Uncertainty Molecular Motor

第三章 可逆コンピュータからの発想

2006/08/23

本章ではコンピュータという、これまでとは別の切り口から「不確定分子モーター」へのアプローチを行う。
なぜコンピュータなのか。それは熱運動の処理の本質が「情報処理」だからに他ならない。本章の出発点は次の問いかけにある。

コンピュータで演算を行う為には、最低限どれだけのエネルギーを必要とするか？

情報処理とは、初期に与えられた状態から必要となる知識を抽出し、残りをガベージ（残渣）として破棄する過程である。そして情報処理に必要なエネルギーとは、即ちガベージの破棄に必要なエネルギーである。
本章では、ガベージの破棄が「タイミング（時間）をずらした、ただ1個の信号によって行える」ことを示す。そこから熱運動の情報処理を行うコンピュータとしての「不確定分子モーター」を描き出そう。

第三章 可逆コンピュータからの発想

コンピュータの世界

2006/08/23

最も古い物理学である古典力学は、ゆらぎ、散逸、不確定性といった「不完全な要素」を含まない、理想的な世界像を描き出した。物理学の故郷である「天体運動の世界」は、この理想的な世界に近いものの1つである。天体運動の解析からおよそ250年の後、物理学はもう1つの理想郷を見出すことになる。それは「コンピュータの世界」である。「コンピュータの世界」は、古くて新しい。

日常的に手の届く「地上界」の運動が常に不測の事態、偶然、ゆらぎに撹拌されるのに対し、遙かなる「天上界」は予定調和に従って運行する。天体とコンピュータの第一の共通点は、決定論的であること。偶然や未知なる要素が全く介在しない、計算通りの世界であることだ。しかし、偶然や誤差を全く含まない真に計算通りの事象などこの世に在りはしない。それは天上界が遙かに手の届かぬ理想郷であり、我々の住む地上界が不条理に満ちていることに似ている。

天体とコンピュータの第二の共通点は、過ぎゆく時間の観念を持たないことである。地上界の運動が常に摩擦、ゆらぎ、散逸の下にすり減ってゆくのに対し、天上界の運動は終わることのない軌跡を描き続ける。天上の運動が地上と一線を画すのは、それが永久であること、「死なない」ことである。天体と同様、コンピュータの世界における時間とは1つのパラメータに過ぎない。そこに、過去から未来へと移りゆく推移の観念は希薄である。

コンピュータに時間の観念が欠けている、と聞かされても納得のいかない方も多いかもしれない。しかし、確かにコンピュータだけで閉じた世界を形作ったなら、その中に不可逆過程の入り込む余地はないのである。不可逆過程は、コンピュータの世界が地上界の産物である熱ゆらぎに接したときに初めて生じる。ここで言うコンピュータとは、正しくは可逆コンピュータのことを指す。電力を消費し、発熱しながら稼働する実在のコンピュータは不可逆であり、どちらかと言えば地上界に近い。

決定論的な古典力学と実際的な熱・統計力学の間には互いに相容れない相克をはらんでいる。同様の相克がコンピュータの問題を扱うときにも表れる。コンピュータの内部は決定論的な力学に従う。そして決定論的なコンピュータが外界の熱ゆらぎに接するとき、初めて過ぎゆく時間の観念が生じるのである。物理の中には、純粋に理論と観念だけで完結した領域と、幾分現実的で理詰めでは尽くせない領域がある。2つの領域を対比させると次の様になるだろう。

「天上界」	「地上界」
純粋、完全、予定調和	不確定、不完全、カオス
古典力学、可逆コンピュータ	熱・統計力学、エントロピー
現実の世界から遠い	現実の世界に近い
理論的に扱い易い	理論的に扱い難い

天上界とは、良く言えば美しく完結した観念の世界であり、悪く言えば「学者さんが考えそうな」幾分現実離れした世界のことだと思えば良い。ともすると観念的な理屈は、それが観念的だという理由だけで疎んじられる傾向がある。良くも悪くもコンピュータとは観念の塊の様な代物なので、最初に食わず嫌いを起こすとコンピュータの全てが理解不能となる。（その意味で向き不向きがあると思う。）コンピュータの理屈にはどうも馴染めないと感じたら、とりあえずこれは「あっちの世界」の話であり、「こっちの世界」に持ってくるには相応の解釈が必要なのだと思って欲しい。

・純粋な理論上のコンピュータは、いわば遠い別世界にあるということ。
・それを身近な世界に落とし込むには、2つの世界間をつなぐ操作が必要となること。

観念的なコンピュータの話を始めるにあたって、以上の点に留意しておいて欲しい。

第三章 可逆コンピュータからの発想

可逆コンピュータとは

2006/08/23

コンピュータで演算を行う為には、最低限どれだけのエネルギーを必要とするか？ 答えはゼロ。理論的には、演算そのものに必要なエネルギーはいくらでも小さくすることができる。なぜ、ゼロで済むのか。それは、コンピュータを全て可逆な素子から作り上げることができるからだ。可逆な素子で構成されているということは、演算結果から演算前の元の状態に戻せるということである。もし演算の途中でエネルギーを失っているのであれば、結果を元の状態に戻すことはできない。従って、可逆な素子から成るコンピュータは余計なエネルギーを一切必要としないことになる。

それでは、可逆な素子から成るコンピュータとは一体どのようなものであろうか。可逆な素子とは、演算結果である出力から入力が再現できる素子のことを指す。例えばNOTという演算は、結果に対してもう一度NOTを施せば元の状態を再現できる。ゆえにNOTは可逆である。一般的にコンピュータはAND、OR、NOT によって構成されている。※ この3つのうちNOTは可逆だが、残るANDとORは可逆ではない。もしANDの結果が0であった場合、その入力は0と1の組だったのか、0と0の組だったのか分からないからである。同様に、ORの結果が1だった場合は、0or1だったのか、1or1だったのか分からない。なので、可逆なコンピュータを作るには AND、ORではない、もっと別の素子を用いなければならない。

考えてみると、AND、ORの様に2本の入力に対して出力が1本だけの素子は可逆にはなり得ない。可逆な素子は2本の入力に対して2本の出力を持つに違いない。ここで、可逆素子NOTを拡張した制御付NOT(Controlled NOT)という素子を考えてみよう。制御付NOTは、通常のNOTにもう1本の制御入力を付け加えたもので、2本の入力と2本の出力がある。そして、制御付NOTは制御入力に1が入力されたときにだけNOTとして働き、信号の反転を行う。制御入力が0の場合には何も行わず、入力信号はそのまま出力される。制御付NOTの残るもう1つの出力は、制御入力をそのまま出力する。

[制御付NOTの入出力表] (Aが制御入出力)

Ain	Bin		Aout	Bout
0	0		0	0
0	1		0	1
1	0		1	1
1	1		1	0

この制御付NOTは、出力から入力を再現することができるので可逆である。単純なNOTと同様、制御付NOTは2回繰り返すと元の状態に戻る。

さて、いま考えた制御付NOTだけでコンピュータが作れるかというと、残念ながらこれだけでは役不足である。制御付NOTだけで（AND、OR、NOTで構成されたコンピュータと同等能力の）コンピュータを作ることはできない。コンピュータを作るには制御付NOTをもう1段階パワーアップする必要がある。そこで、制御付NOTにもう1本制御線を増やした二重制御付NOT(Controlled Controlled NOT)というものを考えよう。二重制御付NOTには、3本の入力と3本の出力がある。このうちの2本は制御入力である。二重制御付NOTは、2本の制御入力が両方とも1だった場合のみNOTとして働き、信号の反転を行う。

[二重制御付NOTの入出力表] (A,Bが制御入出力)

Ain	Bin	Cin		Aout	Bout	Cout

0	0	0		0	0	0
0	0	1		0	0	1
0	1	0		0	1	0
0	1	1		0	1	1
1	0	0		1	0	0
1	0	1		1	0	1
1	1	0		1	1	1
1	1	1		1	1	0

二重制御付NOTも、2回繰り返すと元の状態に戻る可逆素子である。上の説明の「両方とも1だった場合のみ」というくだりがANDと類似していることに注意されたい。二重制御付NOTはANDと同等の能力を有しているのである。上の入出力表でCinを0に固定すれば、Cout は Ain AND Bin となっている。※

二重制御付NOTはAND演算を行うことができる。また、二重制御付NOTはもちろん単なるNOT演算を行うこともできる。さらに、ORという演算はANDとNOTを組み合わせて作り出すことができる。

NOT((NOT A) AND (NOT B)) = A OR B

つまり二重制御付NOTからは AND、OR、NOT 3つの素子が作り出せる。そしてコンピュータは AND、OR、NOT の3つから成り立っているのだから、結局のところ二重制御付NOTのみによってコンピュータが作り出せるのである。二重制御付NOTだけで作ったコンピュータは可逆である。いまここに二重制御付NOTだけで作ったコンピュータがあったとしよう。その隣に、元のコンピュータと全く逆に二重制御付NOTを配置した「反コンピュータ」を置く。もとの「正コンピュータ」の演算結果を「反コンピュータ」に入力すれば、反コンピュータは正コンピュータの全く逆の道筋をたどって演算前の状態を再現するはずだ。

コンピュータを構成するには二重制御付NOTで十分なのだが、その上さらに、入力する1の数と出力する1の数が等しくなるような可逆素子を考え出した人たちがいる。フレトキンと、トフォリという人たちである。彼らの考案した素子はフレトキン・トフォリ・ゲートと呼ばれている。（以下FTゲートと略す） FTゲートには3本の入力と3本の出力がある。このうちの1本が制御線となっている。制御入力が1のとき、FTゲートは残る2本の線の値を交換する。制御入力が0のとき、FTゲートは残る2本の線に対して何もしない。制御出力には、制御入力と同じ値が出力される。

[FTゲートの入出力表]（Aが制御入出力）

Ain	Bin	Cin		Aout	Bout	Cout
0	0	0		0	0	0
0	0	1		0	0	1
0	1	0		0	1	0
0	1	1		0	1	1
1	0	0		1	0	0
1	0	1		1	1	0
1	1	0		1	0	1
1	1	1		1	1	1

FTゲートは値を交換するだけなのだから、当然入力する1の数と出力する1の数が等しい可逆素子となる。そして驚くべきことに、このFTゲートだけでコンピュータを構成することができる。

・Binを0に固定すると、Bout = Ain AND Cin
・Binを1に固定すると、Cout = Ain OR Cin
・Binを1に、Cinを0に固定すると、Bout = NOT A

となるので、確かにFTゲートによってAND、OR、NOT の3素子が作り出せる。

これまでに登場した素子は抽象的な1と0の挙動を追っただけで、それが物理的にどのような実体を持つのかについては一切触れていなかった。ここでFTゲートを念頭に置くと、大変面白い物理的イメージを思い描くことができる。それはちょうどビリヤードのように、運動するボールが互いに衝突を繰り返して演算を行うというイメージだ。2個(ときには1個)のボールの衝突は、ボールが存在する場合を1、存在しない場合を0と見なせば、AND素子と同じ意味を持つ。この考え方を延長して、ボールと、うまくボールを跳ね返す反射板を組み合わせてFTゲートを形作ることができる。

(＊＊＊ここは言葉による説明より図に頼ろう＊＊＊)

ボールと反射板によってFTゲートができるならば、それらをさらに組み合わせてコンピュータを作ることができるわけだ。このボールと反射板でできたコンピュータが可逆なことは直感的にもわかるだろう。ボールを転がすことによって演算を行った後、さらにその先に反射板を置いて正確にボールの運動を逆転する。そうすれば、ボールは来た道の逆を正確にたどって、また元の状態に戻ることだろう。

以上のような「ビリヤードボールコンピュータ」のイメージを持てば、当初の問いかけに対して明確に答えることができる。コンピュータで演算を行う為には、最低限どれだけのエネルギーを必要とするか？ 演算自体にかかるエネルギーに下限はない。摩擦のない床の上を転がるボールがいつまでも止まらないのだから、「ビリヤードボールコンピュータ」も最初にある一定のエネルギーを与えておけば後はいつまでも動き続けるのである。(反射板を置いておけば入力と出力の間を往復し続ける。) ボールを運動させるのに必要となるエネルギーに(古典的に考えた場合)下限はないのだから、コンピュータの演算に必要なエネルギーにも下限は存在しない。(いくらでも0に近い値がとれる。) 最も理想的な状況を考えた場合、コンピュータの演算は惑星の運動の様に永久に続くものだったのである。

■ ※
実はANDとNOTからORを構成することができるので、最低限ANDとNOTだけでコンピュータを作ることができる。また、ORとNOTからANDを構成することもできるので、ORとNOTを基本的な素子とすることもできる。ただ、ANDとNOTだけ、あるいはORとNOTだけではどことなく対称性が悪いので、一般にはAND,OR,NOTを基本とすることが多い。

