と復路の音が重なり合うと、(c) になる。

往路の波と復路の波は吹き口のところで重ねると、必ず 0 になる。これは、吹き口のところが壁であり、空気が動くことが出来ないためである。このようなところでは反射すると波のエネルギーが反転する。一方開口部は常に振動している開放状態である。開放状態のところで反射する波はエネルギーを保持して反射する。このような状態で重なり合った音をみると、音は進行も、後退もせず、同じ場所で振動する波形となる。いわゆる定常波と呼ばれる波である。また、笛の管の特定の位置、図では、吹き口の位置から、笛の長さの $0, 2/5, 4/5$ の場所に全く振動しない。一方、吹き口の位置から、 $1/5, 3/5, 1$ の場所は常に

一番大きく振動している。前者を節、後者を腹という。これが共鳴である。

共鳴はどんな周波数の音でもおこる訳ではない。「波長が笛の長さの $4/5$ となる音」というのが肝である。この例では、吹き口は必ず節となり、開口部は腹となる。この条件を満たす音の波長は、 n を 0 以上の整数として、笛の長さの $4/(2n+1)$ となる。つまり、波長はとびとびとなる。音楽で言うならば倍音である。 $n=0$ のときを基音と呼び、 n が大きくなるにつれて、 $2n+1$ 倍音と呼ぶ。この場合は奇数倍音しか発生しないことが分かる (図 4.5)。

声で言うならば地声と裏声というのがあるが、これは一種の倍音の関係である。裏声をだすときには力を入れなければならないが、すこし力をいれたくらいでは声は裏返らない。ある一定以上の力をおなかに加えないとならない。金管楽器を演奏したことがある人ならばもっと分かりやすい。ピストンを変えずに音を出そうとするならば、マウスピースに加える力を変えればよいのだが、力をだんだん強くして行っても、連続的に音程を変えられる訳ではない。力がある一定以上こめないと次の音程の音へ飛ばない。一方で、一旦ある音程の音を出すことに成功すればその音程がぶれることは少ない⁶。なぜならば、この力が音の変化に対するバリアになるからである。これが、原子における電子に置き換えるならば、量子飛躍の原理である。

4.8 振動・ループ

共鳴によって生じる定常波は、ある一カ所で往復運動する波である。このような「振動」は共鳴のみならず様々なところに出現し一種の安定性を与える。天体の動きや、乗り物の車輪などは円運動またはその類型である。円運動というのは、振動とは一見無縁であるが、これは 90 度位相がずれた縦波と横波の合成運動である (図 4.6)。振動の位相や振幅、周波数が異なると、円運動は崩れ、いろいろな形を形成する (図 4.7)。このように、ある場所から出発し、一定周期で同じ場所に戻ってくる回帰運動はすべて縦振動と横振動の合成である。

戻ってこない運動は、一瞬現れ、やがてどこか観測圏外へ出て行ってしまふ、

⁶実際には、楽器の管は完全な円筒形ではないこと、また空気の粘性等非線形要素が影響し、音程は力の加減で若干ぶれる。しかし大きくぶれることはなく、音を連続的に変えるプロットメントのような奏法はそうとう熟練した奏者でないと難しい。

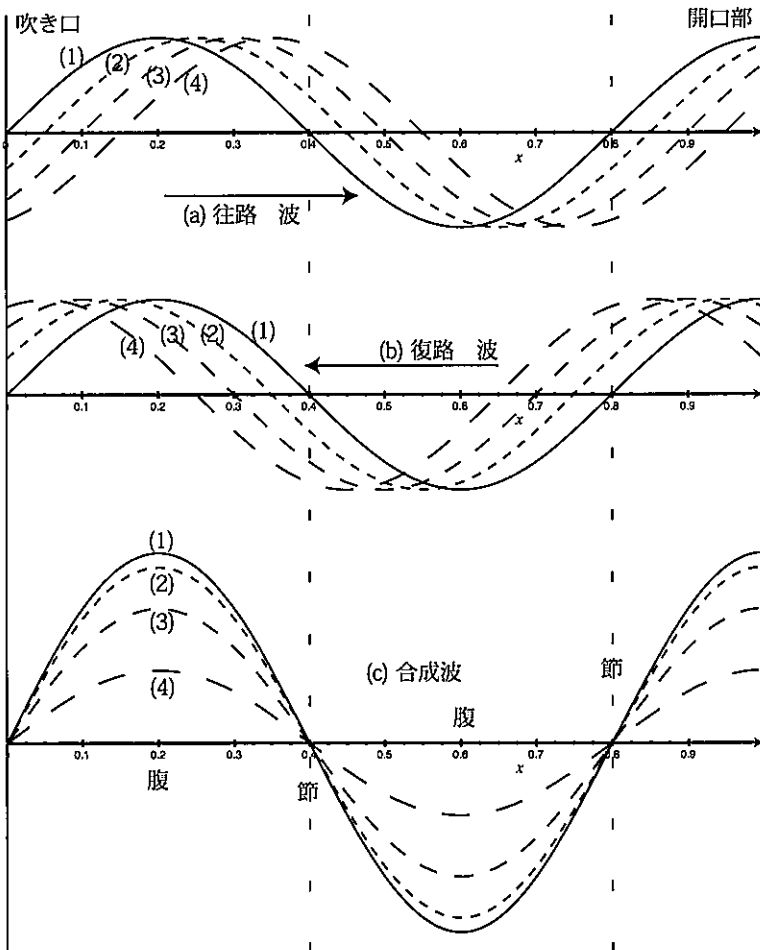


図 4.4: 音波の合成と共鳴

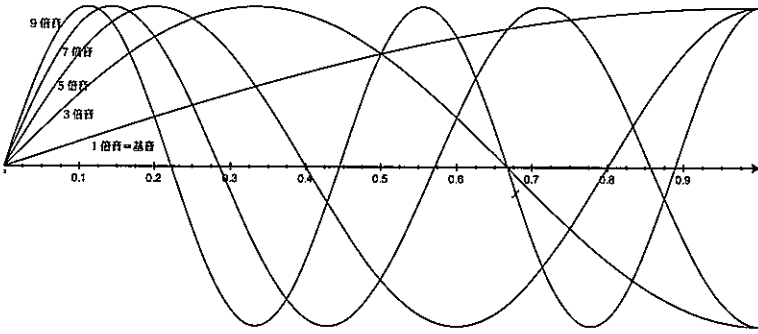


図 4.5: 倍音

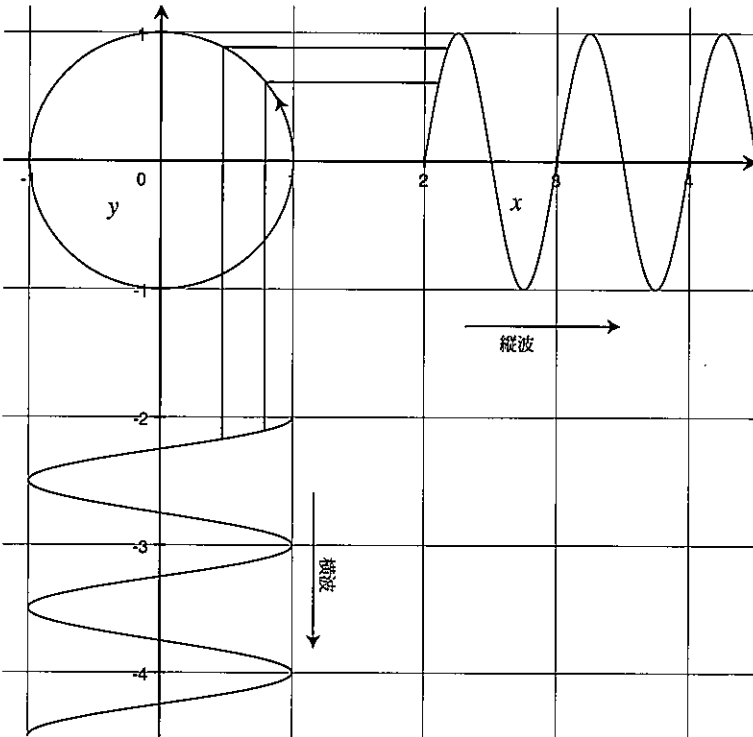


図 4.6: 円運動と波

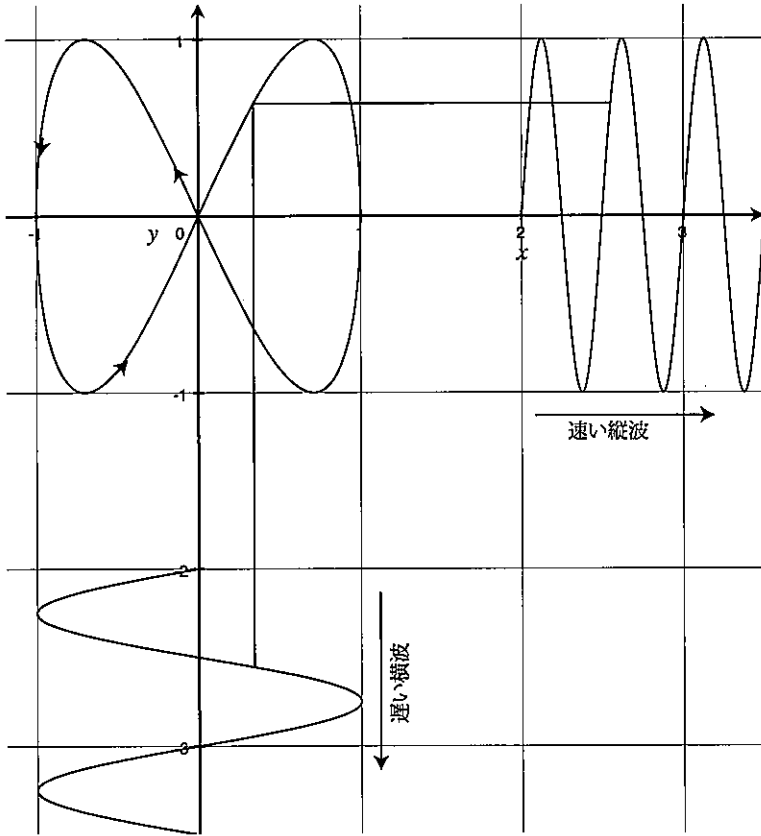


図 4.7: 八の字運動と波

ということである。それが戻ってこないという意味である。つまり全体として安定しない。そのように考えれば安定する運動が往復運動であるというのはある意味自明なことである。私は考え事をするとき、家のなかをあても無く歩き回っている。私の家は狭く、閉じた空間であるために、行き着けるところはおおよそ知れている。従って私の運動は必然的に往復運動になる。往復運動は精神の安定をもたらす。子供達の遊びには往復運動や円運動が多数含まれている。カゴメカゴメなどのミミクリ、ブランコやシーソーといったイリクスにみてとれる。心臓の鼓動、呼吸は往復運動であり、遊戯、ダンス、音楽、詩の朗読が身体運動である以上、心臓や鼓動と連動し、往復運動が沢山含まれることになる。往復運動は一定のリズムを作り出す。朝、日が昇り、夜、日が沈む、という一日の周期、月の周期、太陽の周期という往復運動が暦であり、季節、人や動植物の生理はおおかたこれに同期する。

コンピュータ・プログラムにおいて往復運動はループという形で現れる。ループが全く存在しないプログラムは、起動したとたん終了してしまう。おおよそ実用的なプログラムではない。プログラムにおけるループで代表的なものにイベント・ループがある。たとえばユーザからのキーボード入力やマウス操作を待ち受け、入力が確認されると、その入力を分析し、必要な処理を起動する、というものである。起動した処理が終了すると再び待ち受けに戻る。図 4.8 は、キッチンタイマーを実現したイベント・ループである。キッチンタイマーはボタンが押されたか、カウントダウンが終了したかを常にチェックしている。ボタンが押された場合、それが時間設定であれば、時間設定を行う。スタートストップボタンが押されたのであれば、カウントがまだ始まっていないならばカウントダウンを開始する。カウント中であれば、カウントダウンを中止する。カウントダウンが終了したならば、アラームを出力する。

