1.2 無限と有理数

さあ整数が出揃って、いよいよ数学の世界の奥へ入っていく準備ができた。数は整数の他にも、様々な種類のものがあることは知っているだろう。続けていろいろな種類の数を紹介したいけれど、まだ整数についてさえ十分に語っていないのだ。それは何かって？　それは "..." の部分。特に "..., −10" と "10, ..." のところは、単なる省略ではないことを言いたいのである。

ここに使われた "..." は、この先の数を書く意味がないことを示している。なぜなら、書き切ることができないから。当たり前だよね。ところが "..." には、思いもかけない不思議が詰まっている。

君たちは**偶数**と**奇数**は知っているね。自然数に限れば 2, 4, 6, ... が偶数、1, 3, 5, ... が奇数ということになる。ものの集まりを**集合**と呼ぶことにして、偶数の集合を $\mathcal{E}$、奇数の集合を $\mathcal{O}$、そして自然数の集合を $\mathcal{N}$ で表せば

$$\mathcal{E} + \mathcal{O} = \mathcal{N}$$

であることが想像できる。でも、残念ながら違う。想像が事実に合わないことはよくある話。単純な足し算は、偶数や自然数のような**無限集合**にはそぐわない関係なのだ。それについてカントール[1]は次のように説明している。

たとえば "りんごが 5 個ある" とは、各りんごに自然数 1, 2, 3, 4, 5 が**一対一に対応**していると見る。このとき、りんごの数 （かず）は 5 に等しいと言える。同様のことを偶数と自然数で考えてみよう。すると

$$\begin{array}{lccccccc} \text{偶数} & 2 & 4 & 6 & 8 & 10 & \cdots \\ & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \cdots \\ \text{自然数} & 1 & 2 & 3 & 4 & 5 & \cdots \end{array}$$

のように、各偶数に自然数 1, 2, 3, 4, 5, ... が一対一に対応していることになる。したがって偶数と自然数は同じ数（かず）だけある、と考えるわけだ。このときに数ということばは適切でないように思えるので、カントールは**濃度**という用語を使っている。すると奇数に対しても、奇数と自然数は同じ濃度を持つと言えるのだ。

この用語はまったくもって適切で、われわれの生活感覚にも合っている。なぜなら、同じ濃度の食塩水を混ぜると、同じ濃度を持つ食塩水ができるのだから。そして食塩水の量も増えているので、自然数も偶数だけの "量" に比べると、きっと奇数の分だけ "量" が増えているんだろう。

[1] ゲオルク・カントール（1845–1918）：ドイツの数学者。

ちょっと不思議な感覚だけど、日常からかけ離れたものにはありがちなことではある。（想像を超える大金持ちの資産）＋（想像を超える大金持ちの資産）は、やはり（想像を超える大金持ちの資産）であるのと同様、“...” はわれわれの想像を超えた世界ということなのだ。

さて、**Haskell** が巨大な数を操れることは述べたが、微小な数はどこまで扱えるのだろう？　小さな値になる計算をさせてみよう。

(ghci env.)

```
Prelude> 1/3000000000000000000000000000000000000000000000000000000000000000000000
3.333333333333334e-57
Prelude> 1/10^309
0.0
```

計算結果の `3.333333333333334e-57` は $3.333333333333334 \times 10^{-57}$ を意味する。巨大な数の表し方のように `0.(`ここに `56` 個の `0)3333` と表示してくれない上に、やはり末尾が “`...34`”っていうビミョーな値だ。

でも、それ以上に困るのは `1/10^309` の計算が `0.0` と見なされてしまうことだ。`10^1000` なら難なく計算するのに、である。実際 **Haskell** では、$a \times 10^n$ の表示形式で扱える数の下限はおよそ 10^{-308} 程度である。もっとも、これだけの精度があれば日常的な計算にはまったく不自由しないけれど。

そもそも、無限の大きさや無限に続く計算はコンピュータには向いていない。必ずどこかで計算を打ち切らなくてはならないからだ。では、もし計算を打ち切らなくて済むならどうなるんだろう。たとえば

$$1-1+1-1+1-1+1-1+1-1+1-1+\cdots$$

のように、無限に続く計算はできるのだろうか。先頭から順に見れば

$$\begin{aligned}(1-1)+(1-1)+(1-1)+(1-1)+\cdots &= 0+0+0+0+\cdots \\ &= 0?\end{aligned}$$

のように思える。でも、くくり方をずらせば

$$\begin{aligned}1+(-1+1)+(-1+1)+(-1+1)+(-1+\cdots &= 1+0+0+0+\cdots \\ &= 1?\end{aligned}$$