■ ※
ここではANDと同等の能力を持つ二重制御付NOTを考えたが、もう1つ、ORと同等の能力を持つ二重制御付NOTというものも考えることができるだろう。つまり、「どちらか一方が1だった場合に」NOTとして働き、信号の反転を行う素子というものが考えられる。このような ORを内包する二重制御付NOTも、ANDを内包する二重制御付NOTと同じようにコンピュータの基本素子となり得る。

第三章 可逆コンピュータからの発想

ランダウアーの原理

2006/08/23

可逆コンピュータの演算はエネルギーを消費しない。可逆コンピュータは入力情報から答を導き出すのと同様に、答から入力情報を導き出すこともできる。順方向の演算と逆方向の演算は全く同等の重みを持っており、どちらかが優先されることは無い。前節で紹介したビリヤードボールコンピュータを思い描いてみよう。順方向の演算を終えた後、誰かがボールにブレーキをかけない限り、ボールの運動は続く。ボールの向う先に反射板を置いて運動を反転させれば、反転したボールは逆方向の演算を開始する。可逆コンピュータは順方向の演算と逆方向の演算をいつまでも交互に繰り返すことになる。つまり可逆コンピュータには長い目で見た場合、入力から出力へと向かう明確な方向性がない。行きと帰りは全く同じ重みを持つのだから、その動作は過去と未来に対して対称であり、どちらか一方の向きに優先的に流れることはないのである。

実在する不可逆なコンピュータは入力から出力へ、確実に一方向に動作する。なぜ一方向に動作するかというと、不可逆なコンピュータはエネルギーを消費し、エントロピーの増大に寄与しているからである。つまりコンピュータというものはエネルギーを費やした分だけ意図した方向に進むのである。（もっともこれはコンピュータに限った話ではなく、あらゆる装置、現象について言えることなのだが。）ここで次の様な疑問が生じる。双方向に動く可逆コンピュータに、如何なる操作を施せば意図した方向だけに動作させることができるのだろうか。コンピュータを意図した方向に動作させるために必要となる最低限のエネルギーはどれ程だろうか。現在の不可逆コンピュータは、1個1個のトランジスタが動作する度にエネルギーを消費している（発熱している）。しかしスイッチングにともなうエネルギー消費は原理的に必須ではない。それでもコンピュータ全体を眺めたとき、一方向に動作させるためには何処かで必ずエネルギーを消費しなければならない。それは何処か。また、それはどれだけの大きさのエネルギーか。

ビリヤードボールコンピュータを思い描けば、コンピュータを一方向に動作させる操作とは、演算後のボールに「ブレーキをかけること」だと気付く。ボールを確実に、意図した場所で止める操作が必要なのだ。ブレーキをかける操作は実質的なエネルギー消費であり、発熱を伴うエントロピー増大過程である。全てのボールにブレーキをかけるとすれば、演算に必要な全エネルギーは「ボールの数x運動エネルギー」ということになる。これが一方向の演算に必要最低限のエネルギーなのだろうか。実は、工夫をこらせばもっとエネルギーを減らすことができるのである。

それは計算を終えたボールを「動いたままの状態で箱に入れて回収する」という工夫である。ボールが転がってくる経路の先に、入り口のフタの開いた箱を設置する。ボールが箱に入ったところで入口のフタを閉じれば、閉じ込められたボールは箱の中を往復し続けることになる。ボールが入った箱はそのまま次回の演算に使用することができるので、ボールの運動に着目する限りエネルギーの損失は無い。ボールの回収にあたって、フタの開いた箱はボールが出てくる可能性のある全ての出口に対して仕掛けておく必要がある。実際にどの出口からボールが出てくるかはわからない。（分かっているのであれば演算を行う必要がない。）それゆえ、仕掛けた箱のなかのどれにボールが入り、どれが空なのかは予め知ることができない。我々にできるのは、ボールが出てくるタイミングを見計らって全ての箱のフタを一斉に閉じることだけである。（演算にどれ位の時間がかかるかは、回路を組み立てたときに予め予測できる。）結局、以上の操作を通じて我々の手元に残るのは「どの箱に入っているかわからないボール」ということになる。ここでボールを演算前の状態に（逆の演算を行う以外の方法で）戻すには、ボールの入った箱だけを選び出す作業が必要となる。つまりこの場合、演算に使ったエネルギーとはボールの入った箱を選び出すのに必要なエネルギーのことなのである。

ボールの入った（あるいは入っていない）箱とは、通常のコンピュータに対比させると何に相当するだろうか。答えは「メモリー」である。可逆コンピュータは演算自体にはエネルギーを消費しない代わりにメモリーを余分に消費する。必要とする答以外の余分な記録が演算後のメモリー上に残るのである。このことは可逆コンピュータを構成する個々の素子を思い浮かべれば想像がつく。可逆素子は入力線の本数と出力線の本数が等しい。しかし演算の答に必要なのは出力線の一部だけである。残りの大部

分の線は不要なゴミとなる。このような「データのゴミ」のことを、コンピュータの世界では「ガベージ」と呼んでいる。可逆コンピュータと普通のコンピュータを比較すれば、両者の差異はガベージの有無であることに気付く。従って、可逆コンピュータを通常の不可逆コンピュータと同様に一方向に動作させるには「データのゴミ処理」の分だけのエネルギーを投じる必要がある。

それでは「データのゴミ処理」にかかるエネルギーはどれ程の大きさなのだろうか。最も単純なケース、「2つの箱の中のどちらにボールが入っているか分からない」という状態を考えてみよう。どちらか1つの箱の中に確実にボールを押し込めるには、2つの箱を連結した後、全体を半分に圧縮する操作が必要となる。この操作は、ボールという1個の分子から成る気体の体積を半分に圧縮するのだと捉えることができる。1分子の気体を半分に圧縮するのに要するエネルギーEの大きさは kT＊ln(2) である。つまり、1bitの「データのゴミ」に対して最低限E = kT＊ln(2) だけのエネルギーが必要となる。※ 演算を通じてNbitの「データのゴミ」が生じたならば、必要なエネルギーは E = kT＊ln(N) である。

「不可逆コンピュータに必要なエネルギーの大きさは、不要な情報を消去する過程で決まる。」この考え方はコンピュータの熱力学の開拓者の名を冠して、しばし「ランダウアーの原理」と呼ばれることがある。ランダウアーの論旨をまとめると次の様になる。

・論理的不可逆性（logical irreversibillity）と物理的不可逆性（physical irreversibility）には対応がある。
・論理的に不可逆な演算を行うにはエネルギーの消費、即ちエントロピー増大の過程が不可欠である。
・演算そのものは可逆に行うことができて、そこにエネルギー消費はない。
・演算の結果である情報を消去するには必然的にエネルギー消費が伴う。
・従って、不可逆な演算に必要となるエネルギーの大きさは情報消去の過程から見積もることができる。

エネルギーの観点からすると、コンピュータにとって最も重要な過程は「忘れる」ことだったのである。

■ ※

実質的なエネルギー消費なしに、ボールの入った箱と空の箱を合わせる方法は無いものだろうか。例えばボールが箱の中を縦方向に往復しているものに対して、縦方向に圧縮を試みるならば上に記した様なエネルギーが必要となるが、横方向から空の箱をスライドして重ねてみたらどうだろうか。ボールに触れることなく空の箱を操作するだけであれば、エネルギーは不要ではないだろうか。残念ながらそうはならない。2つの箱のどちらか一方をスライドして他に重ねようとしたとき、どちらの箱にボールが入っているかによって、重ねる操作は少しだけ異なったものになる。空の箱を動かすより、ボールの入った箱を動かす方が余計な力を要することだろう。あるいは、2つの箱を左右から同時に同じ力で動かしたなら、どちらにボールが入っていたかによって最後に箱が重なる位置がずれるだろう。結局のところ、エネルギー消費なしに2つの箱を全く同一の状態に持ち込むことはできないのである。

■ ※

上の説明ははしょっている部分があり、以下のボールの速度に関する説明を付け足す必要がある。ボールの速度をうんと遅くすればエネルギーはいくらでも小さくなるのだが、ボールは少なくとも熱ゆらぎを越える速度で運動しなければならない。それゆえポールの最低速度は温度Tの下の気体分子と同程度になってしまう。

■ ※ランダウアーの原理は本当に正しいのか？

本当に「原理」と呼ぶにふさわしいのか、一部には疑問の声もある。
[link:/memo/LandauerIsTrue.html] を参照のこと。

第三章 可逆コンピュータからの発想

ベネットの悪魔祓い

2006/08/23

コンピュータが演算を行う過程で生じるデータのゴミ処理には、エネルギーの消費が避けられない。この考えを応用すれば、なぜ第二種永久機関が実現できないかについての明確な説明を与えることができる。マックスウェルの悪魔のパラドックスに対する解答は、コンピュータのエネルギー消費についての考察から得られたのである。このことを最初にはっきりと述べたのは、C．ベネットという人物である。ベネットの考察については第一章でも簡単に触れたが、ここでいま一度、その内容について再確認しよう。

コンピュータと第二種永久機関、一見すると何の関係も無い取り合わせに思えるかもしれない。しかし、実のところ両者は同一物の表と裏の姿なのだ。物理法則の許す範囲内で情報処理を行うのがコンピュータ、物理法則を越えて情報処理を行うのが第二種永久機関である。なのでコンピュータの物理的限界がどこにあるのかを突き止めれば、第二種永久機関が不可能なことは自ずと明らかになる。それでは、ここで言う物理法則、物理的限界とはどのようなものだろうか。一言で言えば、それは「エネルギーの損失なしに2状態を1状態にすることはできない」ということだ。

ビリヤードボールコンピュータの動作を思い起こしてみよう。各々のボールが定められた軌道の上を転がっている間は、エネルギーの損失は無い。問題は演算を終えた後、行き先が分からなくなったボールを回収する過程にある。ボールを回収するには、ボールがたどり着く可能性のある全ての場所をチェックしなければならない。全ての場所に箱を置いてボールを回収したとしても、ボールの入った箱と空の箱とを1つにまとめる段階でエネルギーを要する。これは前節に述べた通りである。箱という仕掛けを用いず直接的に、全てのボールの軌道を共通の「回収ポイント」に向わせれば良いと思われるかもしれない。しかし、異なる複数本の軌道を1本にまとめ上げる段階で、やはりエネルギー消費が避けられない。これは第2章で考えた「合流タイプ」に相当する。つまり、箱であれ、メモリーであれ、軌道であれ、「どちらにあるか分からない複数の状態」を「1つの状態」に戻すには、相応のエネルギーを必要とするのである。

ところで、「行き先が分からなくなったボール」というのは「行き先が分からなくなった分子」と一脈通じる節がある。例えばエンジン（熱機関）は、小さく圧縮された気体が膨張する過程を通じて仕事を取り出している。小さく圧縮された気体は大きく膨張した気体に比べて、分子の位置をより狭い範囲に特定できる。言い換えれば、大きく膨張した気体の分子はより「行き先が分からなく」なっている。ひとたび大きく膨張した気体を元の小さく圧縮された気体に戻すには、力を加えて押し込めるなどの操作が必要となる。どこにあるかわからない、より広範囲に散らばった対象を、はっきりと一カ所に、あるいは限定的な範囲に押し戻すには、相応のエネルギーを必要とする。コンピュータのボールと気体分子は、こういった共通の原理に従っていたのだ。気体分子の取り得る範囲の広さのことをエントロピーと呼んでいた。（より正確には範囲の広さの対数）コンピュータのボールは気体分子の様に連続的な範囲を取らない。（量子的に考えれば気体分子も必ずしも連続的ではないのだが。）また、コンピュータの取り得る状態数は膨大な数の気体分子とはオーダーが異なる。確かにそういった違いはあるのだが、エネルギーの挙動から考えるとコンピュータの取り得る状態数をエントロピーと呼んでも良いのではないか。いずれにせよ、コンピュータと熱機関は原理的に同じメカニズムによってエネルギーを消費している。その原理を直感的に言えば「ばらばらに散らばったものを元通りに集めてくるのにエネルギーを要する」ということなのである。

ベネットの考察が革新的であったのは、コンピュータの原理が熱機関にもあてはまることを指摘した点にある。ここでいま一度、第1章”シラードの悪魔”を振り返って、ベネットの主旨を追ってみよう。シラードの悪魔とは次の様なものであった。熱運動から有用な仕事を取り出すためには、ばらばらに散らばった分子を何らかの手段を用いて一カ所に（あるいは特定の狭い領域に）集めて来なければならない。端的な場合として、部屋の中にただ一個だけの分子が運動している状況を想定しよう。ある瞬間に部屋の中央に仕切を入れれば、分子は部屋の右半分か左半分かの狭い領域に閉じ込められる。分子は狭い

場所に閉じ込められられたのだから、原理的にはこの状態から有用な仕事を取り出すことができるはずだ。ただし、分子が右にあるか左にあるかを言い当てることができればの話だが。ここで、分子が右にあった場合は操作Rを、左にあった場合は操作Lを実行する装置というものを考える。この装置は、まず最初に分子の左右を見定める観測の過程と、左右2つの観測結果に対応する操作実行の過程、操作R実行部と操作L実行部から成り立つであろう。操作R＝「右側にある分子から仕事を取り出す操作」、操作L＝「右側にある分子から仕事を取り出す操作」とすれば、いずれの場合でも熱運動から仕事を取り出せることになる。このような装置を組み立てるのは物理的に不可能ではない。（第一章のベネットの装置の図を参照のこと）問題は操作を終えた後の状態にある。以下にLとR、両方の操作を表した図を挙げる。

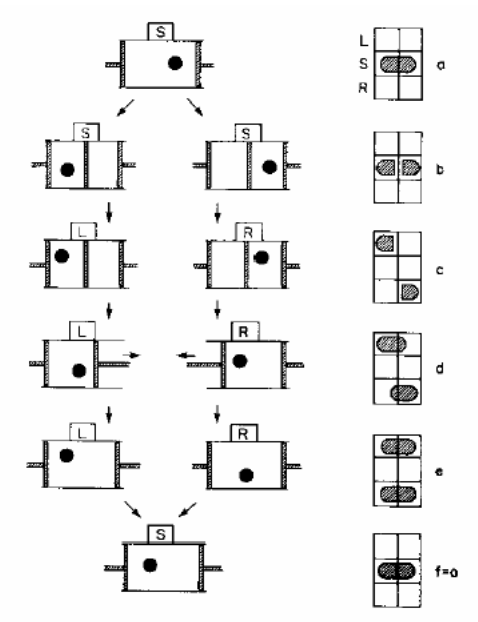

Fig. 12. A one-molecule Maxwell's demon apparatus.