数値計算の世界でもループ、繰り返しが多く存在する。まずは級数と呼ばれるものである。以下は円周率を求める級数の一例である。

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} \quad (4.14)$$

以下のような変わったものもある。

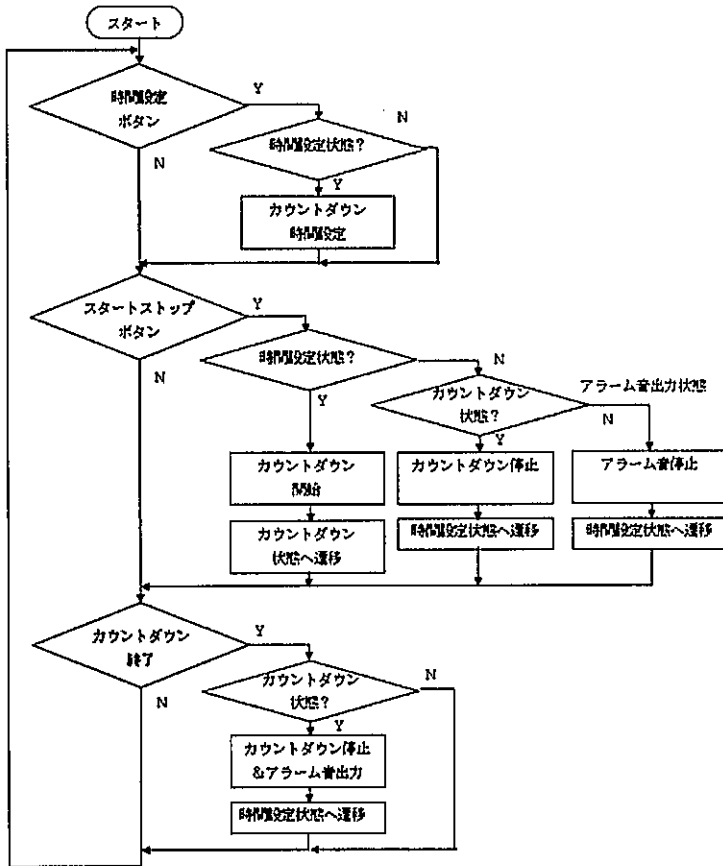


図 4.8: イベント・ループ キッチンタイマーの実現例

出典: [http:](http://monoist.atmarkit.co.jp/mn/articles/1211/07/news003_2.html)[//monoist.atmarkit.co.jp/mn/articles/1211/07/news003_2.html](http://monoist.atmarkit.co.jp/mn/articles/1211/07/news003_2.html)

$$\frac{4}{\pi} = 1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{7 + \frac{1}{9 + \frac{1}{11 + \frac{1}{\ddots}}}}}}} \quad (4.15)$$

4.9 共振・崩壊

しかしときとして、往復運動やループは事態の破壊をもたらす。数値計算の話からしよう。有限な回数の演算は、ゼロによる割り算がない限り必ず解が存在する。しかし無限回の繰り返しでは、以下のように解が存在しない場合がある。

$$x = 1 - 1 + 1 - 1 + \dots = \sum_{n=0}^{\infty} (-1)^n \quad (4.16)$$

この級数は、次のように考えれば、

$$x = (1 - 1) + (1 - 1) + \dots \quad (4.17)$$

$x = 0$ ということになる。しかし、次のように考えれば、

$$x = 1 - (1 - 1) - (1 - 1) - \dots \quad (4.18)$$

$x = 1$ ということになる。次のようなうがった考え方も出来る。

$$x = 1 - (1 - 1 + 1 - 1 + \dots) = 1 - x \quad (4.19)$$

この方程式を解くと、 $x = 1/2$ である。

一般に、各項の絶対値をとって作った級数が発散する場合は、その級数がたとえ収束するものであったとしても、計算の順序を変えると異なる計算結果が得られる⁷。上記の例はその一例である。従って発散する級数である。このよ

⁷絶対値をとった級数が収束する場合、絶対収束という。絶対収束しないが、収束する数列の例としては、

うに無限に繰り返される現象はアルゴリズムにとってはよからぬ結果をもたらす可能性がある。

級数のみならず、日頃プログラミングをやっていると停まらないループに悩まされることはしばしばである。単なるケアレスミスもあるが、本質的に難しい問題も存在する。たとえば、こんな例はどうだろうか。ホームページを作るときには、各ページごとにHTMLという言語で記述されたファイルを用意する。各ページに書かれている文章や絵の中にはボタンを仕込むことはしばしばある。ボタンをクリックすると、別のページへ飛ぶように作る。このように、リンクで結合された多数のページからなるハイパー・テキストを構築して行く。

ホームページの内容が新しくなり、いろいろ書き換えているうちに、ボタンとリンクも書き換えたり、消したりすることはしばしばであろう。しかしこの書き換えによって、どっからも参照されないページが増えて行く。このようなページはほぼ無用なのでページそのものを消すのが良い選択であるが、消し忘れたページのファイルがメモリを食いつぶして行く。

そこで、このような「ゴミ」を回収する方法を考えよう。一番確実な方法は、時々ゴミ回収⁸を行う方法である。トップページ（これはホームページとしてはいつも存在するページであるが）からリンクをたどって得ることのできるページにしるしを付けて行く。新しくしるしを付けるファイルが無くなったところで、今度は、ファイルシステム中のページを片っ端からすべて参照し、しるしの着いていないページを消去して行く。これをマーク・アンド・スイープ型ガーベージ・コレクションとよぶ。これは確かに確実であるが、ページ数が多いと、全てのページをアクセスするので実行に負荷がかかる。またそれだけではなく、ゴミ回収時は編集は禁止である。ゴミ回収とページの編集を同時に行うと、どこまでのリンクをチェックしたか曖昧になり、正しくゴミ回収が出来なくなる

$$\log(2) = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots = \sum_{n=1}^{+\infty} \frac{(-1)^{n+1}}{n} \quad (4.20)$$

がある。符合が交互になっているから収束するが、符合をすべてそろえると収束しない。

この数列を並べ替えて、4に収束する数列を作ろう。まず、4を超えるまで、プラスの項だけを順番に足しあわせる。4を超えたところで、こんどは、4より小さい値になるまで、マイナスの項を順番に足して行く。4を下回ったところで、こんどはプラス項を4を超すまで足す。これを続けて行けば、4に収束する数列に並べ替えることが出来る。これは任意の実数で実現可能である。

⁸ガーベージコレクションという。

ことがある。

このような問題を解決すべく、一つ思いつく方法は、参照カウンタを使う方法である。各ページに、そのページを参照しているリンクの数を示す参照カウンタを設ける。編集時に、リンクの新規生成、更新、消去のときに、一手間かけてこの参照カウンタを更新する。新しくリンクを貼った場合にはリンク先のページの参照カウンタを1増やす。逆に、リンクを消去した場合は1減らす。また、リンク先を変えた場合は、元のリンク先のページの参照カウンタは1減らし、新しいリンク先のページの参照カウンタは1増やす。参照カウンタがゼロになったとき、そのページを消去する。

この方法は、編集時に一手間かかるが、そんなに大きな手間ではなく、不要なページは常に消去され、効率がよい。一見見事に管理できるように思うが、リンクがループしているとき、つまり、リンクをたどって行くとともにのページへ戻って来れるような構造になっているとき、孤島が出来る可能性がある。この方法では孤島に属するページは消去されない。実際には、日頃はこのような参照カウンタ型のゴミ回収を行い、編集作業が行われていないときを見計らい、マーク・アンド・スワイプ型ガーページ・コレクションを行う、といった工夫をする。

このようなリンクによるネットワークと類似するシステムは、インターネットの検索技術、あるいはインターネットのルーティング技術など様々な場面で現れる。例えば、インターネットのルーティング技術において、情報にループが存在すると、送ったパケットが目的地に着く前に、不要なところでループにはまり、永遠に目的地にたどり着けないという現象がおきたりする。ルーティング技術ではこのような問題が発生しないように注意深くアルゴリズムが設計されている。

このようなリンクやインターネットといった径路を扱う問題は、グラフと呼ばれる構造で一般化される。グラフとは、ヴァーテックスと呼ばれる点とエッジと呼ばれる線で構成される図形である。エッジは方向を持っている場合と方向を持っていない、あるいは双方向の場合がある。丁度道路を思い起こしてた抱ければ良い。道路の交差点がヴァーテックスであり、交差点と交差点を結ぶ道がエッジである。道路にも二車線双方向のものと、一方通行のものがある。論理の証明の手順もグラフで扱うことが出来る。ある定理が別の定理で証明さ

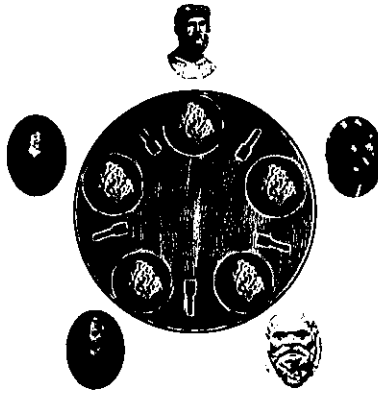


図 4.9: 哲学者の食事問題

れる場合、定理をヴァーテックス、証明において関係をもつ定理をエッジで結ぶ。エッジには、証明に使われる定理から証明される定理への方向が付されている。もし、この証明の手順にループが存在するとき、それは循環論法と呼ばれ、たとえ矛盾がなくとも証明が成立していないことを示している。

情報科学的な問題のあるループの例としてデッドロックという現象がある。図 4.9 は、五人の哲学者がテーブルを前に思索と食事に耽っている様子を示している。各哲学者の前には食事の載ったお皿があるが、食事をするためには自分の両側にある 2 本のフォークが必ず必要である。哲学者はフォークを取って食事をしたり、またフォークをもとあった場所において思索に耽ったりする。おのおのは勝手に思索と食事を交互に行うが、隣同士の哲学者が同じフォークを取り合うことがあり得る。だが哲学者はたとえフォークがいますぐに使えなかったとしても喧嘩はしない。人の使ったフォークで自分も食事をするという結構汚い話ではあるが、もともとは、エドガー・ダイクストラが提起したときには、並列に動く 5 台のコンピュータと 5 台のテーブルードの問題だったが、それをアントニー・ホーアがこのような話を書き換えた。

もし、5 人の哲学者が同時に食事を開始しようとし、右のフォークを取り、次に左のフォークを取りに行こうとした場合、左のフォークはすでに取られているので、5 人は永遠に待ち続けることになる。この場合は五すくみであるが、

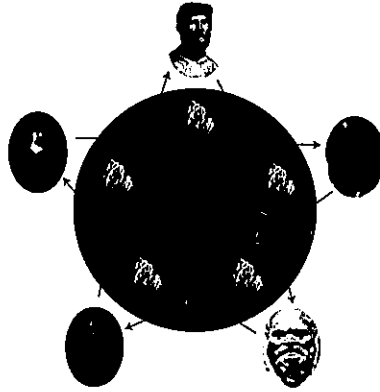


図 4.10: デッドロック

俗に言うところの三すくみの状態である。情報科学ではデッドロックと呼ぶ。フォークを取った哲学者について、取ったフォークから哲学者へ矢印を引き、待っているフォークへもう一つ矢印を引いてみよう。矢印は方向付きエッジであり、フォークと哲学者がヴァーテックスとした有向グラフを作ることが出来る。このグラフは一目瞭然ループをなしている(図 4.10)。さて、このようなデッドロックがおきないようにするためには 5 人の哲学者がどのように行動すれば良いのだろうか、というのがこの問題である。

デッドロックは上述フォークと哲学者の関係をグラフで表したときにループが生じないようにすれば、解決できるはずだ。一つの考え方としてダイクストラが提案した方法であり、フォークに通し番号をつけ、どの哲学者も、必ず番号の若いフォークから取らなければならないというルールを設ける、というものである。こうすれば、たとえフォークを取り合うような状況が生じたとしても、必ずグラフのエッジの方向は、小さい番号のフォークから、哲学者を經由し大きな番号のフォークへととなる。全体としても、エッジの向きは必ず最小値から最大値へととなりループすることはない。

デッドロックの問題を解消したところではあるが、良く食う哲学者に挟まれたよく考える哲学者は、両側の哲学者に常にフォークをとられ餓死する可能性がある。並列プロセスの競合においても同様な問題があり、これをスターバー

ションと呼ぶ。スターベーション（餓死）の由来はこの哲学者の食事問題と言われる。デッドロック問題と共にスターベーションも解決したアルゴリズムを1984年、K. M. Chandy と J. Misra が提案している。方法は、各哲学者に「思索中」「おなかがすいて食事（フォークの空き）待ち」「食事中」という三つの状態を準備する。「食事待ち」から「食事中」へ状態変化するのは、必ず二本のフォークが空いたのを確認してから、ということ、「食事中」から「思索中」へ状態変化するときに、両側の哲学者が「食事待ち」でかつ食事可能であるかどうかを確認し、両方を満たしているときには必ず「食事中」へ移行してもらうというものである。「食事中」から「思索中」への移行の処理がスターベーションを防ぐ働きをする。

