0.3 濃度の話

それでは、もう少し濃度について語ることにしよう。

偶数や奇数は自然数と一対一の対応がつく、という理由から濃度が等しいことを説明した。今度は分数、すなわち有理数の濃度についてだが、結論を先に言えば、有理数の濃度は自然数の濃度に等しい。つまり有理数は自然数と一対一の対応がつくのだ。表を見てもらいたい。

$b \backslash a$	1		2		3		4		5		$\cdots$
1	$\frac{1}{1}$	$\rightarrow$	$\frac{2}{1}$		$\frac{3}{1}$	$\rightarrow$	$\frac{4}{1}$		$\frac{5}{1}$	$\rightarrow$	$\cdots$
		$\swarrow$		$\nearrow$		$\swarrow$		$\nearrow$			
2	$\frac{1}{2}$		$\left(\frac{2}{2}\right)$		$\frac{3}{2}$		$\left(\frac{4}{2}\right)$		$\frac{5}{2}$		$\cdots$
	$\downarrow$	$\nearrow$		$\swarrow$		$\nearrow$					
3	$\frac{1}{3}$		$\frac{2}{3}$		$\left(\frac{3}{3}\right)$		$\frac{4}{3}$		$\frac{5}{3}$		$\cdots$
		$\swarrow$		$\nearrow$							
4	$\frac{1}{4}$		$\left(\frac{2}{4}\right)$		$\frac{3}{4}$		$\left(\frac{4}{4}\right)$		$\frac{5}{4}$		$\cdots$
	$\downarrow$	$\nearrow$									
5	$\frac{1}{5}$		$\frac{2}{5}$		$\frac{3}{5}$		$\frac{4}{5}$		$\left(\frac{5}{5}\right)$		$\cdots$
$\vdots$	$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\vdots$		$\ddots$

( ) と矢印を無視しておけば、ここにはすべての有理数があることが分かるだろう。もっとも 0 と負の有理数を省いているが、この先の話の本質を変えることはないし、簡単のために正の有理数だけで説明を続けるとしよう。

有理数の濃度が自然数の濃度に等しいことを示すには、表にあるすべての有理数に 1, 2, 3, ... と番号が振れることを示せばよいわけである。それには、表の有理数を矢印の順に沿って番号を付けるだけで済む。( ) で囲った数は、すでに番号が振られている有理数と同じ値なので、飛ばしておく。すると有理数が自然数に一対一に対応していることが分かるはずだ。えー？　そうかな？　このような番号の振り方では、対角線の左上に番号が付いても、それと同じ量の番号の付いてない数が右下に残るんじゃないの？　そうだね。確かにそんな気がするかもしれない。でも、それはわれわれが無限をよく理解できないからである。この番号の付け方で、もれなく有理数には番号が付くのだ。いくら右下に無尽蔵の有理数があっても、自然数だって無尽蔵にあるんだから。

すると今度は小数の番である。小数は有理数と違い、自然数によって番号を振ることができない。カントールは次のように説明している。

いま、すべての小数にもれなく番号を振ることができたと仮定しよう。そしてすべての小数を番号順に

$$\begin{aligned} d_1 &= 0.\alpha_1\alpha_2\alpha_3\alpha_4\ldots \\ d_2 &= 0.\beta_1\beta_2\beta_3\beta_4\ldots \\ d_3 &= 0.\gamma_1\gamma_2\gamma_3\gamma_4\ldots \\ d_4 &= 0.\delta_1\delta_2\delta_3\delta_4\ldots \\ &\vdots \end{aligned}$$

と、並べたとする。そして、これらの数をもとに、次の操作で新たな小数 $\dot{x}$ を作る。$\dot{x}$ の小数第 1 位の数字は、d_1 の小数第 1 位の数字と “違う” 数字 $\dot{\alpha}_1$ を選ぶ。また、小数第 2 位の数字は、d_2 の小数第 2 位の数字と “違う” 数字 $\dot{\beta}_2$ を選ぶ。このように、$\dot{x}$ の小数第 n 位の数字には、n 番の番号が付いた数の小数第 n 位とは “違う” 数字を選ぶのである。すると新しい数

$$\dot{x} = 0.\dot{\alpha}_1\dot{\beta}_2\dot{\gamma}_3\dot{\delta}_4\ldots$$

が出来上がるが、$\dot{x}$ はこれまでに番号を振られた数の中にない数である。

ちょっと変だぞ。初めの仮定では、すべての小数にもれなく番号が振られたはずなのに、いま出来上がった $\dot{x}$ は番号を振られた数とは明らかに違っている。この矛盾が生じた理由は、最初にすべての小数に番号を付けることができると仮定したからだ。仮定は間違っている。すなわち、小数には自然数と同じ番号を振ることができない。つまり、小数の集合と自然数の集合では濃度は違うのだ。それも小数のほうが “濃い” 濃度を持っていることになる。