図の右隣に描かれているのは、分子の位相空間を模式的に表したもの、いわば「悪魔の心の中」を描いたものである。位相空間上の領域の広さは、分子から仕事を取り出す段階で2倍に膨れあがる。つまり、この段階で分子のエントロピーは増大する。仕事を取り出し終えた後も、エントロピーは増大したままである。とことが、ここから次のサイクルの最初の状態に戻ろうとすると、領域の広さは半分に縮むことになる。つまりエントロピーは減少したことになる。ということは、仕事を取り出し終えた後の状態から、次のサイクルの最初の状態に移行するまでの間に何らかの矛盾がある。装置全体のことを考えると、操作Rを終えた後の装置は状態Rであり、操作Lを終えた後の装置は状態Lとなっているはずだ。ここで、状態Rと状態Lが具体的に何であるかを追求する必要はない。装置に仕組まれたメモリーの値がRのときに1で、Lのときに0であるとか、あるいは装置内を転がっているボールが軌道Rの上を走っているか、軌道Lの上を走っているとか、何であっても構わない。とにかくRとLが異なる状態であれば、どこかに異なる2状態を1状態に戻す過程が不可欠となる。かくして悪魔の装置は物理的な限界に突き当たる。改めて繰り返すが、物理的な限界とは

「エネルギーの損失なしに2状態を１状態にすることはできない」

ということである。エネルギーの損失とは何であるかというと「明確に運動量が特定できるエネルギーを、運動量がわからない、ばらばらな状態にもってゆく」ことだ。つまりこの物理的限界は次の様にも言い換えられる。「何物かを2状態から1状態にしたならば、その代わりに別の何物かを1状態から2状態にしなければならない。」このとき「2」という状態数を、ある物から別の物に移行するのに必ず一定以上のエネルギーを要するのである。全体として見た場合、「2」という状態数は最初から最後まで変わることはない。少なくとも、状態数が自然に減ることはあり得ない。

エネルギー消費の観点から見たとき、コンピュータと熱機関には共通する物理法則が働いている。それはエントロピーの法則、状態数はけっして減ることはないという法則だったのである。

第三章 可逆コンピュータからの発想

１ボール・チューリングマシン

2006/08/23

前の節ではビリヤードボールコンピュータ、複数のボールが互いにぶつかりあって演算を行う装置を紹介した。ここではさらに新しい種類の可逆コンピュータを考えてみようと思う。ここで扱うのは、たった1つのボールによって演算を行う装置である。ビリヤードボールコンピュータでは AND, OR, NOT といった演算を基本に考えたが、ここではより原理的なチューリングマシンを構成することを考えよう。チューリングマシンとは、本質的にただ1本の道筋をたどる装置である。なのでチューリングマシンがただ1個のボールによって実現できそうなことは、およその想像がつく。以下にその様子を描き出してみよう。

チューリングマシンとは、今日のデジタル・コンピュータの原型に相当する仮想的な装置である。デジタル・コンピュータ上で行われているどんな演算も、原理的には全てチューリングマシン上で行うことができる。チューリングマシンは、無限に長い記録テープ、テープの読み書きを行うヘッド、ヘッドの動作を決定する制御部、の3つから成っている。今日風に言えば、記録テープはメモリーであり、制御部はCPU、ヘッドはメモリーとCPUを結ぶバス(結線)である。記録テープは一定間隔のマス目に区切られていて、それぞれのマス目には値0か1のいずれかが書き込まれている。(書き込む値は1ー9までの数字やアルファベット26文字など多数の記号でも構わないのだが、それらは全て0，1に還元することができる。)ヘッドはテープ上の値0，1を読んだり書いたりしながら、テープの上を行き来する。具体的にヘッドが実行できることは次の5項目である。

・マス目の値を読み取る
・マス目に0を書き込む
・マス目に1を書き込む
・マス目を1つ右に移動する
・マス目を1つ左に移動する

制御部は内部に有限個の状態を持ち、その状態によって次のヘッドの挙動を決定する。(制御部の持つ有限個の内部状態とは「フラグ」のことだと思えば分かりやすいだろう。)チューリングマシンは次のようなサイクルで動作する。

1：現在ヘッドがあるマス目の値を読み取る。
2：読み取った値と、現在の内部状態に従って次の挙動を決定する。
　このとき同時に、内部状態も次の状態へと遷移する。(フラグが立つことがある)
3：挙動に従って(必要であれば)マス目に0か1の値を書き込む。
4：挙動に従って(必要であれば)右か左に移動する。
5：1に戻る。
＊：特別な内部状態「停止」に至った場合、動作を停止する。

たったこれだけのことでコンピュータの持つ演算能力の全てをカバーしているのだ。重要なのは内部状態の遷移規則と、テープ上に最初に書き込んである0，1の配置にある。「内部状態の遷移規則」とはCPUの命令と実行結果、「テープ上に書かれた0，1の配置」とはプログラムに相当する。現代のパソコンを知る我々からすれば、チューリングマシンとはコンピュータの心臓部を取り出したものに見えるだろう。

さて、ここで1個のボールで動作するチューリングマシンを考えるにあたって、まずは用いることができる部品をリストアップしよう。ここで用いることができる部品は次の3種類だけだ。

1：ボールと経路
2：物体の移動
3：条件分岐（合流）

これらの3つの部品だけを用いてチューリングマシンを構成せよ、というのがここでの命題である。（基本的な考え方は第2章、悪魔の装置一号の詳細、構成要素と同じである）以下、それぞれの部品について順番に見てゆこう。

1：ボールと経路

最初に、摩擦のない一本の線上を運動する、ただ1個のボールを考える。ボールの通り道である線のことを「経路」と呼ぶことにする。摩擦がないので、一度動き出したボールは経路上をどこまでも（経路の終わりまで）転がり続ける。経路は空間内の任意の場所に配置することができる。経路は可逆でなければならない。つまり、ボールは経路上を双方向に、順方向にも逆方向にも転がることができる。経路は途中で分岐したり合流したりすることはできない。（分岐や合流は、以下に述べる3：条件分岐だけで行うことができる）

2：物体の移動

運動しているボールと衝突することによって、空間内に配置した特定の物体を移動することができる。静止している物体にボールが衝突すれば、ボールは跳ね返り、物体は移動を開始する。跳ね返ってきたボールを、衝突とは全く逆に移動中の物体にぶつければ、物体の移動はそこで止まる。（これは、先の章で「ハンドル」と呼んでいたものである）物体の移動も可逆である。例えば、ボールが順方向に転がったとき物体がA->Bに移動したとすれば、その反対に、ボールが逆方向に転がったなら物体はB->Aに移動する。

3：条件分岐（合流）

経路上をさえぎる様に物体が置かれていた場合、ボールは物体に反射して運動の向きを変える。これによって、物体が置かれていた場合と置かれていなかった場合で、ボールを異なる2本の経路に導くことができる。つまり物体の置かれている位置を条件として、ボールの運動を分岐させることができる。（これは、先の章で「ゲート」と呼んでいたものである）
この条件分岐の過程を全く逆にたどれば、2本の経路を1本の経路に合流させることができる。逆に言えば経路の合流には、あらかじめ物体が所定の位置に置かれているといった条件が必要なのであって、何も無しに（エネルギーの損失無しに）自然に経路が合流することはない。
条件分岐と、2：物体の移動を組み合わせることによって、コンピュータで扱う様々なロジックを実現することができる。同じ物体であっても、2：物体の移動の際には物体が動くのに、3：条件分岐では物体は動かない、というのはつじつまが合わないと思われるかもしれない。しかし、物体は特定のレールの上をスライドして動くものとして考え、ボールが縦方向から衝突したときは移動、横方向から衝突したときは条件分岐とすれば矛盾は避けられる。（また、条件分岐の場合、物体は必ずしも静止している必要はない。）

以上の3種類の部品、ボールと経路と物体を上手に配置してチューリングマシンを構成しよう。

まず1，0の書き込まれたテープ。これは物体が2状態の位置、例えばLとRのいずれかに置かれているものによって実現できる。1つのマス目が1つの物体に対応しているので、テープ全体は物体がたくさん（無限に）並んでいるものとなる。この物体には名前を付けた方が分かりやすいので、ここでは「メモリービット」と呼ぶことにしよう。テープの読み取りは、この物体、メモリービットにボールを衝突させることに

よって実現する。メモリービットは条件分岐となっていて、例えばメモリービットがLのときにボールは経路Lへ、メモリービットがRのときにボールは経路Rへと導かれる。2つの経路、LとRはそれぞれ別の入り口から制御部へと入る。ボールが入ってくる経路の違いはメモリービットから読み取った値の差異を表している。制御部には幾つかの物体が配置されており、それらの物体の配置によってボールの経路が変化する。つまり「物体の配置＝制御部の内部状態」なのである。制御部内に置かれた物体を、ここでは「フラグビット」と呼ぶことにしよう。フラグビットはボールの分岐条件となっているだけではない。その反対に、ボールの運動によってフラグビットの状態（配置）が変化することもある。制御部の動作の一例として、次の様なチューリングマシンを考えてみよう。

・このチューリングマシンはただ1つのフラグビットXを持つ。
　Xは2つの状態、X＝AとX＝Bの2状態を取ることができる。
・このチューリングマシンは、テープの読み取り値と、Xの状態によって下表の動作を行う。

テープの読取値	０	１
Ｘ＝Ａ	Ｂに遷移、1を書き込む、右に移動	Ａのまま、1のまま、左に移動
Ｘ＝Ｂ	Ｂのまま、0のまま、右に移動	Ａに遷移、0を書き込む、左に移動

このチューリングマシンは何らかの演算を行う目的で作られたものではない。単に簡単な状態遷移と書き込みの例として挙げたものである。状態遷移表は「A,0,左」と「B,1,右」について対称なので（「A,0,左」と「B,1,右」を入れ替えても同じものになるので）どちらか片方の動作だけを追ってみることにする。テープの読みが0で、X＝Aのとき、経路0から制御部に入ってきたボールはフラグビットXに衝突するものとしよう。ボールは衝突によって向きを変え、今度はフラグビットXをX＝Bの位置に移動する。一方、テープの読みが0で、X＝Bのときは、ボールはフラグビットと衝突せず、制御部を素通りする。この時点でボールの経路はXに衝突した方と、していない方の2通りあることになる。これに続く、テープへの値の書き込み、制御部の移動は、どちらも物体の移動の応用なので詳細は省く。移動を終えたチューリングマシンが次のサイクルに入る際に、困った問題が生じる。ボールの経路が2通りに増えているため、次サイクルの開始点も2つ用意しなければならない。経路は1サイクルごとに2倍に増えてゆくので、いつかは装置が破綻することになる。つまりチューリングマシンとは、そのままの形では可逆な部品で構成することができない、不可逆なコンピュータだったのである。このことはチューリングマシンの遷移状態図からもわかる。