解決法を聞くと、なんとも当たり前なもののように思える。実際、情報科学の問題の解法は、数学の高度な論法に比べると、ありふれたものが多い。しかし、情報科学の最終目的はプログラムとして正しく動くことである。たとえば、日常広く使われている並列処理の代表格として、インターネットのネットワーク・プロトコルがある。世界中に無数にあるコンピュータが正しく通信が出来るのも、上述のようなデッドロックの問題やスターベーションの問題が正しく解決されていることが前提である。並列処理におけるデバッグは大変なものである。おのおののコンピュータがランダムなタイミングで動作するので、同じバグが発生するタイミングがごくまれである場合が多い。ものによって一年に一度発生するようなバグというのものもある。しかしひとたびバグが発生すれば、ものによっては重大なデータの破壊などにつながることもある。バグを検知してバグを取るということは非現実的なのである。従って我々は、アルゴリズムを記述するときに、デッドロックやスターベーションのような並列処理上基本的かつ重大な問題がおきないことが論理的に証明できるということが重要である。ダイクストラや、Chandy and Misra の方法に代表されるように、問題解決に対して、証明可能な常套手段を与えるということが常に求められるのである。

論理的なループや情報のループが、情報システムに少なからぬ問題を引き起こすように、物理的な振動も時として破壊的な力を生み出す。関西から遠くで起きたはずの東北大震災によって、大阪の高層ビルでは唯一大阪ワールドトレードセンターがダメージを受けたのは記憶に新しい。これは地震波の周波数がビルの固有振動数と一致してしまったために、ビルが自身のエネルギーを溜め込

んでしまった、いわゆる共振である。また、世界的に有名な例としては1940年11月7日、米国のタコマ橋の崩落事故がある⁹。おりしもこの日吹き荒れた強風によって橋が異常な振動をおこし、最後には崩落してしまった。タコマ橋は施行当時からしばしば異常な振動を起こしていた。当時このような大規模な橋の建設は珍しく、建造物の風の影響について知識が乏しかった。これは風が橋を吹き抜ける時、橋の後ろで渦を作る。その渦の蛇行によって橋が上下に揺さぶられるのだ。橋が上下に揺れることによって風の道筋も揺さぶられ、渦の蛇行はさらに激しさを増す。その相乗効果によって、橋の振動がどんどん大きくなって行く。物体の後ろの渦の蛇行によって物体が振動するのは、いわゆるフォークボールの原理である。このような振動現象を自励振動と呼ばれている。

物理学的には振動は複数のエネルギーの相互変換である。もっと厳密に言えば、自由エネルギーの相互変換である。例えば、振り子の例を挙げて見よう。振り子は支点から垂れ下がったおもりが左右に振れるものであるが、振り子は左右に振れるだけではなく、若干ではあるが上下にも振れる。おもりが丁度支点の鉛直方向真下に来た時、おもりの位置は最も低い位置となる。それよりも左右にずれるに従って、位置は高くなり、おもりの振れが最も右または左に来た時、最も高い位置にくる。一方おもりの速度を見ると、この最も高い位置のとき速度ゼロとなる。速度が最も速くなるのはおもりの位置が最も低い鉛直方向の場所である。おもりの位置エネルギーと運動エネルギーの総和は常に一定であり、この二つのエネルギーが相互変換されるのが振り子である。

別の例では、音波は空気の圧力と空気分子の速度のそれぞれのエネルギーが相互変換されることによって音波が生まれる。笛の管の中の定常波もこの二つのエネルギーの相互変換によって生まれる。円運動の場合は少々分かりにくいですが、縦方向の運動エネルギーと横方向の運動エネルギーが相互変換されているのだと考えると理解できる。地震波も同じ原理である。

タコマ橋の自励振動は振り子等のように単純ではないが、風のエネルギー、橋桁やワイヤーの弾性エネルギーや運動エネルギーといった様々なことなる形態のエネルギーの間の相互変換、ループによるものである。このようなエネルギーのループが存在すると、そこにエネルギーが蓄積され限界を超えると対象物が破壊されてしまう。

⁹<http://www.sozogaku.com/fkd/cf/CA0000632.html>

エネルギーのループが生まれるためには、相互変換可能なエネルギーが2種類以上存在することが条件である。このようなエネルギーのことを自由エネルギーと呼ぶ。それとは反対に、他のエネルギーに転換できないエネルギーというものがある。具体的には熱源が一つの熱である。熱が介在する振動というのはあり得ない。熱は常に振動を止める方向に働く。大気中の振り子はやがて停止する。運動エネルギーは大気との摩擦により熱に転換され、熱はもはや運動エネルギーや位置エネルギーへ戻ってくることはない。

ループの外から入ってくるエネルギーとループから抜けて行くエネルギーがあり、前者が後者より大きいと破壊を招く。釣り合っている場合はループは永久に続き、後者が大きくなると、やがてループは消滅する。振動による破壊を免れるためには、ループへ入るエネルギーを防ぐ方法、つまり、固有振動数をずらし、共振しないようにするという方法、橋の強度を増し、そもそも振動しないようにするという方法である。一方、ループから出るエネルギー量、つまり摩擦や熱を利用する方法である。屋上に水槽を置き、水の揺らぎで免震するシステム、振動を押しえるダンパーやブレーキなどがそれだ。

ここで興味深いのは後者だ。免震システムやダンパーは、本来、エネルギーを熱エネルギーに変換する方法で、エントロピーを増大させるわけだ。一見エントロピーを減少させることが秩序を保つことのように見えるが、このようにエントロピーを増大させることによって目的を達成させることもあり得るということだ。

摩擦や熱が重要だという場面は我々の身の回りに多数存在する。例えば、乗り物は摩擦が存在しなければ、機能しないものばかりである。自動車は、地面とタイヤの間の摩擦を利用して発進、停止する。通常タイヤで走行中、予期せぬ天候不順で大雪に見舞われると、どうなるだろうか。それが坂道でもしたら、路面が氷結状態にでもなったら、車は横滑りをし、ゆっくりスピンし、コントロール不能な動きを始める。スケートリンクに放たれた犬同然である。

勇壮な航空ショー、ブルーインパルス急旋回は空気抵抗があつて初めて可能なのだ。もしこれが宇宙空間だったら、確かに映画「スターウォーズ」ではXウィングが宇宙空間を急旋回しているが、実際にはそのようなことはおこらない。空気抵抗がない宇宙空間で軌道を変更することはそう容易いことではないのだ。

大気圏の中にいる場合、我々に近づいてくるもののスピードには限界がある。あまりにも高速であれば燃え尽きてしまうからだ。だが宇宙空間ではそうはいかない。いくらでも速い飛行物体があり得るのだ。ひとたび衝突すれば、ほんのわずかの塵でもとてつもない破壊力を持っている。大気というものが摩擦と抵抗で、我々の住環境を安定状態に保っている。

振動や共鳴がエネルギーの変換ループにエネルギーを閉じ込めてしまう作用をもつものに対して、摩擦はエネルギーを拡散し一様なものへとする。これはエントロピーを保つ作用とエントロピーを増大させる作用に対応している。ここで振動に関する方程式、つまり波動方程式を見てみよう。波動の伝わる方向を x として、 $u = u(t, x)$ という時間変化する三次元空間に広がる波とする。 s は定数である。

$$\frac{1}{s^2} \frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} \quad (4.21)$$

左辺は、時間 t に対する 2 階偏微分で波の動きの加速度を表し、いわゆる運動エネルギーを司る項である。右辺は、 x 各軸の方向の張力を表す。音波であれば、空気圧の常圧からの変位であり、運動エネルギー以外のエネルギーを司る項である。この二つのエネルギーが交換されている様子が表されている。

これに対して、個体の中の熱の拡散を表す熱伝導方程式を見てみよう。別名拡散方程式という。 u の代わりに温度を表す T を用いると、 T は同様に $T = T(t, x)$ という時間と空間の関数である。 x は熱が拡散して行く方向である。 a は定数として、

$$c \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} \quad (4.22)$$

このように並べてみると、熱伝導方程式は、波動方程式と酷似している。異なる点は、左辺の偏微分の階数である。この背後には以下のオイラーの恒等式がある。

$$e^{i\theta} = i \sin \theta + \cos \theta \quad (4.23)$$

この公式は e という解析学における代表的な定数と、虚数 i という代数学における代表的な定数、それと、 \sin, \cos という幾何学における代表的な関数を結

びつけた式であり、数学におけるこの三つの分野を結びつける式である。実数領域では指数関数 e^x は x が増大するにつれて大変強力に発散する関数として知られている。逆に x を負の方向へ減少させると、関数は急速に 0 に収束して行く。ところが虚数方向では、式 4.23 によれば、複素平面の原点を中心に円運動を続ける関数である。以下のように実数領域と虚数領域を合わせて複素平面へ拡大すると分かりやすい。

$$e^{i\theta+r} = e^r(i \sin \theta + \cos \theta) \quad (4.24)$$

$i\theta$ という虚数領域が、指数関数の値域では円運動に対応するならば、一方で、 r という実数方向の増減は、値域では円運動の半径 e^r に対応する。つまりはオイラーの恒等式によれば、指数関数は回転という波動にまつわる性質と、増大・減少という拡散にまつわる性質の両方が含まれた関数であると言える。そして、波動方程式 (式 4.21) と熱伝導方程式 (式 4.22) の差は、この指数関数のもつこの二つの性質のどちらを引き出すかの違いである。

ために、

$$u = e^{2\pi i(\omega_t t + \omega_x x)} = e^{2\pi i \omega_t t} \cdot e^{2\pi i \omega_x x} \quad (4.25)$$

とおいてみよう。これは時間 t の方向、あるいは、 x の方向へみてみると、複素平面上で円運動する関数である。円運動の実数だけをみれば波である。 ω_t, ω_x はこの円運動の周波数である。これらの値が大きければ円運動は速くなる。

これを式 4.21 に代入してみると、

$$\frac{1}{s^2} \cdot (2\pi i \omega_t)^2 u = (2\pi i \omega_x)^2 u \quad (4.26)$$

となる。簡略化すると、

$$\omega_t^2 = s^2 \omega_x^2 \quad (4.27)$$

という方程式が得られる。この式によれば、時間方向の周波数は ω_t 空間方向の周波数で表すことが出来ることが分かる。ここで重要なのは ω_x が実数であれば、 ω_t も実数であるということである。

では、同様に熱伝導方程式で同じことをやってみよう。

$$T = e^{2\pi i(\omega_t t + \omega_x x)} = e^{2\pi i \omega_t t} \cdot e^{2\pi i \omega_x x} \quad (4.28)$$

とおく。同様な計算を施すと、

$$\omega_t = \frac{2\pi i}{a} \omega_x^2 \quad (4.29)$$

となる。式 4.27 と見比べて分かることは、虚数が消えていないということである。この効果は、これを、式 4.28 に代入すると、すぐに分かる。

$$T = e^{-(2\pi)^2 \omega_x^2 t} \cdot e^{2\pi i \omega_x x} \quad (4.30)$$

となり、時間にかかる虚数 i が消え、時間方向が円運動関数ではなく、減衰曲線へと変化する。

4.10 フィードバック制御

ウィーナーは『サイバネティックス』[44]の中で、生命を制御システムとして捉えようとしている。特にその中でもフィードバック制御というのがテーマとなる。フィードバック制御の代表的な例として私がいつも取り上げるのがライト兄弟による動力つき飛行機の発明である。

飛行機はライト兄弟が発明するまでもなく、じつはそれ以前からあった。早い話、紙飛行機も飛行機である。紙飛行機に、動力としてゴムで回転するプロペラを付ければ、それも確かに飛ぶ。しかしこれらは全て無人であった。人がひとたび乗るとなると、問題は格段と難しくなる。人の重みを空中へ持ち上げるほどの力もち、しかも小型で軽いエンジンが必要ということもさることながら、機上にいる人間の動きが攪乱要因となり、飛行機はバランスを崩して墜落してしまう。

様々な試行錯誤と実験を通してライト兄弟がたどり着いた結論は制御することである。飛行機が右に傾くと機上の人間も右に傾き、重心が右に寄ることによりさらに飛行機は右に傾こうとする。かつて説明した自励振動のよう

に、傾きはどんどん増幅されやがて墜落する。飛行機が右に傾いた時には、右側を持ち上げる力を加えそれをもとに戻す制御が必要である。ライト兄弟は、単に進行方向を決めるだけでなく、翼のねじれやフラップを利用した様々な姿勢制御用のアクチュエータを備えた飛行機を作ったのである。

これは、操縦士の感覚器から入った飛行機の姿勢情報を目標の姿勢と比較し、その変位を打ち消すように、操縦士がアクチュエータにフィードバックし姿勢をたもつ、という、人間が介在するとはいえ、れっきとしたフィードバック制御である。人間というのは本当に便利で有能な制御システムである。ライト兄弟たちもほんの1～2回の飛行練習でかなりの長距離を飛べるようになった。