集合の濃度といっても無数に**要素**を含んでいるので、その量を数値で表すことなどできない。そこでわれわれは、自然数の集合の濃度を $\aleph_0$—$\aleph$ はヘブライ文字で “アレフ” と読む—で表すことにしよう。すると、これまでに分かったことから、偶数の集合と奇数の集合の濃度も、さらには有理数の集合の濃度も $\aleph_0$ である。しかし小数の集合の濃度は $\aleph_0$ ではない。

おおまかに言って、小数で表すことができる数は**実数**と呼ばれている。もちろん有理数は実数である。なぜなら $\dfrac{1}{3} = 0.333\cdots$ や $5 = 5.0$ のように、あらゆる有理数は小数表記にできるからだ。さて、実数の集合の濃度は $\aleph_0$ ではないので、今度はそれを $\aleph$ で表すことにしよう。これまでの話では、どうやら

$$\aleph_0 < \aleph$$

であるようだ。

すると、ちょっとした疑問が湧いてくるだろう。$\aleph$ より "濃い" 濃度を持つ集合はどんな集合だろう。また、$\aleph_0$ と $\aleph$ の間の濃度を持つ集合はあるのだろうか。残念なことに、その話題を続ける力量が私にはない。この先の大秘境に興味がある人は、足を踏み入れてみてはいかが？

濃度の話から少し離れてしまうかもしれないが、数学において一対一の対応は重要な概念である。有理数の濃度が自然数の濃度と同じというのは、有理数が 1 ずつ増える自然数の性質を継承している表れでもある。TEX では 1 ずつ増える処理をどうするのだろう。簡単な例で見ておこう。

```
\newcommand\countup{%
  \newcount\n
  \loop \number\n\ \advance\n1 \ifnum\n<100 \repeat
}
```

このマクロは、0 から 99 までを出力するもので、『じゃあ、数えるよ! \countup!』と書けば『じゃあ、数えるよ! 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 !』となる。出力された数の列は文章の一部となって、適切に改行されているのが分かるだろう。

まず、変数として\n を定義した。いきなりの初登場が\loop〜\repeat 命令だ。\loop 命令は繰り返しをするときに\repeat とセットで使われる。繰り返しは、\repeat までに書かれた文を、\ifnum の条件を満たす限り—ここでの条件は\n<100 だから\n が 99 になるまで—逐次行う。\ifnum が TEX 流の if 文と思ってよい。また、このマクロは\number\n\ \advance\n1 のあとで\ifnum の判定をするので、最低 1 回は\number\n が実行されることにも注意してほしい。もし\ifnum が\loop 直後にあれば、判定後に命令群が処理される。もっともこの場合は、どっちにしても同じ結果になるけれど。

さて、\ifnum による判定が**真**である限り、\loop〜\repeat にはさまれた文が実行され、\n の値を出力して\n の値を 1 増やすことは前のマクロと変わらない。\number\n の直後に "\ " が挿入されていることに注目してほしい。もしここが "\number\n \advance\n1" と書かれていた場合、間の空白は単に前の命令と後ろの命令の区切りでしかなく、その結果、0123456... のような出力になる。そのために、明示的に空白を挿入したのである。以前登場した "\number\n, \advance\n#1" の場合、"," の次の空白は "," に続く "空白文字" である。この辺の感覚は慣れるしかないだろう。

ところで、いまは 1 ずつ回数が増えていく様子を見たが、2 ずつ増やしたいときもある。その場合は、\advance\n1 を\advance\n2 とするとよい。これで偶数だけが出力される。また、見やすさを考慮するなら、\advance \n 2 と書くのもよいだろう。

もし、99 から 0 まで逆順に出力させるなら、`\n=99` から始めて`\advance\n1` を`\advance\n-1` にすればよいが、`\ifnum` を放っておくと無限ループに近い状態に陥ってしまう。なぜなら`\n` が 99、98、... と進んだ場合、いつだって`\n<100` だからだ。ここは`\ifnum\n>-1` に直さなくてはならない。

```
\newcommand\countdown{%
  \newcount\n \n=99
  \loop \number\n\ \advance\n-1 \ifnum\n>-1 \repeat
}
```

これで、『今度は逆に数えるよ! `\countdown!`』と書けば『今度は逆に数えるよ! 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 !』が出力される。

あれ？　さっきは 0 からの出力だったのに`\n=0` は書いてなかったよね。`\n=0` の記述がなくても 0 から出力された理由は、`\newcount` で変数を定義すると 0 で初期化されていたからである。でも、本当は`\n=0` を記述して初期化しておくのが望ましい。また、TEX での数値比較は “<”、“>”、“=” のいずれかであり、“>=” のような演算子は使えない。そのため、`\ifnum` の判定は`\n>=0` ではなく`\n>-1` にしたのである。