＊＊＊ ここに遷移状態図 ＊＊＊

チューリングマシンの動作には複数の異なった状態から同一の状態へと遷移する過程が幾つも含まれている。例えば状態Aに至る過程は、「Aのまま」と「BからAに遷移」の2つがある。こういった異なる状態から同一の状態への遷移は、逆向きにたどることはできない。逆にたどれば、ある同一の状態から複数の異なる状態のどこに戻るべきかか決められないからである。

それでは、チューリングマシンを可逆な部品によって構成することはできないのだろうか。可逆なチューリングマシンを構成するには、増えた経路を1つにまとめる仕組みを追加する必要がある。例えば、ボールと物体によって構成された条件分岐（上記の部品3:）は、逆向きに動かせば経路の合流装置として働く。ここで経路Lと経路Rの2本の経路を経路Xの一本にまとめることを考えてみよう。経路Lから入ってきたボールがそのまま経路Xに直結していたとしよう。このとき経路Rから入ってきたボールも、最終的には経路Xに入るようにしたい。そこで、経路Rの先に1つの物体を置く。ボールは物体に衝突し、その位置を移動させる。物体は経路Xの延長上に移動し、反射後のボールをうまく経路Xに導くような位置にくる。このような移動する物体を1つ置くことによって、経路の合流が実現できるのである。もとのボールが2つの経路のどちら側から来たかという情報は、物体の位置に記録される。上の例では、ボールが経路Lから入ってきた場合、物体は経路X上には無く、ボールがRから入ってきた場合、物体

は経路X上に置かれることになる。言い換えれば、経路が1つになる代わりに、物体の配置が2通りになるのである。ここで経路を1つにするために用いた物体のことを、ボールの経路を記録したという意味で「履歴ビット」と呼ぶことにする。（不可逆な）チューリングマシンを動作させると、各サイクルごとに経路の総数が増えてゆく。なので、経路を合流させるための仕組みは、各サイクルごとに用意しなければならない。結果的にそれは「履歴ビット」をたくさん並べたもの、履歴用の記録テープという形になるだろう。履歴用の記録テープは、最初には既知の値、例えば全て0が書かれている。そして、経路を合流させる度に履歴ビットが使われてゆくので、履歴記録テープには0か1か、いずれかの値が刻まれてゆくのである。

履歴用の記録テープを持つ可逆なチューリングマシンは、およそ次の様な動作を行う。

1：各サイクルを通常通り（上に述べたのと不可逆チューリングマシーンと同じように）実行する。
通常通りのサイクルとは、メモリービットの読み取り -> 内部状態の遷移 -> ヘッドの移動 のことである。

2：1サイクル実行ごとに、経路の取り得る総数は次の数だけ増大する。
経路の総数＝(0，1)の2パターンx遷移パターン数x(右、左、移動なし)の3パターン

3：増大した経路の総数を、履歴ビットを用いて1つにまとめる。
履歴ビット1つを用いて2本の経路を1本にまとめることができるので、経路の総数をSとすれば、Sを1つにまとめるのに必要な履歴ビットの数Nは N >= ln[2](S) となる。
仮に履歴ビットが記録テープのような形状をしていたとすれば、各サイクルごとにNビットの履歴記録テープを書き進めることになるだろう。

4：もう少し詳細を検討すると、履歴記録テープの扱いには幾つかの仕組みが必要になってくる。
例えば
a.1つの履歴ビットに書き込んだ後、次の履歴ビットにコマを進める
b.S本の経路に対してNビットの履歴テープをあてがう。
などである。
これらの機能は全て固定的な仕組みで実現できるため詳細は割愛するが、概ね次の様になるだろう。
a. 履歴ビットへの書き込み装置（書き込みヘッド）は、書き込みの後無条件に次のコマに移動する。
この装置は「物体の移動」によって実現できて、その過程に経路の分岐は無い。
b. S本の経路に対してNビットの履歴テープをあてがう処理はやや複雑である。
最も簡単な方法は、あらかじめ全てのサイクルで発生し得るSの最大値Smaxを割り出し、毎回Nmax >= ln[2](Smax)をあてがうことである。
この方法であれば、固定的な仕組みによって実現できる。
ただし、この方法だとSmaxにならなかったサイクルでは無駄な履歴ビット（Nmax−N）を消費してしまうので、いささか効率が悪い。
それでも、とにかくこの「毎回最大ビット数をあてがう」方法で一応の機能はカバーしている。

5：履歴ビットを用いて経路が1つにまとまれば、次のサイクルを問題なく1つの開始点から（現在ヘッドのある位置から）スタートさせることができる。

以上の様な仕組みを用意すれば、たった1つのボールで可逆なチューリングマシンを実現することができる。チューリングマシンとは本来不可逆なので、可逆に動作させようとすれば何らかの仕組み、例えば履歴記録テープを付け加える必要がある。可逆チューリングマシンは、実在のコンピュータと比べて計算速度の面で猛烈に効率が悪い。それでも、可逆チューリングマシンはエネルギー消費の面では実在のコンピュータを上回るであろうし、何よりもたった1個のボールでコンピュータの全てが実現できることは興味深い。

■ ※
ここで紹介したチューリングマシンは、元来のチューリングが考案したものと同一ではなく、幾つかのアレンジが入っていることをお断りしておく。オリジナルのチューリングマシンでは、1，0だけではなく、複数の記号の読み書きを扱っていた。また、オリジナルでは記号が書いてある部分は有限長であり、そこから先は記号が書かれていない空白の状態としていた。本論で示したチューリングマシンは、記号が1，0だけで、空白状態を考えていない。（もともとテープは0で埋まっているもと考えている。）一見するとこれではコンピュータとしての動作がおぼつかない様に思えるかもしれない。例えば2進数で数字を書き込むことを考えると、データの一部を表す0なのか、空白状態の（最初からテープを埋めていた）0なのか区別がつかないのではないか。確かに、一般的な2進数のコンピュータを念頭に置けば、この疑問はもっともかもしれない。ところが、実のところ工夫次第で1と0しかないテープであってもチューリングマシンを構成することができるのである。例えば、

・2進数ではなく、単純に1の並びの長さによって1つの数字を表現する。
・1の間に挟まった0を、数字同士の区切り文字として用いる。
・0が2個以上続いていたらデータの終点を表す。

という方法で（2進数より効率は悪いが）あらゆる数字列を表現できる。「皇帝の新しい心（ペンローズ）」という本に1と0しかないチューリングマシンの例が載っているので、参照されたい。

■ ※
可逆チューリングマシンを構成する方法として、ベネットは逆動作チューリングマシンをペアにして置くことを考えた。

順方向チューリングマシン –> 結果のコピー –> 逆方向チューリングマシン

「ファインマン計算機科学」に載っていた。

クロックによるゴミ掃除

2006/08/23

前節では、たった1つのボールと履歴記録テープを用いて可逆チューリングマシンを構成する方法を考えた。可逆チューリングマシンが演算の後に残すのは、記録テープに残された演算の結果と履歴の列である。必要とされる演算結果以外の「データのゴミ」を残すのは、可逆コンピュータの特徴の1つだ。ビリヤードボール・コンピュータの場合、演算後にはどこにボールが入っているかわからない箱の列が残った。可逆チューリングマシンの場合には、何が記録されたかわからない履歴記録テープが残される。つまりこの2つは、途中の演算が複数のボールによって並列に進行するか、1個のボールによって直列に進行するかの違いがあるものの、最終的な生成物は本質的に同じなのである。可逆コンピュータであれば、演算の途中で生じた余分な（答として必要とされない）結果をデータの形で残す必要に迫られる。不可逆なコンピュータは、余分なデータが生じるたびに（各素子、各ステップごとに）それらを熱の形でコンピュータの外に捨て去っている。可逆コンピュータの演算後に残された「データのゴミ」は、不可逆なコンピュータが捨て去っていた熱と同じ意味合いを有しているのである。

可逆コンピュータで生じる「データのゴミ」を、不可逆コンピュータの様なエネルギー消費なしに上手く掃除する方法は無いものだろうか。「データのゴミ」を掃除するとは、何が記録されているか不明な状態を、明確な状態に戻す作業のことを指す。履歴記録テープで言えば、全ての履歴ビットを消去して0にする作業のことである。実のところ、前節で挙げた3種類の可逆な部品、

1：ボールと経路
2：物体の移動
3：条件分岐（合流）

だけを用いてデータのゴミ掃除を行うことはできない。（だたし、全ての演算を逆向きに行って元の状態に戻す、という掃除の方法は除く。）データのゴミ掃除を行うには、部品をもう一種類つけ加える必要がある。その4番目の部品とは「周期的に運動する物体」、即ち「クロック」である。

4：クロック（時計）
周期的に運動している物体にボールを衝突させることによって、ボールの経路を変える仕組み。ボールの経路は物体と衝突した時刻によって切り替わる。この仕組みは、例えば回転する反射板にボールをぶつけることによって実現できる。反射板にぶつかるタイミング（回転の位相）によって、ボールは相異なる特定の方向に反射する。単純な回転する板だと、衝突の際にボールと反射板との間にエネルギーのやりとりが生じてしまう。これを防ぐため、衝突の瞬間に反射板をしっかりと固定し、ボールの勢いで反射板の周期運動が攪乱されないような仕掛けを施す必要がある。反射板を固定する仕掛けは同時に、ボールの経路を連続的にではなく、離散的な固定した方向に向ける役割も果たす。固定する仕掛けの付いた反射板は、例えば板とストッパーとクロック用ボールの3つの部品によって構成できるであろう。

＊＊＊　これは図で説明しよう　＊＊＊

反射板はあたかも時計の秒針のように、すばやく移動、停止、を繰り返すことになる。現実の時計はゼンマイなどに蓄えられたエネルギーを少しずつ消費するが、ここでは摩擦の無い世界を考え、クロック自体にエネルギーの損失は無いものとする。

クロックを用いて異なる経路を1つにまとめる方法を考えてみよう。経路を1つにまとめるとは、経路を2つに分岐する過程の逆のことだ。経路を分岐する過程は、回転する反射板のイメージを思い描けば容易に想像が付く。反射板と衝突するタイミングによって、ボールは異なる方向に弾き返される。仮に、

時刻Aでぶつけたボールは経路Aへ、時刻Bでぶつけたボールは経路Bに向かうとしよう。この逆の過程を経れば、2つの経路AとBから来たボールを1つにまとめることができるはずだ。いま、何らかの原因によって、例えばメモリービットの読み取り等によって、ボールの経路が2つに分岐したとする。2つの経路をA、Bとしたとき、Bの経路をAの経路よりも長くとって、通過に余計な時間を要するように配置する。そうすれば、AとBから出てきたボールを異なるタイミングで反射板にぶつけることができる。経路の長さと反射板の位相を調整すれば、2つの経路から来たタイミングの異なるボールを1つの経路に導くことができるだろう。このクロックを用いた経路の合流とは、つまるところ経路の違いを時間の違いに置き換える仕組みである。経路の本数を減らせる代わりに、経路を通過するボールのタイミング、時間に対するパターン数は増大する。

次に、クロックを用いてNビットの履歴記録テープを掃除（0クリアー）する方法を考えてみる。もし、この仕組みを1ボールチューリングマシンの後つなげることができれば、データのゴミを残さない「クリーンな」マシンが実現できることになる。記録テープ掃除の基本的な考え方は、上に示した1ビット（2通り）の経路をまとめる方法の応用に過ぎない。ある記録ビットを読み込み、その値が0ならば何もせず、1ならば値を0に書き換える。その結果、ボールの経路は値が0の場合と1の場合の2通りになる。この2通りの経路を1本にまとめる処理は、1つのクロックを用い、ボールのタイミングをずらすことによって実現できる。これで長さ1ビットの記録テープの掃除はできたわけだが、もっと長いテープの場合はどうだろうか。長いテープを掃除する最も直接的な方法は、全パターン分の経路を用意することだ。長さNビットのテープに書き込める1か0のパターン数は全部で2＾Nである。ということは、1個のボールでNビットのテープを読み取った場合、ボールの経路は2＾Nに分岐する。ここで、1ビットで行った方法をそのまま応用し、2＾N通りのパターンそれぞれについてビットの書き換えと時間調整を行う。つまり下表を全て埋め尽くす作業を行うのである。

・パターン0のとき、何もしない、経路の通過にT秒かかる。
・パターン1のとき、1ビット目を0に書き換える、経路の通過にT＋1秒かかる。
・パターン2のとき、2ビット目を0に書き換える、経路の通過にT＋2秒かかる。
・パターン3のとき、1ビット目と2ビット目を0に書き換える、経路の通過にT＋3秒かかる。
・パターン4のとき、3ビット目を0に書き換える、経路の通過にT＋4秒かかる。
・・・
・パターン2＾Nのとき、全ビットを0に書き換える、経路の通過にT＋2＾N秒かかる。