しかし、これを全自動で行おうとすると至難の業である。飛行機の姿勢制御は本当に複雑かつ高度な制御が必要なものなので、ここではあらたに簡単なフィードバック制御を考えてみよう。底に穴が、それもかなり大きな穴があいたバケツがあるとする。ここに水道から水を注入し、常に水面がバケツの半分のところにあるように制御することを考える。水道の蛇口は開けるか締めるかで、中間の状態がないとする。バケツの半分のところを線を書いておき、それよりも水面が下回った場合、水道の蛇口を開け、上回ってきたら締める、というのを繰り返せばよいはずだ。しかし、実際にやってみると、蛇口の開閉が水量の変化に追いつかず、バケツの中は空っぽになったかと思えばこんどは満タンになり、また空っぽになり、完全にフラッタリングの状態となる。もう少し工夫しよう。水道からの水の流入量を、バケツの目標値からの変位に比例させるというのはどうだろうか。実際にやってみると、確かに、完全なフラッタリングよりはましになるが、水面の高さは目標値の周りで振動を始める。実際に姿勢制御などのフィードバック制御の設計を誤れば、このような状態に陥る。

人間で言えば、酒に酔って千鳥足の状態である。まっすぐ進むという目標値に対して、右に傾いたときに左へ姿勢制御するが、それをやり過ぎ今度は左に傾きすぎる。同様に右へ制御をかけるが、あらためて右へ傾きすぎる。結果、まっすぐ歩くことが出来ない。酔っぱらっていない人間は感覚的に分かっているが、機械的にはどのようにしたら良いのだろうか。目標値からの変位だけでは目的を達成できない。変位がどのような時間変化をするかが重要になる。変位の速度という時間による1階微分、変位の加速度という時間による2階微分、この両方の情報をいい塩梅で加味する必要がある。ここで思い出されるのが、

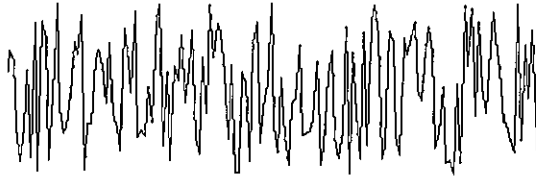


図 4.11: ホワイトノイズ

波動方程式と熱伝導方程式において、時間で何階微分したかという左辺の状態が、振動を生み出すか、拡散を生み出すかの違いを引き起こしたという事実である。共振をふせぎ拡散を促進するようにフィードバックループを組む必要がある。

4.11 $1/f$ ゆらぎ

共振現象によるエントロピーの維持、または縮小、一方で、拡散によるエントロピーの増大を見てきた。さらに複雑な系である流体では、共振や拡散が複雑に絡み合っている¹⁰。思い起こせば、風や川の流れ、またそれらの微振動部分としての音、あるいは乗り物の風きりとといった人工的なもの、ろうそくの炎が揺らぐのも、ろうそくによって暖められた空気の流れによるものである。我々の身の回りに流体はとても身近な存在なのである。

そういった流体の中に存在するある種の振動に $1/f$ ゆらぎという現象がある。心臓の鼓動、鈴虫やコオロギの鳴き声、蛍の光かたなど、流体でなくとも、この $1/f$ ゆらぎが観測されるものもある。おおよそ我々が心地よいと感じる振動と言われている。なぜ心地よいのか、人間原理的に考えればこうなる。もしこれら自然界に充満している振動を不快であると感じるならば、我々は直ちにノイローゼになってしまうだろう。一生の間、感じ続けなければならないのだから、そのストレスのために早死にしてしまうだろう。子孫を残そうとかそういった気になれないかもしれない。こういった振動を不快と感じる祖先たちは、早い

¹⁰ 流体力学の分野では、共振現象と拡散現象を分散と散逸とよぶ。厳密に分散とは、波動の空間的広がりや時間的広がりとの関係である。

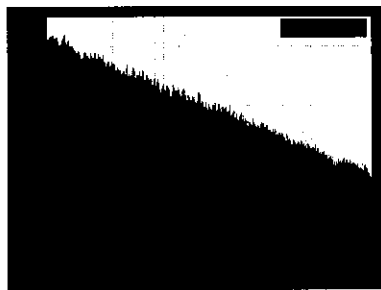
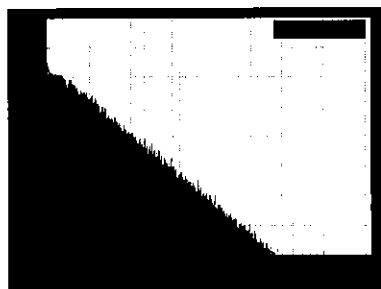
(a) ホワイトノイズ $1/f^0$ ゆらぎ(b) ピンクノイズ $1/f^1$ ゆらぎ(c) ブラウンノイズ $1/f^2$ ゆらぎ

図 4.12: 各ノイズのパワースペクトル

時期に絶滅してしまったのだろう。ただ以前に小鳥のさえずりをうるさいと怒鳴りつけていた子供を見たことがある。例外もあるようだ。ただ、多少の例外はあっても、一般的な進化には影響を及ぼさない。このようにして $1/f$ ゆらぎは心地よい振動ということになったのだろう。話をもとに戻そう。自然界の多くの振動がどうして $1/f$ ゆらぎになるのか、全ての現象について説明できる訳ではない。しかし、少なくとも流体についてはある種の相関関係がありそうだ。

まず $1/f$ ゆらぎの定義から始めたいところだが、もう少し広い概念から始めよう。 $1/f$ ゆらぎには、 $1/f^0$ 、 $1/f^2$ という仲間がある。これらの仲間とあわせて、 $1/f$ ゆらぎを $1/f^1$ ゆらぎと書く。別名、 $1/f^0$ をホワイトノイズ、 $1/f$ ゆらぎをピンクノイズ、 $1/f^2$ をブラウンノイズと言う。

まずこの三つのなかで基本となるのがホワイトノイズである。音における、ホワイトノイズを聞いたことのある人は多いだろう。ザーとか、シャーとかいう持続音である。波形で描くと図 4.11 のようになる。ホワイトノイズはランダムに発生するノイズの代表で、この音、もしくは振動のパワースペクトルは、一定である¹¹。あくまでもパワースペクトルが一定なので、各周波数の位相方向のずれにランダム性がある。もし位相方向がそろっているとすると、ブツツという音、つまり、インパルスとなる。ホワイトノイズを $1/f^0$ ゆらぎという

¹¹ ホワイトノイズ $w(t)$ は、次のように定義される。

$$E[w(t)] = 0 \quad (4.31)$$

$$\text{Var}[w(t)] = \sigma^2 < +\infty \quad (4.32)$$

$$\text{Corr}[w(t), w(t-T)] = \begin{cases} 1 & T=0 \text{ のとき} \\ 0 & T \neq 0 \text{ のとき} \end{cases} \quad (4.33)$$

式 (4.31) はホワイトノイズ全体の平均が 0 ということ、線形性がないことを意味する。式 (4.32) は分散が σ^2 ということ、有界な波形であることを示している。また、パワーが σ^2 であることもある。式 (4.33) はホワイトノイズを最も特徴づける部分であり、 $T > 0$ だけ時間をずらした関数 $w(t-T)$ 、これ自体もホワイトノイズであるが、これと元の波形は全く相関がないということである。もちろん $T = 0$ 、つまり自分自身とは相関がある。

他の式を加味しつつ、式 (4.33) を変形すると以下ようになる。

$$\int_{-\infty}^{+\infty} w(t)w(t-T) dt = \sigma^2 \delta(T) \quad (4.34)$$

δ はデルタ関数である。 $T = 0$ のとき $\delta(T) = 1$ 、それ以外のとき $\delta(T) = 0$ である。この式は畳み込みと呼ばれ、両辺をフーリエ変換すると、

$$|\hat{w}(\omega)|^2 = \sigma^2 \quad (4.35)$$

を得る。つまりパワースペクトルが一定である。

理由は、 $1/f^0 = 1$ であり、周波数 f によらずパワースペクトルが一定という意味である(図 4.12(a))。するともうお分かりだろう。 $1/f$ ゆらぎは、このホワイトノイズをフィルターに通し、パワースペクトルが周波数 f に反比例するようにしたものである。同様に $1/f^2$ ゆらぎは f の二乗に反比例するパワースペクトルもつノイズである。図 4.12 にそれらも合わせて記した。

ランダムな成分をもったノイズのスペクトルといえば、おそらく、これ以外にも多種あり得よう。我々がいやされる振動、あるいは自然界に多く見いだされる振動が、この整数乗反比例のパワースペクトルを持っていることが多い。なぜ非整数乗が存在しないのか。

この問題の一つのヒントが、あるスペクトルをもった関数、波動の積分、微分が、2乗パワースペクトルフィルタの役割を果たすことが挙げられる。パワー $|a|^2$ 、周波数 ω の波、 $a \cos(2\pi\omega t + \phi)$ を積分したとしよう。すると、

$$\int a \cos(2\pi\omega t + \phi) dt = \frac{a}{2\pi\omega} \sin(2\pi\omega t + \phi) + C \quad C \text{ は積分定数} \quad (4.36)$$

となる。位相は 90 度ずれ、周波数は変化しないことが分かる。積分結果のパワーは $|a|^2 / (4\pi^2\omega^2)$ である。つまり、周波数の 2 乗に反比例していることがわかる。波動方程式の解には、積分、微分が現れる。その度に、関数のパワースペクトルは、このように 2 乗単位で上下することになりそうである。しかし、それでは、1 乗単位で揺らぎが変わるといことが説明できない。1/2 階積分ということがあり得るならば話は別だが。

ここで注意しておきたいのは、上記の式では、 a や $a/(2\pi\omega)$ は波動の振幅である。これの 2 乗がその波動がもつパワーということになる。この関係が理解できていないとどのような刻みで、乗数が上下するかが理解できなくなる。整理すると、パワースペクトルに $1/f^0, 1/f^1, 1/f^2$ 対応する振幅スペクトルは、 $1/f^0, 1/f^{1/2}, 1/f^1$ になる¹²。

¹²ここで、フーリエ解析について、すこし厳密な議論をしておこう。フーリエ解析において主張されることは、ある条件を満たす関数 $f(x)$ は、 $\sin(2\pi\omega t + \phi(\omega))$ というサイン波の重み付け和で表されると主張されている。これがスペクトルの基本的な考え方である。 ω が周波数 $\phi(\omega)$ は位相である。位相は各周波数のサイン波ごとにあるので、 ϕ は ω の関数となっている。 a_ω を周波数 ω の重みとすると、式で書くと、

$$w(t) = \int_0^\infty a_\omega \sin(2\pi\omega t + \phi(\omega)) d\omega \quad (4.37)$$

となる。 \int_0^∞ は、 ω を少しずつ可変しながら足し合わせたということである。ところが、 $\sin(2\pi\omega t + \phi(\omega))$ は \sin と \cos の重み付け合成で書けば、位相 $\phi(\omega)$ は消せる。さらに、式 (4.24) を使えば、 $\sin \theta$ と $\cos \theta$ は、 $e^{i\theta}$ と $e^{-i\theta}$ の合成で書ける。となれば式 (4.37) は次のように変形できる。ただし、重みを a_ω と書くのではなく $\hat{w}(\omega)$ と書くことにすると、

$$w(t) = \int_{-\infty}^{+\infty} \hat{w}(\omega) e^{2\pi i \omega t} d\omega \quad (4.38)$$

となる。このとき、 $\hat{w}(\omega)$ を $w(t)$ のフーリエ変換、また、 $w(t)$ を $\hat{w}(\omega)$ の逆フーリエ変換と呼ぶ。あるいは、 $\hat{w}(\omega)$ を $w(t)$ の振幅スペクトルと呼ぶことが出来る。あるいは、 w は時間変化する関数なので時間領域、 \hat{w} は周波数を定義域とするので周波数領域と言ったりする。 w から \hat{w} を求める式は次のようになる。

$$\hat{w}(\omega) = \int_{-\infty}^{+\infty} w(t) e^{-2\pi i \omega t} dt \quad (4.39)$$

式 (4.38) と非常に良く似ているのが特徴である。この式を次のように略して書くと便利である。

$$\hat{w} = \mathfrak{F}[w] \quad (4.40)$$

\mathfrak{F} は線形演算子なので、次のような関係が成り立つ。

$$\mathfrak{F}[v + w] = \mathfrak{F}[v] + \mathfrak{F}[w] \quad (4.41)$$

c を定数として、

$$\mathfrak{F}[cw] = c\mathfrak{F}[w] \quad (4.42)$$

また、次は畳み込み演算と呼ばれる式である。

$$v * w = \int_{-\infty}^t v(t-t')w(t')dt' \quad (4.43)$$

この式については、次のような関係が成り立つ。

$$\mathfrak{F}[v * w] = \mathfrak{F}[v]\mathfrak{F}[w] \quad (4.44)$$

つまり時間領域では畳み込み演算である v, w が周波数領域では両者の積となる。では、式 (4.38) を微分したり積分したりしてみよう。まず t による微分である。

$$\begin{aligned} w'(t) &= \int_{-\infty}^{+\infty} \hat{w}(\omega) \frac{de^{2\pi i \omega t}}{t} d\omega \\ &= \int_{-\infty}^{+\infty} 2\pi i \omega \hat{w}(\omega) \frac{de^{2\pi i \omega t}}{t} d\omega \end{aligned} \quad (4.45)$$