とも思える。どっちが正しいのか分からないけれど

$$1-1+1-1+1-1+1-1+1-1+1-1+\cdots = x$$

とおけば、これを

$$\begin{aligned} 1-(1-1+1-1+1-1+1-1+1-\cdots) &= x \\ 1-x &= x \end{aligned}$$

と見直すことさえ可能だ。だったら $x=\frac{1}{2}$?。

うわあ、一体どうなっているんだろう。無限の足し算には普通の計算法則が成り立たないのか、無限の計算だから答も無数に出るのか...。いずれにせよ、従来の考えが成立しないことは確かである。庭いじりが造園みたいな大規模作業になっても困るので、この話は一旦打ち切っておこう。

ところで **Haskell** は無限集合を扱える、と言うと驚くかもしれないね。**Haskell** は厳密に集合を扱えるわけではないけれど、集合に近い扱いができる**リスト**がある。リストは `[1, 2, 3, 4, 5]` や `["red", "blue", "yellow", "green"]` のような記述をする、もののリストである。

おほん！　でも **Haskell** のリストは単なるリストとはひと味違うんだ。とくに整数値のリストには便利な記述法がある。等間隔の数値に限るが、".." を書けば間を補ってくれるのだ。

(ghci env.)

```
Prelude> [1..5]
[1,2,3,4,5]
Prelude> [3, 7..20]
[3,7,11,15,19]
Prelude> [20, 17..0]
[20,17,14,11,8,5,2]
```

`[1..5]` なら 1 ずつ増加する数列を表示し、`[3, 7..20]` なら最初の 2 数から数列の間隔を予想してくれる。減少する数列も同じだ。

そして、驚くことに無限列のリストが扱えるのだ。なんと `[2, 4..]` は 2 以上の偶数の、無限のリストである。あ、でも待って！　これを GHCi に食わせないように。そんなことをすると、**Haskell** はいつまでも偶数を取り出し続けてしまうから。うっかり入力してしまったら "ctrl+c" キーで止めよう。

それだったら `[2, 4..]` みたいな書き方は意味ないと思うでしょう。その通り。無限列のリストはそのまま使っては意味がない。この場合は次のように使う。

(ghci env.)

```
Prelude> take 12 [2, 4..]
[2,4,6,8,10,12,14,16,18,20,22,24]
```

`take` n で無限列の先頭から n 個の要素を取り出すのである。これは **Haskell** の**遅延評価**と呼ばれる機能で、実際に値が必要になることが分かるまで—この例は `take 12` で 12 個の値が必要なこ

とが分かるまで—処理を特定しなくてよいので、無限集合を記述しても平気なのだ。追々、遅延評価の例も出てくるだろう。

実際の数学においても、無限は必ずしも扱いにくいものではない。なぜなら $0.333\cdots = \frac{1}{3}$ だから、小数点以下が無限であっても何ら問題ない。このようにわれわれは、端数を表すために**分数**や**小数**を使っている。身近に分数や小数を使っていると、それらの数は自然発生的に生まれたように感じてしまうものだ。しかし、分数や小数はわれわれ人間が作り出した数なのである。

分数や小数はどちらも “数” と名乗っているものの、性格には大きな違いがある。いまでは $\frac{1}{8}$ と 0.125 は、どちらも同じ量を表す数の扱いを受けている。けれども $\frac{1}{8}$ は、8 つの量に対する 1 の量という比を意味し $1:8$ と表す。すなわち $\frac{1}{8} = 1:8$ だ。一方 0.125 は、1 ほどに大きくない量の 0.1 に、0.1 ほどに大きくない 0.02 を加え、さらに 0.01 ほどに大きくない 0.005 を加え合わせた量である。どっちも同じじゃないかって？　そう、量という点ではね。

結局のところ、分数でも小数でも細かい量—これは精密な量と言っても構わない—を表せることに変わりがない。たとえば $\frac{10}{3}$ や 3 という値は、**円周率**をおおまかに表している。もちろん円周率というのは、直径に対する円周の比のことだ。でも、これよりも $\frac{22}{7}$ や 3.14 とすることで、さらに $\frac{355}{113}$ や 3.141592 とすることで、円周率をより精密に表せる。ここで注目してもらいたいのが、分数の作り方である。分数はあくまでも自然数の比で表している。一方、小数は数字を増やすことで対処しているが、これは小数に小数を足していく操作になっている。つまり、分数が自然数を継承して作られる数であるのに対し、小数は自分自身を継承している点に注意してほしいのだ。ここに分数と小数の性格の違いを見ることができるわけだ。

今日では、われわれは $a:b$ の比は $\frac{a}{b}$ という分数で表すのが一般的になっている。このように整数 a, b（ただし $b \neq 0$）を用いて $\frac{a}{b}$ で表すことができるすべての数を**有理数**と言う。有理数の呼称は慣れないと戸惑うかもしれない。$\frac{a}{b}$ で表した数は a rational number（比の数）と呼ばれるが、日本語訳が有理数となっているだけのことなのだ。すると、この約束に従えば -5 は $\frac{-5}{1}$、0 は $\frac{0}{1}$ と表すことができる数なので、整数も有理数であると言えるのである。