この表では、パターン番号をビットの2進数に対応させてある。最後に、2＾Nに分岐した経路を1つにまとめるため、反射板の向きが2＾Nだけあるようなクロックを用意する。時計の秒針が60通り、64＝2＾6だから、時計の秒針程度の刻みだと6ビットに対応することもできない。長い記録テープを1まとめにするには、時計よりもずっと細かい刻み巾の反射板が必要となるだろう。なので、非常に高い精度を要求されることにはなるが、とにかく理屈の上では高精度のクロックさえあれば長い記録テープをただ1個のボールによって0クリアーすることができるのである。もちろんこのとき、Nビットのテープが0クリアーされる代わりに、ボールの時間に対するパターン数は2＾N通りに増大する。

1個のボールによるデータのゴミ掃除の本質は、上でほぼ出尽くしている。即ち「Nビットのデータのゴミ処理を行うには、ボールの時間に対するパターン数を2＾N通りに増やさなければならない。」以下はやや蛇足気味ではあるが、もう少しスマートなクロックの仕組みを紹介しよう。上ではただ1枚の反射板によって2＾Nの経路を1まとめにすることを考えた。今回は反射板の数を増やす代わりに、1つ1つの反射板を単純化することを考える。反射板はN個用意する。その代わり、個々の反射板は2通りの経路を1つにまとめる機能だけを持つことにしよう。Nビットの長さの記録テープがあったとする。最初の1ビット目を読み取った値によって、ボールの経路は2つに分岐する。1ビット目の値を0に書き戻した後（あるいは何も行わずに値を0のまま保持した後）、1つ目のクロック（反射板）によってボールのタイミングをずらす。このとき、ボールのずれの間隔を2＾（N−1）単位時間だけ開けておくのである。（「単位時間」とはクロックの時間分解能のこと。「秒」と読み替えてもよい。）2ビット目は、1ビット目と同様の操作を行

うのだが、ボールのずれの間隔を2^(N−2)単位時間とする。以下、iビット目ではずれの間隔を2^(N−i)単位時間とする。つまり、後のビットになるほどクロックの刻みが細かくなってゆく。この操作を最後のビットまで行えば、最後にボールが1つの経路から出てゆくときには2^N通りだけの時間に対するパターン数を持つことになる。ちょうど時計で短針、長針、秒針、といった具合に針を分けているのと同じように、この操作ではクロックごとに時間の「桁」を分けているのである。ただし時計の針は60ごとに1つ上の位(針)となるのだが、ここでのクロックは2単位ごとに1つ上のクロックに繰り上がっている。

空間上に配置されたパターンは、特定の操作を通じて時間軸上に配置されたパターンに変換することができる。1個のボールによるデータのゴミ掃除は、この事実をプリミティブなモデルによって確認したに過ぎない。可逆チューリングマシンとその後の処理によって、ただ1個のボールを使って演算を行い、記録テープをクリーンに保てることが確認できた。このとき、全ての処理を終える時間の長さは固定にはなり得ない。処理時間の長さにパターン数を持たせることによって初めて、データのゴミ掃除が可能となるのである。

■ ※
合流は任意の時間、位相でできるわけではない。例えば(A:1秒後)、(B:2秒後)の組があった場合、(A:2秒後)の動作は不定だ。反射板のようなモデルで考えると、ボールが経路を外れてあさっての方向に飛んでいってしまう。ただ、ボールが許されない時間に入ってこないように、反射板の位相に合わせて経路の入り口にフタをすることは可能である。

クロックは悪魔に成り得るか

2006/08/23

これまでに述べてきた可逆コンピュータの知見を元に、本節以下ではマックスウェルの悪魔が生き残る道を探ってみる。ランダウアーの原理を応用して、ベネットは悪魔の問題点を指摘した。ここでは1ボールチューリングマシンの考え方を応用して、悪魔の可能性を考えてみようと思う。残念ながら、悪魔の可能性を見出すまでの道程は決して短くはない。そこで、先に結論を対比的に示しておくと次の様になる。

・ランダウアーの原理
コンピューターでエネルギーを消費するのは、演算の過程ではなく、メモリーを消去する過程にある。
・ベネットの悪魔祓い
悪魔が実現不可能なのは、測定にエネルギーを要すからではなく、測定後の状態を元通りに戻せないからである。

・1ボールチューリングマシンの考え方
処理に要する時間の長さを可変にすることによって、不要な履歴データを消去することができる。
・悪魔の可能性
１サイクルに要する時間の長さを可変にすることによって、悪魔の破綻を防ぐことができる。

まずはメモリーと気体分子の類似性について確認しよう。コンピュータにおけるメモリービットと熱統計力学における気体分子は実質的に同じものである。というより、両者を同じものだと見なせば、これまでに述べたようなコンピュータの物理的限界を論じることができる。反対に、コンピュータの考え方を気体分子にあてはめれば、熱統計力学の要請を確認することができる。可逆コンピュータの演算において不可避的に生じるのは「どの箱に入っているかわからないボール」であった。「データのゴミ」、あるいは「不定なメモリービット」と呼んでも本質は変わらない。要は、値のわからない、未知の状態が演算後に必ず残ることになる。一方、熱運動する気体分子から仕事を取り出したとき、その後に残るのは「どこに行ったのかわからない気体分子」である。仕事を取り出し終えた後の気体分子が飛び回る部屋を細かい区画に分割すれば、「どの箱に入っているかわからない気体分子」が手元に残るであろう。両者を対比させると、コンピュータにおける演算処理が、気体分子において仕事を取り出すという操作に対応付けられる。なので、コンピュータの演算に要するエネルギーは、対応する気体分子から取り出した仕事に匹敵するであろうという推測が成り立つ。また、コンピュータ上でいかなる演算を施してもデータのゴミを消去することができないので、熱運動する気体分子から無尽蔵に仕事が取り出せないだろうという説明が成り立つわけだ。

さて次に、前節で示した「クロックを用いたデータのゴミ掃除」の方法を熱運動する気体分子に適用することを考えてみよう。扱う対象としては「どの箱に入っているかわからない、ただ一個の気体分子」を取り上げる。これまで大雑把にメモリーと呼んできたものには、実は2つのタイプがあった。1つ目は「ボールが箱に入っているか、空か」によって記録するタイプ。これはビリヤードボールコンピュータの所で登場した。2つ目は「物体がどの位置に置かれているか」によって記録するタイプ。これはチューリングマシンの所で登場した。確かにこの2つは異なるタイプのメモリーなのだが、両者に記録された情報は相互に変換することができる。1番目のタイプに記録されている情報を、ボールや箱といった道具立てを用いて2番目のタイプのメモリーに写すことができる。（ただしメモリーのタイプによって箱の数は変わってくる。例えば「4つの箱の中のどれか1つにボールが入っている」情報は、「2通りの置き方のできる2つの物体の位置」の上に移すことができる。）なので、どちらのタイプのメモリーを選んでも本質的な差異はない。ここで1番目の「ただ一個の気体分子」タイプのメモリーを選んだのは、こちらの方が物理的に単純で扱いやすいからである。

気体分子との類似性からすると、複数の箱の中からボールの入った箱が特定できれば、それは利用可能なエネルギーを得たのと同じ意味合いを持つことになる。「ボールの入った箱が特定できる」とは、より具体的には「ボールの入った箱を定められた特定の位置に置く」ことと等価である。例えばボールの入った箱を一番左端に置く、ということができれば、左端に置かれたボール＝気体分子がピストンを右側に押すことによって利用可能なエネルギーを取り出すことができるだろう。ボールの入った箱を見出す作業を念頭においてマックスウェルの悪魔を思い描けば、どのようなものが考えられるだろうか。例えば次のようなプロセスはどうだろうか。

STEP1: 全ての箱を一列に並べ、1個ずつ順番に調べる。
STEP2: 箱に気体分子が入っていたら、ピストンを配備し、利用可能なエネルギー取り出す。
STEP3: 箱が空だったら何もせず、次の箱を調べる。

このようなプロセスが実現可能かどうか、以下に検討してみよう。

箱が空かどうかをエネルギーの損失なしに調べることはできるだろうか。例えば箱を静かにゆっくりと潰してみて、手応えがあったら分子があると判断すれば良い。この方法はベネットの考察で用いられていた「準静的過程」と同じである。箱を潰すときに加えたエネルギーは、潰れた箱を元に戻す際に回収できる。別の方法として、箱の重さを計るという方法もあるだろう。箱を秤、つまりバネの上に静かに乗せてみて、秤が沈んだら分子があると判断する。秤は分子の熱運動のため上下動するかもしれないが、平均値をとれば有意な差が得られるはずだ。静かにゆっくりと操作を行うという点では、箱を潰してみる方法と重さを計る方法に本質的な差異はない。いずれにせよ、エネルギーの損失を限りなく小さく抑えつつ箱を調べることは十分可能だ。

次に、複数の箱を1個ずつ順送りすることはできるだろうか。最も簡単な方法は、箱を円環状に並べて回転させることだろう。もちろん最初に円環を回転させる際にはエネルギーが必要だが、もし回転に摩擦が無いのであれば、円環はその後いつまでも回転し続ける。箱を順送りにするためには単に円環を回すだけではなく、回転を任意のタイミングで開始、停止する必要がある。回転の制御には、さらにもう1つのボールと幾らかの仕掛けが必要となるが、決して不可能ではない。詳細は下図を参照してほしい。

* 円環の外縁に歯車のようなギザギザがあって、そのギザギザに対してボールが衝突してスタート、ストップを行う。
* ぶつかるボールの経路上に仕切を入れて、ボールの動きを封じれば、円環の停止が実現できる。

「ピストンを配備し、利用可能なエネルギー取り出す」過程は複雑ではあるが、とにかく1連の機械仕掛けのシーケンスによって実現できるであろう。

*** できればここも簡単な図で説明しよう ***

ここまでの仕組みについてであれば、特に原理的な問題点は見あたらなかった。しかし、このプロセスにおいて最大の問題点は次の点にある。「箱に気体分子が入っていた場合と、入っていなかった場合で異なる状態となった装置全体を、再びもとの状態に戻すことができるのか」STEP2:の次の状態と、STEP3:の次の状態を、共に同じSTEP1:の状態とすることができるのか、という問題である。他に何の痕跡も残さずに、2状態を1状態に持ち込むことが不可能であると、これまで幾度となく強調してきた。単純な演算の組み合わせのみによって、箱に気体分子が入っていた場合と、入っていなかった場合の2状態をもとの1状態に戻すことはできない。そこで登場するのが、1ボールチューリングマシンの節で一役買った「クロック」である。クロックといった仕組みを導入することによって、問題を解決することはできないだろうか。STEP2:とSTEP3:の後に、前節で用いたような回転する反射板を配置して、2本の経路を1本の経路に合わせることを考える。例えばクロックを用いて、

- 箱に気体分子が入っていなかった場合は1単位時間後（1秒後）に、

・気体分子が入っていた場合は2単位時間（2秒後）に、
同一の初期状態に戻るように装置を構成したとしよう。１個目の箱の処理を終え、2個目の箱の処理を開始するときに、クロックの状態には次の２つの場合がある。

・１単位時間後（1秒後）の状態

・2単位時間後（2秒後）の状態

この2つの状態間で、クロックの反射板の角度は当然異なっている。なので、同じクロックを再度使い回して装置の状態を1つに戻すことは、もはやできない。2個の箱を扱うには、

・1個目の箱に気体分子があるかないかの2通り

・2個目の箱に気体分子があるかないかの2通り

で、合わせて2x2＝４通りの場合の数に対応しなければならない。従って、これら全ての場合をカバーするには4単位時間まで計れるクロックが必要となる。前節の1ボールチューリングマシンの場合、クロック自体の精度を上げる（クロックの刻み角度を小さくする）、あるいはクロックの数を増やす、等の方法で長い履歴テープに対処していた。同じ考え方をここでのプロセスにあてはめると、クロックの精度をどこまでもあげ続けるか、クロックの数をどこまでも増やし続けるかしないと、いつかプロセスが破綻することになる。1ボールチューリングマシンで扱っていた履歴テープは、扱うデータの長さが有限だという前提があった。今考えているプロセスは、いつまでも動き続ける、無限の長さを要求している。結局のところ、クロックを導入したとしても、ここで考えたようなプロセスは実現できないのである。

時間をずらすというアイデアは一見良さそうに思えるのだが、なぜクロックを内蔵した装置はマックスウェルの悪魔に成り得ないのだろうか。いま仮に、クロックを内蔵したある装置から不定な時刻に利用可能なエネルギーが取り出せたとしよう。この装置は一見すると、作用する時刻の全く知り得ない、不定なエネルギーを発しているように見える。しかし、もし観測者が装置内のクロックを見たとすれば、時刻を知り得ないといった性質は失われてしまう。装置内のクロック自体が、エネルギーの作用する時刻を記録しているからだ。従って、装置から発されるエネルギーは真の意味での「時刻不確定」ではない。マックスウェルの悪魔を実現するためには、装置内に残った「クロック」という記録を消し去る必要がある。