これは、

$$\mathfrak{F}[w'] = 2\pi i \omega \mathfrak{F}[w] \quad (4.46)$$

ということを示している。 n 階微分は、

$$\mathfrak{F}[w^{(n)}] = (2\pi i \omega)^n \mathfrak{F}[w] \quad (4.47)$$

と表せるはずだ。同様に、積分は、

この問題をいろいろ調べていたら面白い論文に出会った。「非整数 $1/2$ 階微分で与えられる履歴を伴う非線形波動」[45] というタイトルであり、 $1/2$ 階積分ではなく $1/2$ 階微分であるが、なにか近い物をおわせる。この論文の要旨は、様々な流体力学方程式を解くと、しばしば $1/2$ 階微分に相当する式が現れる。この $1/2$ 階積分というのは、様々な流体现象に履歴をもたらすというものである。

空気中を流線型の物体が移動していたとしよう。その物体の周りには少なからず空気の流れが出来る。しばらくたってその物体が静止したとする。しかし、静止しても、その物体の周りの空気の流れはすぐには取まらない。物体が静止するまえの状態がしばらくの間持続する。静止した時点における空気の状態が、それまでのいくらかの時間の状態に影響を受けている。これが履歴である。微分方程式の上では、しばしばこの履歴が $1/2$ 微分という形で現れるというのだ。

この議論の背景には積分と微分の相反する性質の違いがある。積分とは与えられた関数のいくつかの軸と関数で囲われた図形の面積を求める仕組みである。面積というのはその図形の大局的な大きさを表しているが、細かい部分については埋もれてしまう。それこそ平均化されかき消されてしまうのである。しかし一方で、 0 から t までの積分値というのは、この間の関数の状況によって決まる。 $0 \sim t$ の間に関数に変化があったとすると、その影響が t のところの積分値に現れてくるのである。一方、 $f(x)$ の微分は x の点での $f(x)$ の傾きと定義される。 x 以外の点には左右されることなく、 x 近傍の局所的な状況によって決まる値である。微分方程式は局所的な近傍での挙動を記述していることになる。近傍が近傍に影響を与え、影響が徐々に遠くへ伝搬して行く。もし、途中

$$\mathfrak{F} \left[\int_{-\infty}^t w(t') dt' \right] = \frac{1}{2\pi\omega i} \mathfrak{F}[w] \quad (4.48)$$

である。 n 階積分を、

$$I^n w(t) = \int_{-\infty}^t \int_{-\infty}^{t_1} \cdots \int_{-\infty}^{t_{n-1}} w(t_n) dt_n dt_{n-1} \cdots dt_1 \quad (4.49)$$

と定義すると、

$$\mathfrak{F}[I^n w] = (2\pi\omega i)^{-n} \mathfrak{F}[w] \quad (4.50)$$

となる。丁度積分と微分が逆の関係になっていることが分かる。周波数領域では微分や積分は $(2\pi i\omega)^n$ を掛け合わせるだけなのである。

のある点が不連続であったりした場合、そこで伝搬は止まってしまい、それ以上の解は求められなくなってしまう。積分方程式の場合、積分は大局的な状況を表すので、極端な話ある点の値が無かったとしても、結果に影響は与えない。

もし流体力学の微分方程式がこの微分の性質が色濃く出ていたとするならば、流線型の物体が停止した瞬間に周りの流体の運動も停止することになる。微分方程式の中に何らか積分の性質が入ってこないと過去に履歴が残るということがおこらない。[45] では、 $1/2$ 階微分の現れるいくつかの例を示している。その中の一つとして、無限に広がる動く平板の上に h の深さにたまった流体の動きがある。平面の動く方向を x 軸として、その鉛直方向を y とする。 $y = 0$ が動く平面とする。流体は平面の動きに合わせて x 軸方向へ流動する。その動く速度を $v = v(t, y)$ とする。平面は無限に広く、 x 軸のどの点でも同じ動きをするために v は x によらない。このとき、

$$\frac{\partial v}{\partial t} = \frac{\partial^2 v}{\partial y^2} \quad (4.51)$$

という関係となる。境界条件として、 $y = 0$ のとき $v = v_0(t)$ は平板の速度となる。

さてこれは、拡散方程式 (4.22) と酷似している。なので、この例を引き合いに出したのだが。直感的な話をすれば、平板の動きが流体中へ徐々に拡散して行くということである。またこの例は、[45] において挙げられている例の中でも、比較的簡単に $1/2$ 階微分が現れる理由を説明することが出来るものである。式 (4.51) における左辺等辺の階数の比が $1:2$ になっていることが重要な意味をもつ。この式の右辺を左辺に移動すると、以下ようになる。

$$\frac{\partial v}{\partial t} - \frac{\partial^2 v}{\partial y^2} = 0 \quad (4.52)$$

いまだ、 $1/2$ 階微分というものがどういうものかは分からないが、 $1/2$ 階微分なるものを $\partial^{1/2}/\partial t^{1/2}$ と書くことにする。 $1/2$ 階微分を 2 階繰り返すと、 1 階微分したことになるのでと約束しよう。このように約束すると、上式は以下のように分解できる。

$$\frac{\partial^{\frac{1}{2}} w}{\partial t^{\frac{1}{2}}} - \frac{\partial w}{\partial y} = 0 \quad (4.53)$$

$$\frac{\partial^{\frac{1}{2}} v}{\partial t^{\frac{1}{2}}} + \frac{\partial v}{\partial y} = w \quad (4.54)$$

式(4.53)によってまず w を解き、この解をもって式(4.54)から v を解くということである。式(4.54)を式(4.53)に代入してみると、式(4.52)に戻ることが確認できよう。とりあえず $w = 0$ とおくと、式(4.53)は満たされている。すると式(4.54)から、

$$\frac{\partial v}{\partial y} = -\frac{\partial^{\frac{1}{2}} v}{\partial t^{\frac{1}{2}}} \quad (4.55)$$

という方程式が得られる。この左辺はせん断応力と呼ばれるものである。さて次に問題となってくるのは、 $1/2$ 階微分の正体はなんだろうかということである。答えから言ってしまうと、

$$= -\frac{1}{\sqrt{\pi}} \int_{-\infty}^t \frac{1}{\sqrt{t-t'}} \frac{\partial v}{\partial t'} dt' \quad (4.56)$$

である。式(4.56)において $1/2$ 階微分は積分という形で表現され、つまり時刻 t 以前の状況が履歴として t の時刻に影響を及ぼしている。この式は、[45]によればコーシーの積分公式であるという。コーシーの積分公式、あるいはコーシーの積分定理は、実数や複素数空間上の多重積分や多重微分を一つの積分に書き直せるという大変便利な公式、定理である¹³。上記の式は、 $\partial v / \partial t'$ の部分

¹³ここで使われているコーシーの積分公式：

$$I^n f(t) = \frac{1}{(n-1)!} \int_0^t (t-t')^{n-1} f(t') dt' \quad (4.57)$$

これを、 n が実数でも展開できるようにしたのが、ガンマ関数 $\Gamma(n)$ をつかい、

$$I^\lambda f(t) = \frac{1}{\Gamma(\lambda)} \int_{+\infty}^t (t-t')^{\lambda-1} f(t') dt' \quad (4.58)$$

$0 < \nu < 1$ の間の ν 階微分は、一階微分し、 $(1-\nu)$ 階積分すればよい。よって、

$$\frac{d^\nu f(t)}{dt^\nu} = \frac{1}{\Gamma(1-\nu)} \int_{+\infty}^t \frac{1}{(t-t')^\nu} \frac{df(t')}{dt'} dt' \quad (4.59)$$

で1階微分を行っている。そこに、コーシーの積分公式をつかって1/2階積分を行ったもので差し引き1/2階微分を行ったのと同じということである。ただし、コーシーの積分公式はあくまでも、整数階の多重積分に対する公式でしかない。これをむりやり非正数階へ展開すると上記の式が導出される。無理矢理なので、これが正しいかと言われれば若干不安が残る。しかし、1/2階積分を2度施すと、確かに1階積分の答えと一致する。

では、1/2階積分や微分が、周波数領域でどのような挙動をするのかを調べてみよう。ここでは1/2階積分の場合を確かめてみる。 $f(t) = e^{2\pi i \omega t}$ という周波数 ω の単振動を1/2階積分し、係数がどのように変化するかを見てみれば良い。

$$I^{\frac{1}{2}} e^{2\pi i \omega t} = \frac{1}{\Gamma(1/2)} \int_{+\infty}^t (t-t')^{-\frac{1}{2}} e^{2\pi i \omega t'} dt' = (2\pi i \omega)^{-\frac{1}{2}} e^{2\pi i \omega t} \quad (4.60)$$

である¹⁴。これは、周波数 ω の振幅スペクトルが、 $(2\pi i \omega)^{-\frac{1}{2}}$ 、パワースペクトルが、 $1/f$ となることを示している。

では、最終課題である、なぜ自然界には、 $1/f^n$ ゆらぎが生じるのだろうか。流体を微分方程式化すると、その解の一部に、 $n/2$ 階微分が現れる場合が多い。様々な初期条件の中に雑音があると、この解が一種のフィルターの役割を果たす。たとえば、式(4.56)で表されるような現象であれば、 $\partial v / \partial t'$ がホワイトノイズであれば、 $\frac{\partial v}{\partial y}$ は $1/f$ ゆらぎとなる。具体的にいえば、動く平板の加速度がホワイトノイズを作っていたとすると、流体の揺らぎが $1/f$ ゆらぎとなる。

この $n/2$ を決める要素は、微分方程式の微分の階数である。これが小さい整数比で表され、そのなかで最もシンプルなのが1(ホワイトノイズ)または1/2(ピンクノイズ)である。微分の階数は、複雑な微分方程式になればなるほど、

詳しくは [45] を参照されたい。

¹⁴ $r^2 = t - t' \geq 0$ とおくと、式(4.60)は、

$$-\frac{1}{\sqrt{\pi}} e^{2\pi i \omega t} \int_0^{+\infty} e^{-2\pi i \omega r^2} dr \quad (4.61)$$

と変形できる。積分はフレネル積分をつかって計算し、式(4.60)の右辺を得る。これは、式(4.50)と照らし合わせても、周波数領域での積分の効果と一致する。

多階のものも存在する。しかし解に対する影響の強さから考えれば、大きい階数の微分項は影響力が小さくなるだろう。

式(4.52)の微分の階数を多項式のようにとらえ、素因数分解することが出来る。すると以下のようになる。

$$\left(\frac{\partial^{\frac{1}{2}}}{\partial t^{\frac{1}{2}}} - \frac{\partial}{\partial y} \right) \left(\frac{\partial^{\frac{1}{2}}}{\partial t^{\frac{1}{2}}} + \frac{\partial}{\partial y} \right) v = 0 \quad (4.62)$$

このように書いてみると、階数の比が、最終的に微分方程式の解に立ち現れてくるメカニズムが分かる。

1/f ゆらぎの例は、微分方程式が離散的構造を内包している一つの例である。自然界が微分方程式に従って挙動している以上、このような離散的構造が、音、風などの自然な振動の中に組み込まれることになる。生物は進化の過程で自然とその離散的構造を認知し識別できるようになる。

離散構造を生み出す骨格構造は物理学の法則の中にあるというよりは、まず、数学、特に解析学の諸法則の中にあるということだ。素粒子論や相対性理論にこの世が従っていなかったならば、確かに、いま観測できる様々な現象の中で消滅するものもあるだろう。だが新たに生まれるものもあるかもしれない。少なくとも、解析学的な構造に基づいているならば、離散構造を生み出す素地をもっているといえる。しかし、それだけでは、小さな整数比にならない。大きな整数比は限りなく連続に近づいて行き、離散構造はどんどん見えなくなる。高周波よりは低周波、高階よりは低階へ、あらゆる整数比の中で小さい方へシフトするその原因は拡散現象である。物事はエネルギーの低い方へ流れる。その背後にあるのは熱力学第二法則である。高エネルギーの際立った状態からより乱雑な方向へ流れようとすることによって、かえって、解析学的な離散構造が際立つことになる。

ピタゴラスは、その狂信的とも言える信条から「万物は数である」と言った。ところがまもなく弟子たちによって無理数が発見されることになる。確かにこの信条は厳密には正しくないが、ある種の真理をついている。「美しい和音は簡単な整数比となる」というピタゴラスの主張は音の共振現象であり、それは量子飛躍の問題にも通じる。またこの1/f ゆらぎにも同様な主張が成り立つ。ピタゴラスの時代は現代のような解析学も物理学も発達していた訳ではない。し

かしその時代においてこのような主張が出来たということは、彼に先見の明があった、というよりは、生物学的な進化論の過程が、かれをしてそう主張させた。おそらく生物であれば誰でも感じていた自然界の離散構造を彼はただ人間の言葉にただけである。

