# 4　$1+1$（その 1）

では、集合をもとに数を構成してみよう。なぜ数の構成に集合を使うのか、疑問に思うかもしれない。しかし、もっともな理由がある。それは、数は概念だからだ。

日常において 1 という数は、たとえばひとつのりんご、たとえば一杯の水、たとえば気温の 1°C、... など具体的な対象物を指すので、概念というほうがおかしな気がするかもしれない。だが、そうではないのだ。ふたつのりんご、2 杯の水、気温の 2°C には、数 2 をあてがうことができる。では、ひとつのりんごと一杯の水に 2 をあてがえるだろうか。もちろん、具体物がふたつあるということで 2 をあてがうことは可能だろうが、私たちの生活感覚からすると無理があると思う。まして、ひとつのりんごと気温 1°C で 2 を想像するのは、いくらなんでも常軌を逸している。

数は概念である。私たちは、概念を当てはめることができるものに対して数をあてがっているのである。数を概念で表現するにはどうするのだろう。

まず、数の基本単位として空集合 $\{\ \}$ をあてがうことにしよう。基本にする集合は、$\{A\}$ であっても $\{1\}$ であってもよいけれど、そうすると、まず $A$ や 1 が何であるか明らかにする必要が生じてしまう。その点、空集合なら要素を持たない集合という、きとんとした定義ができるので適切と思えるからだ。その上で、$\{\ \}$ に要素 $\emptyset$ を含む集合 $\{\emptyset\}$ を考えよう。この瞬間、$\emptyset$ の次に $\{\emptyset\}$ が作られたことになる。この旅では、“(何か) の次を作る操作” を “(何か)$^+$” と表現しておく。$\{\ \}^+ = \{\emptyset\}$ ということである。これを繰り返せば、きっと数の集合ができあがるだろう。

さて、$\emptyset$ を 0、$\{\emptyset\}$ を 1 と呼ぼう。$\emptyset$ や $\{\emptyset\}$ は数の概念そのものであるが、概念だけで考えを進めるのは難しい。そのために、概念を区別できるように呼び名—すなわち**数詞**—を与えるのである。

1 の次、つまり $1^+$ はどのような操作になるだろう。それは $\{\emptyset\}$ を作ったときと同じく、いま存在する集合 $\{\emptyset\}$ にものを放り込めばよい。といっても、使える概念は集合であり、かつ集合しかないことを思い出そう。$\{\emptyset\}$ に放り込めるのは $\emptyset$ だけであるから、$1^+ = \{\emptyset, \emptyset\}$ だ。しかし、集合の概念はどれだけのものが集まっているかではなく、どんなものが集まっているかであったはずだ。つまり、$\{\emptyset, \emptyset\}$ は $\{\emptyset\}$ と同じことである。これでは $1^+$ は 1 の次ではなく、1 そのものになってしまう。このことから、$\{\emptyset, \emptyset, \ldots\}$ のように、次々 $\emptyset$ を放り込む方法では、0, 1 から先の数が作れないことが分かる。

さあ困った。どうすれば $\emptyset$ と集合の概念だけで 1 の次の数を定義できるのだろうか。こんな考えはどうだろう。いま存在する集合に $\emptyset$ を放り込むのではなく、いま存在する集合に $\emptyset$ を加えた集合を作る、というものだ。

これなら、$\emptyset$ の次は $\{\emptyset,\emptyset\}=\{\emptyset\}$ であり、その次は $\{\{\emptyset\},\emptyset\}$ である。前に述べたように $\{\emptyset\}$ と $\emptyset$ は別ものであることに注意しよう。そして、ここには $\emptyset$ と集合の概念以外は何も使われていない。この調子で、その次も定義できる。その次は、$\{\{\emptyset\},\emptyset\}$ に $\emptyset$ を加えた集合—すなわち $\{\{\{\emptyset\},\emptyset\},\emptyset\}$ —である。これで数は

$$\emptyset,\quad \{\emptyset\},\quad \{\{\emptyset\},\emptyset\},\quad \{\{\{\emptyset\},\emptyset\},\emptyset\},\quad \dots$$

と、無限に続くことになった。

□ **数の定義**

- 数は $\emptyset,\quad \{\emptyset\},\quad \{\{\emptyset\},\emptyset\},\quad \{\{\{\emptyset\},\emptyset\},\emptyset\},\quad \dots$ と定義できる

**PowerShell** で、この様子を眺めてみよう。けれど、集合自身に新たな要素を加えることを、**PowerShell** でどう実現するのだろう。それは、次のようにする。

[ps script]

```
PS C:\Users\Yours > $one = 0
PS C:\Users\Yours > $two = $one, 0
PS C:\Users\Yours > $thr = $two, 0
PS C:\Users\Yours > $thr
0
0
0
```

まず、0 を $\emptyset$ と考えよう。そして$one に 0 を代入して、1 である $\{\emptyset\}$ を定義したと考えてもらいたい。このとき、one は集合である。$two が集合$one に要素 0 を加えたことは見て分かるだろう。$thr も同様で、集合$thr には確かに $\emptyset$ が 3 つ入っているようだ。しかしこれだけでは、$thr が $\{\emptyset,\emptyset,\emptyset\}$ なのか $\{\{\{\emptyset\},\emptyset\},\emptyset\}$ なのか区別がつかない。"$thr"と入力してもよいけれど、sort を用いて確認してみよう。

[ps script]

```
PS C:\Users\Yours > $thr | sort -uniq
0
0
0
```

-uniq をつけて sort したのに、相変わらずの結果である。これは重複がなかったので、削除するものがなかったことを意味する。つまり、ここに見える 0 はそれぞれ別ものであるということだ。実際 **PowerShell** では、これらはネストされた配列になっている。

$thr の次は$fou であるが、せっかくいくつかの数がそろってきたので、$fou は$thr に 0 を加えるのではなく、$two に$two を加えて作ってみよう。私たちの感覚で $2+2$ を作ることになる。$1+1$ じゃなくて悪いけど、後の都合があるので勘弁してもらおう。さて、fou への代入がうまくいけば、そこには 4 つの 0 が入る

はずだ。

[ps script]
```
PS C:\Users\Yours > $fou = $two, $two
PS C:\Users\Yours > $fou
0
0
0
0
```

確かに$fou は 4 つの 0 を含んでいるから、$2+2=4$ ができたように思える。でも、結論を下すのは次のことを確認したからだ。重複しているものがあるかもしれないので、sort してみよう。

[ps script]
```
PS C:\Users\Yours > $fou | sort -uniq
0
0
```

何てことだ。重複を削除すると $2+2=2$ になっているじゃないか。それもそのはずで、ここで構成した足し算は $\{\{\emptyset\},\emptyset\}$ に $\{\{\emptyset\},\emptyset\}$ を加えた集合になっているので、結局 $\{\{\emptyset\},\emptyset\}$ でしかないのである。いま行った足し算とは、$\{\{\emptyset\},\emptyset\}+\{\{\emptyset\},\emptyset\}$ ではなく $\{\{\emptyset\},\emptyset\}\cup\{\{\emptyset\},\emptyset\}$ だったのだ。

とりあえず、$\emptyset$ と集合の概念だけで数を構成したものの、この方法では日常的な足し算にならない。だからといって、これがまるで無意味ということではない。これはこれで数の体系をなすけれど、私たちが馴染んでいる演算とは様子が違っているということだ。

□ **数の和**

- 数を $\emptyset$ と集合の概念で定義して、和集合をもって和としても、$1+1=2$ と言えない