悪魔の装置第二号

2006/08/23

マックスウェルの悪魔は、たとえクロックの様な仕組みを導入したとしても実現できない。時間を記録する仕組みを導入すれば、その記録が残ることによって悪魔は破綻するであろう。もちろん履歴テープの様な時間以外の記録を残すこともできない。つまるところ可逆な部品だけを用いて悪魔を作り出すことはできないのである。残る可能性は可逆部品ではなく非可逆な構成要素、つまりエネルギーを投じて記録を消し去ることである。経路の合流、例えばN本の経路を1つに合わせるには E＝KT＊lnN だけのエネルギーを必要とする。であれば、素直にこれだけのエネルギーを投じてみてはどうだろうか。ただし、このエネルギーを投じるタイミングを不定にすることによって、クロックが果たしていた役割を投じるエネルギーに担わせるのである。
クロックとは、着目している運動と、基準となる運動の時間的な関係のことである。1ボールチューリングマシンの場合、着目している運動とはボールであり、基準となる運動は反射板の回転であった。ここでは着目している運動に、投じるエネルギーをあてはめる。そして基準となる運動には世界そのものを、つまり系の外界をあてはめる。つまり、系が外界から熱を吸収した時点を基準として、系が外界にエネルギーを投じるタイミングを不定とするのである。エネルギーの出入りをクロックに見立てる方法、果たしてこのようなやり方で上手くゆくのだろうか。以下に検討してみよう。

エネルギーを投じることによって経路を合わせるとは、具体的にどのような状況を指すのだろうか。例えば前章で見たような「Y字型の管」はエネルギー消費を伴う経路の合流であろう。しかしY字型の管では合流点で何が起こっているかを詳細に調べることができない。合流の様子と投じたエネルギーの関係を考察するには、やはり気体分子のモデルに立ち返るのがよいだろう。いまここに圧縮された気体があったとしよう。この気体を押さえつけているピストンのストッパーを外すことによって、気体は膨張し、何らかの仕事を行う。ここでは何らかの物体を移動し、経路をふさぐ、あるいは経路を変更するというのが妥当であろう。これらの道具立てによって、次のような最も単純な合流モデルができる。

・最初に、1個のボールが経路Aから進入する。
・ボールはピストンのストッパーに衝突して、これを外す。
・ストッパーを外し終えたボールは経路Xに入る。
・気体が膨張し、物体を移動することによってボールの来た道＝経路Aをふさぐ。
・次にボールが経路Aから来たときには、移動済みの物体にさえぎられるので、前回と同じ経路は通らない。
・物体が移動した後では、経路Aに変わって経路Bからのボールが経路Xに入る。

このモデルでは気体の膨張を利用して、A→Xという経路をB→Xに非可逆的に切り替えているわけだ。注意すべきは、このモデルが100％確実に意図した動作を行わない点にある。気体の為す仕事が熱ゆらぎに比してさほど大きくない場合、熱ゆらぎによって気体の膨張が「押し戻される」ことがある。投じるエネルギーが小さくなるほど押し戻される確率は上がってゆく。気体が押し戻された場合、経路Bと経路Xが結びつくべきところで、経路Aと経路Xが結びつくことになる。

上の気体の膨張を利用した経路の合流と、前節の「箱を1個ずつ調べるプロセス」を組み合わせることによって、悪魔の装置を構成することができる。

STEP1: 全ての箱を一列に並べ、運動するボールを用いて箱の中身を1個ずつ順番に調べる。
STEP2: 箱に気体分子が入っていたら、ピストンを配備し、利用可能なエネルギー取り出す。その後、ボールは経路Aに入る。
STEP3: 箱が空だったら、ボールは経路Bに入る。
STEP4: 上に述べた合流モデルを用いて、2つの経路AとBを同一の経路Xに合わせる。ここで合流のためのエネルギーが必要なのだが、それにはSTEP2:で得られたエネルギーをあてがう。

STEP5: 経路XからSTEP1:に戻る。

一見するとSTEP2:で得られたエネルギーはそのままSTEP4:で消費され、後には何も残らないように思われるかもしれない。しかし、経路合流の仕組みを思い起こせば分かるのだが、エネルギーはSTEP4:で消えてしまうわけではないのである。気体はボールの運動に関連付いたあるタイミングで膨張しているだけであって、ボールに対して何らかの仕事を為しているわけではない。それゆえ経路の合流に用いたエネルギーはそっくりそのまま回収できるか、あるいは他の有意な目的に振り向けることができる。装置全体を見直せば、STEP2:で用いるエネルギーを取り出すためのピストンと、STEP4:で用いる経路合流のために膨張する気体は、実は同一の仕組みで兼用できることに気付くであろう。つまり STEP2とSTEP4: は、ただ1つのピストンによって同時に行うことができる。気体を膨張させて仕事を取り出すと同時に、膨張のきっかけとなったボールの運動経路を制御する、単純に「ピストンが動く」というモデルを想定すれば、2つのSTEPが1つの動作によって実現できることが分かるだろう。

このようにピストンとボールの運動を互いに連動させると、装置自身がどうしても避けられない性質を帯びることになる。それは、ピストンが動作するタイミングがボールの運動に依存すること、つまりエネルギーを取り出すタイミングが不定となることである。ここで改めて得られたエネルギーは何処からもたらされたのかを考えてみると、取り出すタイミングに重要な意味があることに気付く。取り出すタイミングが不定とは即ち、プロセス全体が「クロック」の性質を有しているということだ。利用可能なエネルギーは、プロセスが不確定な周期で実行されていることによってもたらされているのである。いま、N個の箱の中のどれか1つに気体分子が入っている状況を考えてみよう。このN通りの状況に対応して、プロセスの1周期に要する時間はN通りだけ存在する。例えばボールが経路Aを通った場合に2秒、経路Bを通った場合に1秒かかるのだとすれば、エネルギーを取り出すまでに要する時間は最短で2秒、最長でN＋1秒のN通りとなる。プロセスの1周期に要する時間を変化させることによって、装置内部に溜まるはずであったデータのゴミを、取り出されるエネルギーと同時に装置の外に運び出すことができるのである。

さて、上では簡単にプロセスの流れを追ってみたのだが、より詳細に検証すると実は複雑な要素をはらんでいることが明らかになる。

まず、このプロセスは常にSTEP1:からSTEP5:に向けて順方向に動作するわけではない。熱ゆらぎの影響を受けて、経路の合流が意図通りには運ばないことがある。その結果、各STEPにおいて逆方向にプロセスが進行することが起こり得る。また、STEP2:で「箱の壁を外してピストンを配備し、利用可能なエネルギー取り出す」といった操作を行うためには、どうしてもSTEP3:の経路においても「箱の壁を外す」といった操作を施す必要が生じる。そうしなければ、経路Aと経路Bでその後の箱の壁の状態が異なったものになってしまうからである。これらの要素を考慮して完全なモデルの構成を試みると、かなり複雑なものとなる。（この複雑なモデルの詳細は次節に述べよう）

実のところこの複雑なモデルを煎じ詰めると、最後には第2章で紹介した「悪魔の装置第一号」に近づいてゆく。ここで考察したモデルでは、箱を調べて分子の有無によって2つの経路A，Bに分岐を行っていた。この分岐を、2つの経路ではなく、1つの経路をそのまま進むか、反射して戻ってくるか、に置き換える。つまり単純なスイッチ、経路の（継続，切断）に置き換える。同様に、経路A、Bの合流する箇所も単純なスイッチ、経路の（継続，切断）に置き換える。こうして出来上がったものは、第2章の「悪魔の装置第一号」と同じである。第一号も第二号も、本質的には同じものだったのである。装置の詳細な仕組みの検証に追われていると、ともするとその本質が見えにくくなってしまう。これら悪魔の装置の本質とは「クロック」の性質、つまり装置の動作周期が不定な長さを持つということである。

悪魔の装置第二号の詳細

2006/08/23

前節で挙げた”悪魔の装置第二号”の詳細を検討しよう。ここで述べる内容は、第2章「悪魔の装置第一号の詳細」にかなり類似している。

■ 構成要素

第一号で定義した「信号」「経路」「ゲート」「ハンドル」は、ここでも用いられる。「ゲート」とは、外界の物理的条件によって信号の流れを制御する装置のこと。「ハンドル」とは、信号の流れによって外界の物理的対象を操作する装置のこと。いずれも本論だけの特別な呼び方である。詳しくは第2章「悪魔の装置第一号の詳細、構成要素」を参照のこと。これらの構成要素を応用して、ここではさらにもう1つの要素をつけ加えよう。

＊：分岐ゲート
特定の物理的条件に従って、信号の経路を切り替える装置のこと。例えばある分岐ゲートは1つの経路Xの先を、物理的条件がaの場合には経路Aと接続し、物理的条件がa以外の場合には経路Bと接続する。分岐ゲートを逆向きに用いることによって、複数の経路を1つに合わせることができる。例えば、物理的条件がaのとき経路Aから来た信号と、物理的条件がa以外の場合に経路Bから来た信号は、分岐ゲートによって同一の経路Xに導くことができる。これ以外の場合、例えば物理的条件がaのとき経路Bから来た信号は、分岐ゲートで反射してもときた経路Bを戻るものとする。

■ 装置の構成

装置の挙動を追って、必要とされる構成要素をあてがってゆこう。

まず、複数の箱の中のどれか1つ気体分子が入っている状況を想定する。1つの箱を取り出し、その中に分子が入っていれば経路Aへ、入っていなければ経路Bへと信号を導くような分岐ゲートを置く。この分岐ゲートを、以下「プローブゲート」と呼ぼう。経路Aに入った信号は、続いて以下2つの動作を行う。

・箱を全てつなげて1つにする。
・ピストンを配置して気体分子から仕事を取り出す。

これらを実行するために、2つのハンドルが必要となる。それぞれ「箱接続ハンドル」と「ピストン移動ハンドル」とする。経路Aを通った信号と、経路Bを通った信号はどこかで合流する必要がある。この合流を行う分岐ゲート（分岐ゲートを逆向きに配置したもの）を「マージゲート」と呼ぶことにする。マージゲートによって2つの経路AとBは1つの経路Xへと導かれる。マージゲートを制御する物理的条件は、ピストンの位置に依るのが妥当であろう。ピストンが圧縮された状態にごく近い場合、マージゲートは経路Aと経路Xを接続し、それ以外の場合は経路Bと経路Xを接続する。

こうすれば、経路Aを通った信号がピストンを圧縮位置に移動した直後にだけ、経路Xに入る逃げ道が開くことになる。経路A軽油で経路Xに入った信号は、その後

・ピストンが仕事を取り出し終えるのを待つ。
・1つにつながった箱を元の複数の箱に戻す。

といった2つの動作を行う。仕事を取り出し終えるのを待つために、経路Xの先には1つのゲートを置く必要がある。このゲートを「リカバーゲート」と呼ぶことにする。リカバーゲートは、仕事を取り出し終えたピストンが膨張位置に達したときだけ開くように配置する。リカバーゲートの直後に、1つにつながった箱を元の複数の箱に戻すための処理、「箱分断ハンドル」を配置する。「箱分断ハンドル」は、先の「箱接続ハンドル」のちょうど逆である。箱接続ハンドルを逆の向きに信号が通過すれば、それは箱分断ハンドルとして機能する。次に経路Bのことを考えると、経路Bの上にも「箱接続ハンドル2」と「ピストン移動ハンドル2」、「リカバーゲート2」が必要なことに気付く。経路Bを通過したときも、経路Aと同様に箱を

接続しておかないと、後で経路Xから出るときに箱を分断することができない。ピストン移動ハンドル2も同様の理由による。経路A上に置かれたピストン移動ハンドルがピストンを圧縮位置に置くのに対し、経路B上のピストン移動ハンドル2はピストンを膨張位置に置く。信号が経路Bの上を逆向きに走ったときでもピストンが必ず膨張位置にあることを保証するため、経路B上にはリカバーゲート2が必要となる。リカバーゲート2は、ピストンが膨張位置にあったときだけ開く。経路Xからリカバーゲートを抜けた信号は、再び最初のプローブゲートの入口に戻って、箱の中身の調査を開始する。当初の「箱を1個ずつ順番に調べる」というアイデアからすると、「箱を1個だけ順送りして次に移動する」という処理が必要と思われるかもしれない。しかし、この順送りは意味を成さない不要な処理である。なぜなら、箱をつなげて1つにした時点で分子はどこに飛んでゆくのか分からなくなるので、次に調べる範囲（箱）はどこでも同じだからである。