「厳密には正しくないが、ある種の真理をついている。」味わい深い言葉である。

4.12 連続と離散の循環

これまでの議論を図にまとめてみると図 4.13 になる。この図は左へ行くほど乱雑性が強くなり、右へ行くくと規則性が強くなる。これは時系列的な流れではない。矢印は、それぞれの事象の関係を示しているもので、番号をふった各事象は同時に成り立っている。一つのことを乱雑と見るか、規則的とみるかという視点の違い、視点の移動なのだ。矢印は、ある視点で見えている事象から、考え方のシフトによって別の視点へ見方を移せるという意味である。

(0-1) の乱雑性から確率分布の矢印が (0-2) の連続量へ伸びている。(0-2) の連続量は、さしずめシュレディンガーの波動方程式である。(0-1) の乱雑性は、粒子の観測の乱雑性ということになる。2.3.8 節で述べたように波動方程式の解はほとんど至る所正則なので、正則的性質「あり」の矢印をたどることになる。これは、2.3.8 節におけるサンプリング定理により連続的な性質を離散的に分解する(量子化)ことが出来ることであつたり、または、前節のように微分方程式の性質として、共鳴等の離散的な性質を抽出できるということである。これによって対象が、原子、分子といった粒子として人間にとって認識可能になるのである。乱雑性の中からだんだん規則性が増して行くプロセスである。このようなプロセスは、量子力学、熱力学といった物理学の世界だけではなく、社会的なビックデータや、データマイニングの世界にも見られる、いわゆる平均化のプロセスである。

この図では、量子力学的でない連続量もあり得るだろうと想定している。その場合は正則的性質「なし」の矢印をたどることになる。この矢印の一方は、(0-3) 複雑性へと通じている。微分方程式としては表せない連続性は、複雑性もしくはフラクタルと呼ばれる。有名なものにペアノ曲線であつたり、マンデ

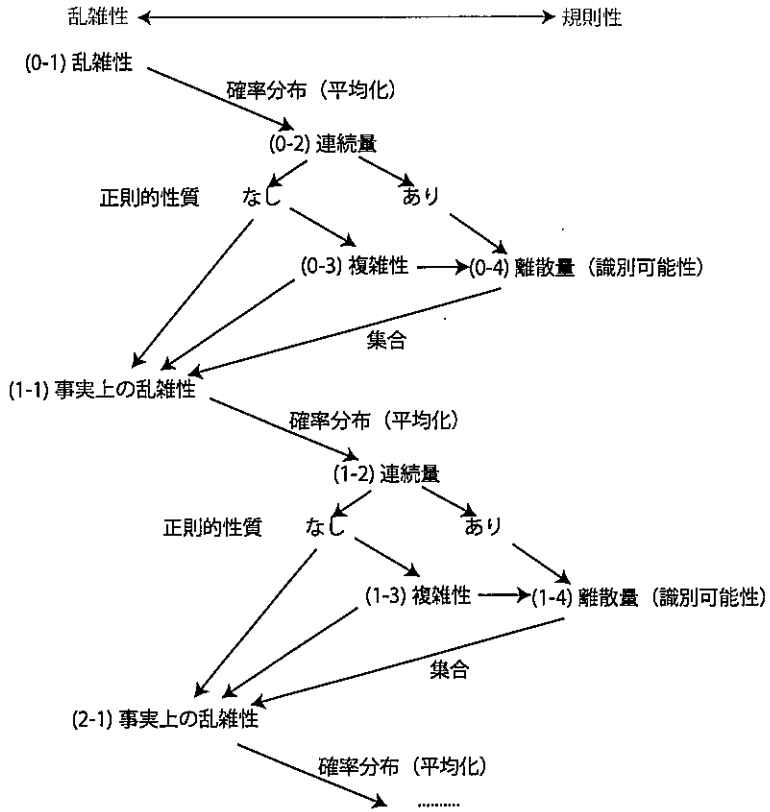


図 4.13: 連続性と離散の循環

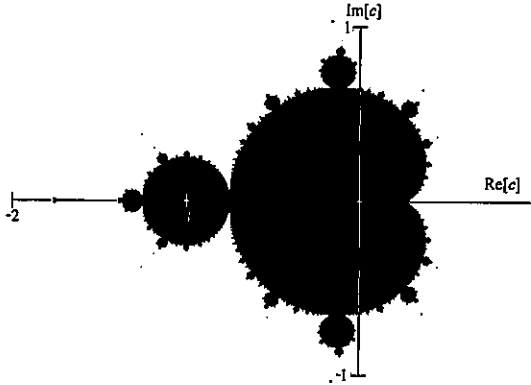


図 4.14: マンデルブロ集合

Hi-resolution Mandelbrot set with axes *Wikipedia*

ルブロ集合 4.14 がある。これらはある種の反復性、自己同相性といった性質もっている。有限な空間に折り畳まれた無限の長さをもつ線であり、その複雑な折り畳まれ方ゆえに、1次元より大きな次元をもつ¹⁵。

複雑性を示す事象は、樹木の形、入道雲、水にたらしめたインクなど、自然界に沢山見いだすことが出来る。これらの現象にも、ある種の規則性を見いだすことが出来る。つまり、これらを入道雲や樹木として認識できる。一方で、非

¹⁵ハウスドルフ次元。フラクタルは線だけではない。面、立体、あるいはそれ以上の次元のフラクタルもある。次元の考え方はいろいろある。最も一般的なものは空間的次元であり、座標系の軸の数を言うものである。有限な 2 次元図形つまり面のもつ面積は有限の量である。体積は 0 である。一方長さは？面の長さというのは少々おかしな言い方であるが、面の中に線を敷き詰めたところを想像すれば、無限大と見なすことが出来る。長さ (1 次元測度)、面積 (2 次元測度)、体積 (3 次元測度) という n 次元図形の大きさを表す量を n 次元測度と呼ぼう。すると、 d 次元図形の n 次元測度は、

$d > n$ の場合、無限大である。

$d = n$ の場合、有限である。

$d < n$ の場合、0 である。

という風に一般化できる。

ペアノ曲線やマンデルブロ集合のように D 次元の中に複雑に折り畳まれた d 次元図形の n 次元測度を考える。 $n = d$ のとき無限大であることは明らかである。 n を微妙に d より大きくすると、有限のポイントがあるはずだ。空間的次元より大きいこの次元をハウスドルフ次元といい、折り畳まれた図形の複雑度を表す尺度である。

ちなみにペアノ曲線もマンデルブロ集合もハウスドルフ次元は 2 次元である。

正則な性質があるが故に、それらがある種の規則性に従っていながらも、その成長は予測不能な部分が多い。事実上の乱雑性という性質をもち得る。(0-3)から矢印は、(0-4)と(1-1)へ伸びている。

規則的なものも沢山集合すれば、一つ一つの対象の振る舞いは事実上の乱数と見なすことが出来る。対象の総体は再び、確率分布として捉えることが出来、再び連続量へと編成可能だ(1-2)。このように見れば、視点の動きは、連続と非連続、乱雑性と規則性の間を行ったり来たりする、循環として捉えることが出来る。

この図を見ると、世界は乱雑性であるという人間論的視点か、あるいは世界はすべて規則的であるというラプラスの悪魔か、という二者択一ではなく、二者は同時に成り立っており、視点は常にこの二つの間を循環している。

第5章 主観的生命

5.1 生命

機械化する生命、生命化する機械、いま社会や科学技術でおこっていることをどう捉えたら良いのか。発達心理学、人類学の一つの考え方「心の理論」に注目してみよう [46]。「心」とは他者の中にも同種の「心」があると認める共感性の能力であるという考え方がある。この「心」の定義は、「心」自身の定義の中に「心」が現れるという再帰構造を取っている。従って、「心」を公理主義的に決定することは不可能である。だが、主観的な私が私自身に「心」があると認めるならば、私が対峙している他者にも同様な「心」があるだろうと共感できる。そうするとその他者の「心」も私同様に『他者の他者にも「心」があるだろう』と認める能力を持っているはずで、つまり他者の他者も私と同様な「心」を持っているはずである。生命とはなにかという問いに対して同じように、他者に生命を見いだすという、一種の共感として捉えることが出来る。

冬の時期、インフルエンザがはやり始めると、ワクチンを接種する人も多いかと思う。しかし、インフルエンザに新種が現れると効かないものも登場しかねない。特に最近では鳥インフルエンザのパンデミックがニュースの話題にのぼることがある。報道では「ワクチンを打っても死なないウィルスが出現する。」と語っている。だが、生物学的には代謝を行わないウィルスは生命ではない。「死ぬ」という言葉は当てはまらない。「ワクチンを打っても破壊しないウィルスが出現する。」というのが正しい。しかしこの言い方がしっくりくる一般人は少ないだろう。どうしてもウィルスに対して生命的なイメージを抱くのが一般人の性である。ひっくり返して言うと、我々は心の理論のように、外界のなにかに生命的なイメージをかぶせてみる能力をもっている。

心の理論全体は、人間や霊長類の発達におけるより高次な心理の一部として

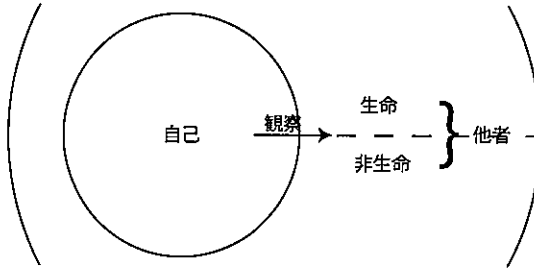
位置づけられており、ウイルスや機械などには見いだせないことである。しかし、「生命」として「共感」しているか、もつというなれば「生命」として「認識」しているか、ということに限れば、低次の現象にも拡張できるのではないか。もうすこし生物学者には納得できない話を続けさせてもらおうと、ウイルス自体も、自分が寄生できる細胞と出来ない細胞を判別する能力を持っている。そして、寄生を繰り返し増えて行く。これは、他の生命を認識する能力と言える。だからウイルスから他の細胞も生命として映っている。もうすこし厳密に言うならば、ウイルスから他の細胞も生命として映っている、というように私には映っている。ここに、私 → ウィルス → 細胞 → ウィルスという生命の認識の連鎖が存在する。

そもそも、生物学的に生命ではないものに対して生命を見いだす行為は珍しいことではない。幽霊や宇宙人といった想像上の産物、あるいは、天体現象に生命を見いだすことは、よくあることである。「他を生命として認識すること」という生命の定義の連鎖のネットワークは主観と矛盾に満ちているが、自然な生命の定義といえる。この矛盾に満ちたネットワークの中から切り出され、分離された、生物学的に整合性のとれた部分ネットワークが生物学的生命である。

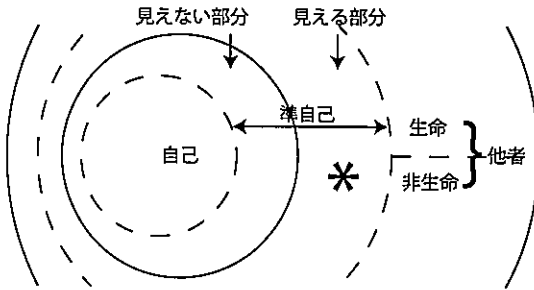
この話は、図 5.1(a) のように「自己」がある。この「自己」がその外界とおぼしき「他者」の中に「生命」を見いだす、というように図式化できる。「他者」の中で「生命」でないものも存在する。「他者」がなにか、ということも「自己」が勝手に決めている。「自己」が異なれば、「他者」も異なれば「生命」も異なる。しかし、「自己」→「生命」のネットワークを作ることが出来る、ということである。

ここで問題にしたいのは、この「自己」「他者」「生命」の境界線である。特に「自己」「他者」の間は、実際には連続的である。この境界領域を「準自己」と名付けよう。「準自己」はさらに「自己」から見える部分と見えない部分に分けられる。見えない部分というのは自分の体の内部のようなものである。一方、見える部分というのは、たとえば道具のようなものである。道具というのは他者であるが、自己の意思の延長を担うものでもある。使い慣れた道具は「自己」と同化して行く、とも言える、つまり見える部分から見えない部分へ移行して行くのである。

ここで図 (b) の「*」の部分が生命として認識されるのか、それとも非生命



(a) 心の理論からみた生命



(b) 生命の拡張

図 5.1: 生命

として認識されるのかである。西垣通が機械と生命を区別するとき [1]、この機械に対するまなざしは「*」であると言えないだろうか。すべて「*」の中身は知り尽くされたものであるからコントロール可能なのである。それは、自己からコントロール可能な延長線上にありながら、自己とは区別された位置にある。コンピュータ科学者としては「*」に位置するコンピュータはあくまで非生命であり「自己の支配下」なのだ。だが、一方で、コンピュータの中身を知り得ない人々にとって、「他者」から迫ってくる道具である。あるいは図 (b) の「他者」の場所に位置するものである。それは自己の延長線上ではなく、時として自己を脅かすものである。ある意味、ウイルスに感染するようなものである。それは物体でありながら、生命的イメージをもち得るのである。