＊＊＊以上の様子を図示してみよう＊＊＊

■ 装置の挙動

以上で考えた経路上で、信号が順方向、逆方向に回る確率をそれぞれ見積もってみたい。もし順、逆の確率が全く同等であれば、全体としてこの装置は意味を成さないことになる。もし順、逆の確率に差異があれば、この装置は分子の熱運動から有意なエネルギーを取り出せることになる。ここでは経路上の2つの状態間の遷移確率を調べることにしよう。信号が箱の中の分子を調べる直前にあるときを「状態0」とする。信号が「マージゲート」と「リカバーゲート」の間にあって、気体分子がピストンを押している（あるいはピストンに押されている）状態を「状態X」とする。この2つの状態、0とXを結ぶ経路は3本ある。箱の中に分子が見いだされ、信号が経路Aを通ってピストンを移動した後に状態Xとなる経路「ルートA」。箱の中に分子が見いだされず、信号か経路Bを通って何もせずに状態Xとなる経路「ルートB」。信号が箱を調べるのとは逆の方向に走り、リカバーゲートから状態Xとなる「ルートR」。この3本のルートについて、それぞれ順方向（状態0 -> X）と、逆方向（状態X -> O）の確率を調べてみよう。

状態O -> X:

順方向と逆方向を同じ重みと考え、Pr（ルートA）＋Pr（ルートB）＝ Pr（ルートR）とする。
部屋の数はN個あるものとして、分子を発見する確率は 1/N 。

ルートA
　　1/2 ＊ 1/N
ルートB　　1/2 ＊ (N-1)/N
ルートR　　1/2

状態X -> O:

ルートA~
　　押し戻される確率 ＊ 箱の中に分子が見つかる確率:
　　　　S ＊ 1/N ＊ exp(− ΔW/kT) ＊ 1/N
ルートB~
　　伸びきる確率 ＊ 箱の中に分子が見つからない確率 :
　　　　S ＊ 1/N ＊ exp(ΔW/kT) ＊ (N-1)/N
ルートR~
　　伸びきる確率 :
　　　　S ＊ 1/N ＊ exp(ΔW/kT)

ここで S は Pr（ルートA）＋ Pr（ルートB）+ Pr（ルートR）の値を1に規格化する定数で、

S = N/2 ＊ { (exp(− ΔW/kT) ＊ 1/N) + (exp(ΔW/kT) ＊ (N−1)/N) + exp(ΔW/kT) }

= N/2 * { (exp(− ΔW/kT) * 1/N) + (exp(ΔW/kT) * (2N−1)/N) }

といった値をとる。

ピストンが押し戻される確率と伸びきる確率は次の様にして求めた。

まずエネルギーの差異が、系の状態にどのように寄与するかを考える。
いま、2つの異なる状態間にΔWだけのエネルギー順位の差があったとする。
系が2つの状態それぞれにある確率を P1, P2 とすれば
ΔW = kT ln[P2 / P1]
この式を、分子を圧縮領域に押し込むには ΔW だけの外力が必要だと読み替え、P1 を系が圧縮領域にある確率、P2 を圧縮領域外にある確率と解釈する。
次に、部屋の広さの寄与を考える。
空間全体に対して圧縮領域は1/Nの広さを占めているので、広さの違いとエネルギーの偏りの両方を掛け合わせて
押し戻される確率 Pcompress = 1/N * exp(− ΔW/kT)
膨張領域に伸びきっている確率は、圧縮領域の逆になる。
ただし、ここでは膨張領域も全体の1/Nとしてある。
伸びきる確率 Pexpand = 1/N * exp(ΔW/kT)
式の形としては expの中の符号が変わるだけである。

仕事を100%準静的に取り出した場合、力がつりあっているのでどこが有利ということもない。このとき

押し戻される確率 Pcompress = 伸びきる確率 Pexpand = 1/N

となる。これは上の式でΔW = 0の場合に相当する。
ΔW = 0として状態X -> O:を見直すと次の様になる。

状態X -> O:

ルートA~
押し戻される確率 * 箱の中に分子が見つかる確率:
S * 1/N * 1/N
= 1/2 * 1/N
ルートB~
伸びきる確率 * 箱の中に分子が見つからない確率 :
S * 1/N * (N−1)/N
= 1/2 * (N−1)/N
ルートR~
伸びきる確率 :
S * 1/N
= 1/2

この場合は S = N/2 であり、上の状態0 -> X:と全く同じ値を取る。つまりΔW = 0のときは状態X -> O:と状態0 -> X:は完全に等しくなり、順、逆方向のどちらかが一方的に優先されることはない。
ΔW ＞ 0のときは、ΔW = 0 のときに比べて Pr(ルートA~)が小さくなり、Pr(ルートB~)と Pr(ルートR~)が大きくなる。つまりΔW ＞ 0のとき、信号がルートAから入ってルートB、又はルートRから抜ける確率が大きくなる。信号がより高い確率でルートAに入るということは、分子がピストンを押して熱運動を仕事に変える確率が高い、ということである。

以上は式を追わずとも、ある程度まで推し量ることができるだろう。仮に、この装置が常に釣り合いの状態を維持し、熱運動＝ピストンにかかる外力であったとしよう。このとき装置は常に順方向と逆方向が釣り合っているのだから、どちらか一方向だけが優位にはならない。次に、ピストンにかかる外力を少なくして、気体が部屋いっぱいに広がる向きの方が優勢だったとする。すると、気体を圧縮状態に戻す確率、即ちルートAの逆は起こりにくく、気体を膨張する確率、即ちルートRの逆は起こりやすくなる。つまり、気体が膨張するのが自然な傾向だとするならば、その分だけ熱を仕事に変換できることになるのである。

■ 秘密はどこにあるか

装置の詳細を追ってみると、まるで手品の様に熱運動が有効なエネルギーに変換されたように感じられるかもしれない。しかし、この装置の最も重要な原理は複雑なゲート構成や確率の評価に潜んでいるのではない。次にエネルギーが取り出されるまでの時間の長さが不定なこと、これが最大の鍵である。
可逆コンピュータから考察を進め、コンピュータと類似の装置にはデータのゴミを捨て去る仕組みが必要なことが分かった。しかし、ここで考えた装置には履歴テープもクロックも無い。装置から取り出されるエネルギー自身がデータのゴミを捨て去る働きを担っているのである。装置内で「リカバーゲート」と呼んだ部分、2つの経路A、Bの合流点がデータのゴミを破棄している部分に相当する。リカバーゲートでは、気体分子がピストンを押す力＝装置から取り出されるエネルギーを用いて経路の合流を実現している。なぜエネルギーを稼ぎ出し、なおかつデータのゴミを捨て去ることが可能なのか。それは装置の1周期の長さ、一度エネルギーを取り出してから次回にエネルギーを取り出すまでの時間の間隔が不定なためである。仮に、装置の周期に要した時間を何かに記録しておいたとすれば、その時間についての記録こそが履歴テープと同じ意味を持つ。装置から得られたエネルギーと、履歴テープ作成に要するエネルギー、つまり時間の長さを記録するのに要するエネルギーは等しくなる。ところが、この装置では内部には一切記録を残さない。ただ、取り入れた熱運動と、取り出されたエネルギーの関係にのみ記録は残されている。つまり「エネルギーの取り出された時刻＝履歴の記録そのもの」なのだ。それゆえ装置自身は破綻せずに、熱運動を有効なエネルギーに変換することが可能となるのである。

第三章 可逆コンピュータからの発想
未来予知と悪魔の関係

2006/08/23

<u>「エネルギーを取り出す時刻を不確定とすれば、マックスウェルの悪魔は実現可能」</u>
これが本論の主張である。なぜ時刻が不確定であればエネルギーを取り出すことができるのか。その意味がいまひとつつかみ取れない、あるいはどことなく話が上手すぎるように思える、そんな方々も多いことと思う。ここでは時刻不確定の意味を明らかにするため、コンピュータによる未来予知についての話題を取り上げる。時刻不確定なマックスウェルの悪魔がたとえ存在したとしても、コンピュータによる完全な未来予知を行うことはできない。このことは悪魔がエントロピー増大則を犯していないことと同義であり、それゆえに悪魔の存在が許されているのである。

今日、コンピュータは様々な局面で未来予測に役立っている。例えば天気予報には大がかりなシミュレーションを行うスーパーコンピュータが導入されている。コンピュータが1つの答をはじき出したとき、それまで全く予測のつかなかった未来に一縷の手掛かりを与える。もちろん天気予報がしばし外れるように、コンピュータの答とて間違うこともある。しかしそれは、コンピュータ予測という方法自体が間違っていたというより、データが不完全である場合が多いであろう。もし金と労力に糸目を付けず超高性能なコンピュータを作成し、徹底的にプログラムのバグを排除し、細大漏らさず完全なデータを与えたとしたら、コンピュータの答は神託の重みを持った予言となるのではないか。残念ながら今日の見解では、いかに優れたコンピュータを作ろうとも完全な未来予知は不可能だと言われている。その理由は主に2つある。

1：非常に小さな世界では、物体の位置と運動量を同時に正確な数字で表すことができない（不確定性原理という）。
2：未来を知る為には現在の世界についての情報の全てを細大もらさずinputしなければならず、それは原理的に不可能。
（第1章13節、ラプラスの悪魔を参照）

ここでは理由1：には深入りせず、理由2：について考えることにする。

未来のできごとについて、コンピュータが1つの答を導き出すためには必ずある程度のエネルギー消費をともなう。ただしこのエネルギー消費は、コンピュータの演算だけではなくデータ入手の過程、つまり測定の過程まで含んでの話である。可逆コンピュータの節で見たように、純粋な演算だけであればエネルギー消費は限りなくゼロに近づけることができる。しかし、対象の測定までをも含む全過程においては、エネルギー消費を避けることができない。実際にエネルギー消費の過程がどこで生じるかは、具体的なコンピュータの設計に依存する。測定の段階か、個々の演算の段階か、最後のメモリー消去の段階なのか、いずれにせよどこかの段階で必ずエネルギーを消費することになる。もしコンピュータがYes/Noの2択に対して答を導き出したなら、そこに要する必要最小限のエネルギーは1bit相当＝kT＊ln(2)となる。仮に求める答が 3.1415926 だったとして、これを 0.0000000から 99.9999999 の範囲から選択するものであったなら、kT＊ln(1000000000)だけのエネルギー消費が必要とされる。このようにコンピュータによって得られた情報とは、エネルギーと引き換えにもたらされたものなのである。ところで、このkT＊ln(N)というエネルギーを熱・統計力学の観点から解釈するとどうなるか。これだけの大きさのエネルギーを理想気体が熱として受け取った場合、気体分子の持つ「場合の数」はN通りだけ増大する。コンピュータがN通りの未来の中の1つを選び取るには、その代償として他の何物か、例えば気体分子などの場合の数がN通りだけ増大するのである。ここにシンプルで美しい法則がある。（コンピュータ＋気体分子）全体の場合の数は、常にN通り以上となる。このN通りという数は決して減らすことができないので、どれほどコンピュータを駆使しようとも世界全体の未来予知はできないのである。「場合の数は決して減らすことができない」これがエントロピー増大則と呼ばれるものの本質である。

ところで、私はこれまでに「時刻を不確定とすれば、マックスウェルの悪魔は実現可能」という話をしてきた。この悪魔の方法を用いれば、コンピュータのエネルギー消費を限りなくゼロ近くに抑えつつ、答を導き出すことができるのだ。コンピュータから答が出る時刻をずらせば、一見不可能に見えたエネルギーゼロの要求を満たすことができる。例えば、もしコンピュータにYes/Noの2択の問題を解かせるならば、Yesの答が出るのを1秒後に、Noの答が出るのを2秒後に仕掛けておく。そのような「答によって処理時間の長さが異なるプログラム」を、あらかじめコンピュータに組み込んでおくのである。我々がコンピュータの答を得たとき、その答によって未来の持つ可能性はYes/Noの2通りから1通りに絞られる。しかしそれと同時に、答自体がはじき出された時刻が異なるため、未来の持つ可能性は2通りに広がる。もし我々がコンピュータの答に従って何らかの行動を起こすのだとすれば、1秒後に行動を起こす場合と、2秒後に行動を起こす場合では異なる未来が待っているはずだ。結局のところ（Yes：1秒後）と（No：2秒後）の2通りという場合の数は、全過程において変わってはいない。計算時間が予想不能なコンピュータであれば、その計算時間によって未来が変わってしまうので、未来の持つ可能性の幅を減らすことができないのである。それゆえに、計算時間が予想不能なコンピュータはエネルギーの消費なしに作動することが物理的に許される。計算時間が確定しているコンピュータであれば、こうはならない。もし計算時間が決まっているコンピュータが未来についての答を次々とはじき出していったなら、未来の持つ可能性の幅はどんどん減少することになる。そのようなコンピュータが完全な未来予知を行わないように、物理法則はエネルギーという制約を課したのである。つまり、神は未来予知の能力をコンピュータから奪っておいたのだ。逆に言えば、どれほど演算を行っても未来予知に到達しないような「不完全な」コンピュータであれば、その存在は神から許される。計算時間が予測できない「いいかげんな」コンピュータとは、そういった「許された存在」なのである。

それでは、もし我々が正確な時計を持ち、答が出た時刻が何時何分であったか正確に記録しながら、時刻不確定なコンピュータを使い続けたらどうなるだろうか。つまり、未来に関する答はコンピュータから得つつ、時刻に関する情報の不備は時計によって補えば、完全な未来予知ができるのではないか。残念ながらそうはならない。なぜなら時刻を測定する過程、つまり時計にエネルギーを消費するからである。「時刻不確定なコンピュータ＋時計」は、「計算時間が確定しているコンピュータ」と同じ意味を持つ。どこかにエネルギーの消費が発生するので、完全な未来予知には至らない。つまり、時計によって得られるはずの情報を捨てることによって、その分だけのエネルギーが稼ぎ出せるのである。類似のアイデアで、答が確定するまで待ってはどうかという疑問もあるだろう。上の例では3秒後まで待ってから次の行動を開始すれば、時刻不確定といった性質はなくなるのではないか。ところが、この場合もエネルギー消費は避けられない。1秒後に出た答を2秒待つのと、2秒後に出た答を1秒待つのとは、相異なる2つの行動である。この2つの行動を1つに「合流」しようとすれば、どうしてもそこに1bit相当のエネルギーの消費が避けられない。

ラプラスの悪魔は存在することが許されないが、マックスウェルの悪魔は存在することが許される。（はじめに、悪魔兄弟の違い参照）マックスウェルの悪魔は、時刻不確定の贖罪を以て熱力学の神から許されているのである。

CoffeeBreak 〜 未来予知について

2007/03/01

未来予知というのは興味深いトピックなので、ここに少々の蛇足を付け足そう。
この節は本論に直接関係のない脱線なので、読み飛ばしてもかまわない。

古典力学の頃ならいざ知らず、今日では計算によって未来が完全に予言できると信じている人はあまりいないであろう。そう聞いて何となく安心感を覚える方も多いのではないだろうか。現代社会とは、全てがスケジュール通りに動き、またそうなる様にあらゆる努力が為されている所である。しかし、その中を生きる一人の人間としては、やはり未来は決定されていない方がいい。10年後、20年後の未来が計算でピタリと当てられてしまっては、夢も可能性もないではないか。しかし、未来予知が不可能ということは、ある意味で科学の敗北であるとも言える。（実用に役立つ）科学というものは、未知なる結果を予言するために存在するのだから。以下に挙げる「未来予知ができない理由」は、そのまま”科学の限界点”を示している。むろん、限界があるからといって、即「科学は無用の長物」だとはならない。何事にも適応範囲というものがある。科学とて万能ではない。限界を知ることによって、逆に力量を正しく評価できるのではないだろうか。

＊ 未来予知ができない理由

■ １： この世のあらゆる情報を集めることは不可能

これまで本論で中心的に語ってきたことである。情報を得るには、その対価としてエネルギーの散逸がともなう。情報と利用可能なエネルギーは等しい価値を持つ。それ故に、第二種永久機関は実現できない。

■ ２： 不確定性原理による決定論的な記述の限界

ある種の物理量は、複数同時に、正確に測定することはできない。物体の位置と運動量は、同時に正確には定まらない。同様に、時間とエネルギーも同時に正確には定まらない。確かに不確定性原理にはパラドキシカルな一面がある。位置や運動量などの物理量は、「本当に実在として」正確な値を有していないのだろうか。それとも単に観測者が知り得ないだけで、「本当は」正確な、決定的な値があるのだろうか。観測値の解釈は、現代においてもしばし頭を悩ませる。ただ、少なくとも現在の我々は、正確な値を得るための物理的な手段を持ち合わせていない。それ故、我々にとって正確な観測値は存在せず、観測値に基づく正確な未来の予言も存在しない。

■ ３： 個々に単純なルールでも、組み合わせることによって思わぬ複雑さを生み出す

現象を記述する法則がわかったからといって、現象そのものが完全に解けたことにはならない。囲碁のルールを覚えるのは簡単だが、誰もが囲碁の名人になれるわけではない。物理におけるルール、法則は、主に微分方程式で記述される。微分方程式で記述できれば、それは即ちルールを記述したのと同じことだ。いったん方程式ができあがってしまえば、後はルールに従って操作を施すだけで良い。つまり微分方程式を解けば良い。ところが、実際には解けない微分方程式がある。と言うよりむしろ解ける方程式が特別なのであって、現実には解けないものの方が圧倒的に多い。記述されたルールは単純かつ明確であっても、その結果は複雑で予想が付かない。こういった現象を、しばし「決定論的カオス」と言う。自然は個々の要素の単純な（線形な）重ね合わせではない。

■ ４： 予言者自身を含んだ世界の未来は１つに収束しない

仮に、未来を予言できるスーパーコンピューターがあったとしよう。あなたはこのコンピューターに向かって「明日の自分自身の行動」を予言してもらったとしよう。「あなたは明日８:００に起床して、朝食にトーストを紅茶を食べ、いつものように通勤電車に乗って、会社では制作中のプログラムの続きを・・・」という結果が得られたとする。この予言結果を見て、あなたはどう思うだろうか。ここで少々あまのじゃくな精神を持った人ならば「機械の言うなりになってたまるか。確かに明日はトーストにしようと考えていたが、ここは変更しておにぎりにしよう」などと考えるのではなかろうか。そう考えたとたん、カタカタとコンピューターが動き出し「訂正：明日はおにぎりにしようと考えている」という答が吐き出される。あなたは内心ぎくりとしながらも「やっぱり明日はコーンフレークだ！」と考え直すだろう。すると、再びコンピューターが動き出し「訂正：やっぱり明日はコーンフレーク」と表示される。ここであなたは怒って机をたたき「いったい何が本当の答なんだ、もう訂正は効かないぞ、最終的な回答を出してみろ！」とコンピューターに要求する。すると・・・コンピューターは沈黙するはずだ！（壊れるかもしれない）
この話を聞いて「やはり人間様は計算機を超えるすごい存在なんだ」と感じたであろうか。私はそうは思わない。答が出せない理由は「人間の自由意志」ではないからだ。常に命令に背くという行為は、常に命令に従うのと同じくらい決定的な行動だ。人間でなくとも、常に命令と反対の動作を行う「あまのじゃくロボット」で事足りる。人間の意志でないとすれば、一体何が予言を妨げているのだろうか。それは「コンピューター自身の出した答えが、未来に影響を与えてしまう」からだ。コンピューターが人間（あるいはあまのじゃくロボット）に未来を語った瞬間に、未来は変わってしまう。どれほどコンピューターが優れていようとも、自分自身の出した影響を再び自分自身に組み込んで、1つの確定した答を出すことはできない。フィードバックを組み込んだ制御系が発散することがあるのと同じ理由で、予言者自身を組み込んだ世界の未来は1つに収束しない。
ひょっとすると沈黙したコンピューターは、内心「あなたは私が黙ったのをよいことに、通常通りトーストを食べる」などと考えるかもしれない。だが、残念なことにその答を世界に出すことができないのである。出せば、その答の影響を受けて未来が変化してしまう。答の出せない予言、というのはもはや予言としての意味を為さない。
あるクレタ人が「クレタ人は嘘つきである」と言ったなら、この言葉は果たして嘘だろうか、本当だろうか。このお話は、予言者が避けて通ることのできないパラドックスだと思う。

■ ５： この世界こそが最速の計算機である

たとえば、投げたボールがどのように飛んでゆくか正確に知りたかったとしよう。最もシンプルな方法は、ボールを質点と見なして、初速と角度から飛跡を計算することだ。これは簡単な方法だが、空気抵抗の影響を無視している。次に、空気抵抗を考慮してそのような項を付け足そう。答は前よりは幾分正確になるが、これではボールにカーブがかかっていることまではわからない。ボールの大きさ、回転、表面のでこぼこ等も重要だ。これらを含めると相当大掛かりな計算となるが、より正確な答が得られるだろう。これで厳密に正確な答かというと、さにあらず。地球の自転が影響しているかもしれない。風が吹いているかもしれない。究極的には、ボールの通る道筋の、空気分子1個1個の衝突が影響を与えているはずだ。ここまで来ると、計算機によって答を得ようとする努力をあきらめざるを得ない。計算するよりも、実際にボールを投げてみた方がはるかに早く答に達することだろう。
ここから先は予想に過ぎないのだが、もし現実世界を寸分違わずシミュレートできる計算機があったとしても、その計算機が答を出す速度は現実世界を上回ることはできないのではないか。この考えに根拠は無い。あるとすれば「きっと自然というものは無駄なことははしていないだろう」という思い込みだけである。計算結果というものは正確であるに越したことはない。が、いかにわずかの手間で、早く答を得るかということもまた重要だ。もし計算の方が実験より膨大な手間がかかると言うのであれば、計算という操作は無意味だ。考えてみれば、ボールを1つの点と見なすなどというのは「現実」を無視した大胆な切り捨てであろう。だからといって「点という概念は間違っている」とか「不正確で話にならない」ということにはならない。ここまで思い切って簡略化したおかげで、最もわずかの手間と時間で答に達することが可能となったのだから。これほど切り捨てたにも関わらず割と正確な答が導き出せるということは、「質点の概念」が優れた思考方法だという証だ。

そもそも科学の本質は「思考の経済」にある。事実を簡略化した分だけ頭にかかる負担が軽くなり、それだけ遠くに飛べる様になる。それでは、少しも簡略化することなしに、ありのままの現実をそっくりそのまま受け入れることはできないのか。これは科学の限界、というより人間の認知能力の限界であろう。私たちは物を見て、考えはじめた瞬間に、現実の一部を切り取っている。余計な物を切り捨てているからこそ本質が浮き彫りにされるのだ。残念なことに、この切り取り操作がある限り、この世の全てをカバーした真理には到達できない。これが科学の限界であり、人の限界だとも思うのである。

まとめ 〜 可逆コンピュータに宿った悪魔

2006/08/23

コンピュータで演算を行う為には、最低限どれだけのエネルギーを必要とするか？

この問いかけから出発して、本章では可逆コンピュータからマックスウェルの悪魔までの道程をたどってきた。これまでの道程を振り返ってみよう。

1：可逆コンピュータ
コンピュータの演算自体に要するエネルギーは、原理的には限りなくゼロに近づけることができる。

2：ランダウアーの原理
不可逆なコンピュータがエネルギーを消費するのは、演算で生じた「データのゴミ」を消去する過程においてである。

3：ベネットの悪魔祓い
マックスウェルの悪魔が実現不可能なのは、分子操作の後に残された「データのゴミ」を、悪魔が忘れ去る手段を持たないからだ。

4：クロックによるゴミ掃除
ただ1個のボールによって動作する可逆コンピュータ「1ボールチューリングマシン」を考えたとき、周期的に運動する物体「クロック」を用いればデータのゴミ掃除ができる。

5：クロックは悪魔に成り得るか
「クロック」をそのままマックスウェルの悪魔に応用したとしても、悪魔の体内に時間に関する記録が残るので破綻をきたす。

6：悪魔の装置第二号
系から取り出されるエネルギーに「クロック」の性質を担わせることによって、つまりエネルギーの取り出される時刻を不確定にすることによって、マックスウェルの悪魔を実現することができる。

こうしてたどり着いた「悪魔の装置第二号」は、先の第二章で見出された「悪魔の装置第一号」と本質的に似通ったものとなった。ここで重要なのは「ゲート」や「ハンドル」といった装置の詳細ではない。「1周期に要する時間の長さが不確定」という、悪魔の装置特有の性質こそが重要なのである。

悪魔に至る道程では、1ー3：までがいわゆる定説であり、4ー6：に独自の主張が込められている。ここでの主張と定説とでは、悪魔の擁護と撲滅であり、真っ向から対立することになる。しかし、仮にここでの主張が正しかったとしても、定説が「間違っていた」ことにはならない。結論に至るまでの思考過程を見れば、ここでの主張はむしろ定説の延長上に位置することを納得いただけるものと思う。

改めて繰り返すが、可逆コンピュータの持つ最も重要な性質は「経路に合流点を持たないこと」である。単純にマックスウェルの悪魔を形作ろうとしても、どこかに「合流点」が必要となるので矛盾をきたす。これが定説の主張であった。しかし、実のところエネルギーが取り出される時刻をずらしさえすれば「合流点」は必須ではなくなる。装置内を巡回する信号を1つに合わせる代わりに、外界に対して相異なる時刻にエネルギーを放出することによって「装置＋外界」全体の場合の数を保つことができる。それゆえマックスウェルの悪魔は破綻を免れることができるのだ。

「計算時間が予測できないコンピュータは、その存在を熱力学の神から許されている」

これが本章での結論である。