分子生物学者の立場は矛盾に満ちている。研究対象は自分が造ったものではない「他者」としての「生命」を追い求め、科学的に同定する。この行為自体、上記の定義に照らせば、生命現象のひとつと言える。しかし研究がすすめば進むほど、研究対象の「生命」は「自己の支配下」へと鞍替えして行く。研究の成果が上がれば上がるほど、その成果は論理的なプロセスを経て似たものを造ることが可能となっていく。似たものはたして生命なのか。あやうい立場へ追いやられて行く。機械なのか生命なのかという対立概念で考えている以上、科学的な「『生命』の定義」それ自体が自己矛盾である。

一方、情報科学者やエンジニアが、人工生命や人工知能を開発しながらも、それが、生命に似せたものでありあくまでも生命ではないと思う理由もうなづける。自分が開発したものは、もともと「自己の支配下」である。開発者というものは、元来、開発対象が、自己の支配下あることを自負する。そういった気概が西垣のように情報はあくまでもヒトが生成するものであるとする立場の根拠となっていると考えられる。この立場は、科学的な「『生命』の定義」のもつ自己矛盾を注意深く避ける行為である。

だがもしや、自分の開発したものが一人歩きはじめ、開発者の制御を逃れて行くようなことがあれば、それは人工物が生命を得たことと言えないだろうか。コンピュータ・ウイルスやウェアムなど、生命的な響きをする命名がなされるのはそういった理由である。自己増殖するロボットはまだ開発されていないが、このような開発者を出し抜くような機械が生まれてくると、話が変わってくるのかもしれない。人工的に造られた DNA、遺伝子操作、自己増殖マ

イクロマシ、多くの「生命化」した機械への警告、警戒が存在するのは、「自己の支配下」から飛び出し、「他者」化した新しい「生命」への恐れでもある。SF的な世界と考えられるかもしれないが、現代においてもすでに、多くのコンピュータ学者でない人々は、コンピュータ、あるいはコンピュータを支配する組織や人々とのバータのなかで暮らさなければならない。コンピュータの無かった時代に生まれた高齢者と、そうではない若者層という比較、コンピュータの普及した社会とそうでない社会といった比較の中で、デジタル・ディバイドという言葉が生まれた。しかし事態はそう簡単ではない。コンピュータに慣れ親しみ、なんなく操作できる若者が、コンピュータを有効に支配し、操作しているかといえば必ずしもそうではない。慣れ親しむということ事態が知らず知らずのうちにコンピュータから支配されていることになっている。このような現代におけるデジタル環境のなかでは、コンピュータに対して生命的なイメージを抱かざるをえないと言えるかもしれない。実際どんな「生命」も我々の存在を脅かす敵となり得るのである。

「生命」の定義は様々なものがあっても良いし、また時代によって揺れ動く。おそらく未来においては、機械と生命の住み分けは我々の思っている以上に変化しているだろう。しかしその中でも普遍的に成り立つのは「生命」の再帰的な定義ではないだろうか。私という一つの「生命」からリカーシブに「生命」を定義して行く。その中に西垣流の「生命」も、ワトソン・クリック型の「生命」も、お互いは矛盾すれど再帰的な「生命」の認識と定義の連鎖の中で共存している。アラン・チューリングが提起した人工知能の定義「チューリング・テスト」も、この「生命」の再帰的な定義によく当てはまる。

5.2 科学的な確からしさ

科学的方法論の基礎には論理学と数学がある。論理学や数学の大きな特徴は、現実世界を想定しなくても成り立つ方法論である。これらの学問の真偽判定の基準は唯一、矛盾を起こすか起こさないかである。矛盾を起こさなければなんでも許されるのである。これが、数学に次いで論理的であるといわれる物理学になるとどうであろうか。地球上の我々の身の回りの現象であればそれは直感的に正しいかどうか分かるが、遠い宇宙の話、あるいは、極小の世界となると

実は理論の検証可能かどうかという問題が生じてくる。検証され得ないものは存在しないのか、存在しないのか、きわどいところである。物理学の世界では人間原理という方法論がある。多くの物理定数がなぜそのような値になっているのか、ということの説明するために、実は我々の住む宇宙以外にも多数の宇宙があり、そこではそれぞれ異なる物理定数を持っているというのである。しかし、その中で、人間が生まれ、このような科学的思考が可能となるためには、この宇宙の物理定数の組み合わせしかなかった、という。ありとあらゆる組み合わせがあり、その中でたまたま我々がいるのはこの一つ、という論法だ。ランダムにおこる突然変異のなかでたまたま環境に適合したものが後世に伝えられるという進化論の自然選択説も一種の人間原理である。

これは、乱高下する株価をすべてズバリと言い当て、それを信じ込んだ人を鴨にして儲ける、ある詐欺師の手法に似ている。詐欺師はまず、鴨となる人々1000人の電話番号を用意する。これを500人ずつ二つのグループA,Bに分ける。Aのグループには

「来週の株価は上がります。我々の開発した世界初の株予測システムの計算結果ですの的中するはずです。」と電話をかける。一方、Bのグループには、

「来週の株価は下がります。我々の開発した世界初の株予測システムの計算結果ですの的中するはずです。」と電話をかける。1週間後、実際の株価をみて上がっていればBのグループを切り捨て、下がっていればAのグループを切り捨てる。予想が一致したグループを再び二つのグループにわけてそれぞれに以下のように電話をする。「さて、予想通りでしょう。株価予測システムの威力を見ましたか。では来週の株価をもう一度予測しましょう。来週の株価は上がります。」もう一方のグループには、「さて、予想通りでしょう。株価予測システムの威力を見ましたか。では来週の株価をもう一度予測しましょう。来週の株価は下がります。」

一週間後、外れた方のグループは切り捨てる。当たったグループの鴨に、「今回も予想が的中しました。株価予測システムの確からしさが証明されたはずです。是非我々に投資を。」と呼びかける。このようにすれば250人の鴨を得ることが出来る。母集団を大きくとれば、原理的にいくらでも、長期間的中させることができ、それだけ信用度を上げることができる。

乱雑な数列の中からはあらゆる法則的数列が抽出できるということを利用し

た詐欺である。これは便利な論法であるが、これを多用すると、証明できないものまでも証明できてしまう危険性がある。こういった危険な証明の背後にはある共通した構造がある。母集団の大きさの限度以上に、それを超えて論法を適用した行き過ぎがある。

別の例として『生命とはなにか』[2]に出てくるケルビン卿のこんな話はどうだろうか。いま仮に、コップ一杯の水の分子に全て目印をつけることが出来たとする。次にこのコップの中の水を海に注ぎ、海を十分にかきまわして、この目印のついた分子が七つの海にくまなく一様に行き渡るようにする。もし、そこで、海の中のお好みの場所から水をコップ一杯汲んだとすると、その中には目印をつけた分子が約百個見つかるはずだ。しかもシュレディンガーに言わせれば95%信頼区間が±10個程度である、という。

さていこの話は本当だろうか。我々は実際にこれを証明するにふさわしい、あるいは忌まわしい出来事を経験している。福島、チェルノブイリの事件は放射性元素でマーキングされた気体分子を大気中に放出した。だが、世界中の大気が一様に行き渡るようなことはなかった。どんなに高く成層圏に到達するような放出をしたとしても、その濃度は地域によってばらつきを見せる。このケルビン卿とシュレディンガーの話は、「一様に行き渡るようにすれば」と言っている。しかし、話の帰結として述べられている、「コップ一杯に百個見つかる」という話は、平均値の話である。平均値とは、一様性を特徴づける一般的な性質に過ぎない。最も簡単に言ってしまうとこの話は「一様に行き渡るようにすれば一様となる。」というトートロジーなのだ。

巨大な母集団の極限には「無限」という数学的概念のもつパラドックスがある。古くはアキレスと亀のパラドックス。4.7節で示した無限級数の例も一例である。近代、カントールの集合論においてバートランド・ラッセルのパラドックスがある。数学においては矛盾が見いだされなければ問題ない訳で、これらの無限という概念も、論理的にきちんと整理し、矛盾を消し去ればことなきを得る。だが、無限は現実世界には存在しない概念である。乱雑性、および無限性、また、無限という概念と密接につながっている解析学的な連続性は、秩序や離散性と対峙して人間の認識に誤作動をおこす。数学者ラプラスが、現在のすべての粒子の位置と速度が分かれば、未来も過去も完全に予測できると言明した。古典力学を認めればこの理屈は論理的には正しい。だが、実際問題、全

ての粒子の位置と速度をどうやって知るのかと考えれば不可能である。このラプラスの悪魔問題も無限性と連続性の誤作動である。

論理学や数学の世界においては無矛盾であることが唯一の正当性であった。現実世界を対象とする学問で必要となってくるのは「説得力」である。近年コンピュータの発達により、様々な分野でシミュレーションが盛んに行われるようになった。たとえば、地球温暖化の予測がなされている。気象学における様々な研究の蓄積を背景とし、過去の気象や現代の気象をシミュレーションした結果と、実際の測定結果と比較し、差が許容範囲内であることが示せる。これは、実際のデータに基づいているので、純粋に論理学や数学の範疇の真偽性で判断できる。ここからが問題である。これを未来に適用することが出来るだろうと推察できるかどうかである。「過去に適用出来たことを未来に適用できる」という命題が正しいかどうかは論理学や数学では導出できないはずであるが、これを認めることは自然である。類似の現象への拡張適用であり、数学的には外挿と呼ばれる。数学的でない、哲学的帰納法である。これがシミュレーションのもつ説得力である。

もう一つ例を挙げよう。生物学的な生命の特徴として自己複製が挙げられる。我々が生きている時間の範囲、あるいはそれを若干拡大した数千年くらいの間で考えるならば、確かに、生物は自己複製し、周期的な繰り返しを続けているように見える。だが、数万年～数十万年に時間を拡大したらどうだろうか。進化論からすれば、同じことといっても少しずつ個体に変化している。完全な複製ではないことが分かる。しかし、少なくとも四十億年前の生物誕生から現在に至るまでは、変化しつつも「擬似的な」あるいはポジティブに捉えるならば、「進化を伴った」自己複製が続いてきたと言えよう。話を未来に転じ、未来もこれがずっと続くだろうか。「過去に適用出来たことを未来に適用できる」という命題が正しいと認めるならば、続くであろう。過去四十億年続いたのだから、未来四十億年は続くであろうと言えるかもしれない。では、八十億年、百二十億年と続くか、と問われればどうだろうか。適用命題の能力を超えてはいないだろうか。論理学や数学の正当性の適用範囲はすべてであるのに対し、説得力というのは適切なスケールと状況の中で適用して初めて説得力が生まれる。一方で、適切なスケールをこえた適用が一人歩きを始めると、それは問題だ。

我々社会が「科学的方法論」をもって未知なるものを探求し「自己の支配下」

を増やして行こうという行為は止めることは出来ない。また、様々なパラダイムの変化によって機械と生命の境界線が仕切り直されることもあり得ることだ。しかし、図 4.13 の、乱雑性と秩序の循環、連続と離散の循環は、どんなに探求が進んでも、乱雑性と連続性の中に、未知なる他者、自己の支配下となり得ない他者がいる、ということを示している。

5.3 おわりに あるいは $\lim_{t \rightarrow -\infty} e^t = 0$

最後に、最近取りざたされている 2045 年問題について意見しておこう。この問題提起の根拠となっているのがムーアの法則である。半導体の集積密度やハードディスクの容量など、様々な工業技術の成長が、指数関数的であるという、ゴードン・ムーアが 1968 年に提唱した経験則である [47]。例えば半導体の集積密度は 1.5 年で 2 倍に成長してきた。当然半導体メモリの容量も同様のペースで成長し、CPU の計算測度や機能も、半導体の集積度によるので指数関数成長をする。また、ハードディスクは半導体記憶装置ではないが、市場において半導体メモリとの競争にさらされている訳で、結果、同様の成長をなすことになる。だが、原子 1~2 個分に相当するような小さなトランジスタは、集積度の限界であり、それ以上は小さくならないことから、ゴードン・ムーア自身、このような指数関数的成長はどこかで頭打ちになるだろうと言っている。

レイ・カーツワイルは、集積度や計算測度について、リレーや真空管という古い時代のテクノロジーも考慮して考えると、そこにもやはり指数関数的成長が認められると主張する。新しいテクノロジーが導入され技術革新が行われることによって、指数関数的成長が維持されている。このように考えれば、確かにここ数年のうちにトランジスタの集積度は行き詰まってしまうが、なんらかの技術革新によって指数関数的成長は未来も維持されるに違いないと予測する。かれはこれを単なる容量の増大化にとどめず、人間の知能や文明の進化にまでも適用し、それら全体が指数関数的な成長をすると考えた。

この成長が順調に続くならば、2045 年ごろにコンピュータの集積度や容量、能力までもが、人間のそれを追い抜くというのである。これは、技術的特異点と呼ばれる。ノーベル賞級の発見はほとんどコンピュータが行う世界、様々な

労働はコンピュータが行い、人間は大量失業か、あるいは悠々自適の生活か……。このような話は想像するといろいろ面白い。ここで問題にしたいのは、このような話が荒唐無稽かそうでないかということではない。いままでに、いくつかの能力においては、コンピュータは飛躍的に人間を追い抜いてきた。また、技術革新は確かに我々の想像を超えていくだろうし、その根拠となりそうな現実も存在する。ムーアやカーツワイルの議論もそれなりの根拠があつてのことだ。私が問題としたいのは、話の前提にある種の偏りがあるということなのだ。

まず、知能という問題である。2045年問題ではスイッチング素子の数と知能が同列に議論されている。だが知能とはなにかという問題は様々な学問において未だなお解決していない問題である。ことに、知能という問題は、文化や文明と深く結びついており、それは生物個体があつた性質だけでは説明不十分な問題なのだ。もう一つ人間を追い抜くという技術の発展史というものも、歴史として捉えるならば、議論の上でどのような史観を持ちうるかが重要な鍵となってくる。

このようにしてみれば、2045年問題は数直線的な思考をしていることは明らかだ。ゴードン・ムーアのトランジスタの集積度や、それを、リレーや真空管に拡大するところまでぐらいは良いのだ。だがこの数直線的な思考を知能や技術史の問題へと展開する時（2045年問題の面白みはそこにあるとも言えるのだが。）、植民地主義的な史観を彷彿とさせられる。遅いシステムと速いシステムがあり、遅いシステムはやがて速いシステムに置き換わる。小容量と大容量、といったことも同じである。それと同様に劣った知能と優れた知能があり、劣った知能は優れた知能に置き換わって行く。2045年に技術的特異点を超えた優れたコンピュータが出現するならば、そのコンピュータはまず次のように言うだろう。「2045年を論ずる前にまずカーツワイル自身の頭の時計を十九世紀的思考から二十世紀的思考へスイッチせよ」二十一世紀的思考をするまえにだ。

文化史的、進化論的な時間の進み方は地図の木のような樹形図である。あらゆる事象が分岐を繰り返しながら多様性を繰り返して行く。確かに、そのごく一部の局所的な部分を見れば直線的かもしれない。だがそれを全体へ拡大するのは哲学的帰納法の過剰適用である。いまやコンピュータは直線的であるより、複雑に絡み合ったネットワークであり、その中をいかにブラウズするかということが先進的である。コンピュータ・サイエンス的にもなにか古めかしさを感じ

じる。

ノーベル賞をとるような人工知能は遠からず実現するだろう。なぜならば、そのような知能は賞賛的であり、誰もが欲しいものだからである。どこかの国家が、あるいはなにがしの巨大企業が、大枚をはたいて完成させるだろう。2045年問題の興味があるのは数直線の右端である。本質的に造ることが出来ない人工知能は数直線の左側にある。指数関数的成長の左端はゼロに収束する。速度が0、集積度が0、役に立たないものである。

全てを失い砂の中に散って行く、映画『道』のザンパノは、ノーベル賞的人工知能では実現しえない。理由は簡単である。優秀な社会はザンパノのような人間を求めないからだ。しかし2045年になっても、人の心にはザンパノが宿っているだろう。そして、知能や生命の本質はどうもそこにある。生命は死の存在を思い知ったとき「自己」を「他者」に託すことを覚える。継承することはここからはじまる。教育することもここから始まる。文化が生まれ、文明が生まれる。高度な機械も、そして情報も、こうして生まれてきたものなのだ。

参考文献・関連図書

- [1] 西垣通. 基礎情報学 生命から社会へ. NTT 出版, 2004.
- [2] シュレーディングー著, 岡小天・鎮目恭夫訳. 生命とは何か. 岩波書店, 2008.
- [3] 福岡伸一. 生物と無生物のあいだ. 講談社現代新書, 2007.
- [4] H. Lodish, Chris A. Kaiser et.al, 石浦 章一 et.al. 分子細胞生物学 第 6 版. 東京化学同人, 2010.
- [5] John von Neumann. *First Draft of a Report on the EDVAC*. Between the United States Army Ordnance Department and the University of Pennsylvania Moore School of Electrical Engineering University of Pennsylvania, June 1945.
- [6] Turing. A.M. Computing machinery and intelligence. *Mind*, Vol. 59, pp. 433–460, 1950.
- [7] J.R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, Vol. 3, No. 3, pp. 417–457, 1980.
- [8] Langley. P., Simon. H.A., Bradshaw. G.L., and Zytkow. J.M. *Scientific Discovery: Computational Explorations of the Creative Process*. MIT Press, Cambridge. MA, 1987.
- [9] Douglas C. Engelbart. Augmenting human intellect: A conceptual framework. Technical Report AFOSR-3223, SRI Summary Report, Oct. 1962.
- [10] M. Spreitzer and M. Theimer. Computer-augmented environments : Back to the real world. *Communication of ACM*, pp. 24–97, July 1993.

- [11] M. Weiser. The computer for the 21st century. *Scientific American*, Vol. 265, No. 3, pp. 94–104, 1991.
- [12] I.A. Sutherland. A head-mounted three dimensional display. In *Proceedings FJCC 1968*, pp. 757–764, Washington DC, 1968. Thompson Bools.
- [13] L. Underkoffler and H. Ishii. Urp: A luminous-tangible workbench for urban planning and design. In *Proceedings of Conference on Human Factors in Computing Systems (CHI'99)*, pp. 386–393, Pittsburgh, Pennsylvania USA, May. 1999. ACM Press.
- [14] NISHIO Shuichi, DOI Miwako, TSUBOUCHI Takashi, and KOTOKU Tetsuo. Standardization of robotic localization service. *The Journal of the Institute of Electronics, Information and Communication Engineering*, Vol. 91, No. 5, pp. 374–379, 5 2008.
- [15] TOKUDA Hideyuki and NAKAZAWA Jin. Basics of networked appliance at home. *IPSJ Magazine*, Vol. 42, No. 11, pp. 1063–1069, 11 2001.
- [16] M. Weiser. Some computer science issue in ubiquitous computing. *Communication of ACM*, Vol. 36, No. 7, pp. 75–84, Jul. 1993.
- [17] B. Shneiderman. *Designing the User Interface – Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 1998.
- [18] S. Brave, H. Ishii, and A. Dahley. Tangible interfaces for remote collaboration and communication. In *Proceedings of CSCW'98*, pp. 169–178, Seattle, Washington USA, Nov. 1998. ACM Press.
- [19] P. Frei, V. Su, B. Mikhak, and H. Ishii. curlybot: Designing a new class of computational toys. In *Proceedings of Conference on Human Factors in Computing Systems (CHI'00)*, pp. 129–136, The Hague, The Netherlands, Apr. 2000. ACM Press.
- [20] H. Ishii and B. Ullmer. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of Conference on Human*

- Factors in Computing Systems (CHI'97)*, pp. 234–241, Atlanta GA USA, Mar. 1997. ACM Press.
- [21] B. Piper, C. Ratti, and H. Ishii. Illuminating clay: A 3-d tangible interface for landscape analysis. In *Proceedings of Conference on Human Factors in Computing Systems (CHI'02)*, Minneapolis, Minnesota, USA, Apr. 2002. ACM Press.
- [22] P. Wellner. The digitaldesk calculator: Tangible manipulation on a desk top display. In *Proceedings UIST '91 ACM Symposium on User Interface Software and Technology*, pp. 27–33, Hilton Head, S.C., Nov. 1991. ACM.
- [23] WADA Koji. Probe car information system project. *IPSJ Magazine*, Vol. 43, No. 4, pp. 363–368, 4 2002.
- [24] IZUMI Michiko, WAKIKAWA Ryuji, KAWAKITA Yuusuke, and AKIYAMA Yoshikazu. Internet ITS project [overview]. *IPSJ Magazine*, Vol. 43, No. 4, pp. 369–375, 4 2002.
- [25] TOKITSU Naoki and TAKAHASHI Kunihiko. Internet ITS project (verification test). *IPSJ Magazine*, Vol. 43, No. 4, pp. 376–385, 4 2002.
- [26] NAKAMURA Jiro, MARUO Yasuko Y., and OGAWA Shigeo. Visualization of environments by ubiquitous sensing system. *The Journal of the Institute of Electronics, Information and Communication Engineerings*, Vol. 90, No. 11, pp. 942–948, 11 2007.
- [27] EGUCHI Ikuko. A ubiquitous system for improving human mobility by making environment virtually barrier-free. *IPSJ Magazine*, Vol. 45, No. 9, pp. 912–917, 9 2004.
- [28] NAKASHIMA Hideyuki, KURUMATANI Koichi, and ITOH Hideo. Supporting a society with ubiquitous computing. *IPSJ Magazine*, Vol. 45, No. 9, pp. 907–911, 9 2004.

- [29] 高野明彦, 吉見俊哉, 三浦伸也. 311 情報学 メディアは何をどう伝えたか. 岩波書店, 2012.
- [30] 海野一隆. 地図の文化史 世界と日本. 八坂書房, 1996.
- [31] John Noble Wilford. *The Mapmakers*. Alfred A. Knopf. Inc., 1981. 日本語訳 鈴木主税「地図を作った人びと」河出書房新社.
- [32] 辻垣晃一, 森洋久. 森幸安の描いた地図. 国際日本文化研究センター, 2001.
- [33] UESUGI Kazuhiro. Networks for collecting maps in the 18th century. *Geographical Review of Japan*, Vol. 80, No. 13, pp. 823–841, 2007.
- [34] 有坂道子, 藤井譲治・杉山正明・金田章裕 (編). 木村兼葎堂と地図. 大地の肖像 – 絵図・地図が語る世界, pp. 388–409. 京都大学学術出版会, 2007.
- [35] 上杉和央, 藤井譲治・杉山正明・金田章裕 (編). 本居宣長の地図利用 – 日本図・世界図を中心に. 大地の肖像 – 絵図・地図が語る世界, pp. 368–387. 京都大学学術出版会, 2007.
- [36] 上杉和央. 地図作成者としての森幸安. *歴史地理学*, Vol. 47, No. 4, pp. 13–33, 2005.
- [37] 上杉和央. 森幸安の地誌と京都歴史地図. 平安京 – 京都 都市図と都市構造, pp. 99–121. 京都大学学術出版会, 2007.
- [38] 倉野憲司 (校注). 古事記. 岩波文庫, Jun. 1963.
- [39] 坂本太郎, 家長三郎, 井上光貞, 大野晋 (校注). 日本書紀 (全五巻). 岩波文庫, 1994/1994/1994/1995/1995.
- [40] *Google Earth*. <http://earth.google.co.jp/>.
- [41] International Lettriste. Potlatch, novembre 1957.
- [42] 伊藤昌亮. フラッシュモブズ. NTT 出版, 2011.
- [43] 伊庭斉志. 人工知能と人工生命の基礎. オーム社, 2013.

- [44] ノーバート・ウィーナー著, 池原止戈夫, 彌永昌吉, 室賀三郎, 戸田 巖翻訳.
ウィーナー サイバネティックス —— 動物と機械における制御と通信. 岩波書店, 2011.
- [45] 杉本信正. 非整数 $1/2$ 階微分で与えられる履歴を伴う非線形波動. 数理解析研究所講究録, Vol. 740, pp. 1–26, Jan. 1991.
- [46] 板倉昭二. 霊長類における「心の理論」研究の現在. 霊長類研究, Vol. 15, No. 2, pp. 231–242, 1999.
- [47] G.E. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, Vol. 86, No. 1, pp. 82–85, Jan. 1998.

情報とは何か

発行者 総合研究大学院大学・学融合推進センター研究費助成プロジェクト
「日本における諸科学の編成と基礎概念の検討—分離統合の有効性をさぐる」
平成 24 - 25 年度代表 鈴木貞美

国際日本文化研究センター 文化資料企画室 森 洋久

非売品

平成 26 年 3 月 27 日発行

大学共同利用機関法人 人間文化研究機構 国際日本文化研究センター
〒 610-1192 京都市西京区御陵大枝山町 3 丁目 2 番地
電話 (075) 335-2222 (代表) ホームページ <http://www.nichibun.ac.jp/>
編集 (有) K&K 事務所
〒 151-0005 東京都渋谷区千駄ヶ谷 2-34-9-406
制作 (有) アートマン
〒 145-0004 東京都江東区森下 4-24-8

© 国際日本文化センター 2014 Printed in Japan
ISBN-978-4-901558-71-6