

今日からは、現代的な「体論」に入るが、

その前にちょっとお話から。

「数」とは何だろうか。

(というか、

我々が普通「数」だと思っているものは、

どのようなものだっただろうか。)

「数」体系の拡張

N : 自然数全体

\cap

Z : 整数全体 (整数環)

\cap

Q : 有理数全体 (有理数体)

\cap

R : 実数全体 (実数体)

\cap

C : 複素数全体 (複素数体)

$$N \subset Z \subset Q \subset R \subset C$$

我々がこれを

「数」体系の拡張

と想ってきたのは、

新たに付け加わったもの達も

「数」

だと思ってきたということである。

ところが …

「数」? の例:

合同式 ($a \equiv b \pmod{m}$) は有用であった。

(現代風に言えば、剰余環 $\mathbb{Z}/m\mathbb{Z}$ を考えた。)

一方、 R 内では $X^2 + 1 = 0$ に解がなかったが、

この“解”を仮想すると便利なので、

$i = \sqrt{-1}$ を「数」だと認識した。

さて、例えば

$\mathbb{Z}/3\mathbb{Z}$ 内には、やはり $X^2 + 1 = 0$ の解がない。

X	$X^2 + 1 \pmod{3}$
0	1
1	2
2	2

では、ここに “ $\sqrt{-1}$ ” を付け加えて

「数」を拡張することが出来るのか。

(この “ $\sqrt{-1}$ ” は「数」なのか?)

別の例:

実数は

有理数 (特に十進有限小数) の極限

として定式化された。

$$\begin{aligned}\pi &= 3.141592653598793228462643383279 \dots \\ &= 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2} + \dots\end{aligned}$$

計算は、適当な精度で途中で打ち切って行なう。

一方、20世紀初頭に Hensel は、

次のような“極限”を考えることを提唱した。

素数 p に対し、

$$x = a_0 + a_1p + a_2p^2 + a_3p^3 + \cdots$$

\cdots p 進数 (p -adic numbers)

任意の N に対し、 $\text{mod } p^N$ での計算は、

その“精度”で途中で打ち切って出来る。

$X^2 = -1$ を $\text{mod } 5^N$ で解こう。

$$-1 \equiv (2 \cdot 5^0)^2 \pmod{5^1}$$

$$-1 \equiv (2 \cdot 5^0 + 1 \cdot 5^1)^2 \pmod{5^2}$$

$$-1 \equiv (2 \cdot 5^0 + 1 \cdot 5^1 + 2 \cdot 5^2)^2 \pmod{5^3}$$

...

$$-1 = 2 \cdot 5^0 + 1 \cdot 5^1 + 2 \cdot 5^2 + \dots$$

この最後の式の右辺

$$2 \cdot 5^0 + 1 \cdot 5^1 + 2 \cdot 5^2 + \dots$$

は「数」なのか？

反省:

我々はどんなものを「数」と思ってきたか？

“「数」の範囲”の満たすべき性質は？

→ 不自由なく計算 (四則演算) が行える

→ 公理化 (公理的な「体論」の誕生)

体: **Körper** (独), **corps** (仏), **field** (英)

「体」の公理

$K = (K, +, \cdot, 0, 1)$ が (可換)体 とは、

(1) $(K, +, 0)$: 可換群

(2) $(K \setminus \{0\}, \cdot, 1)$: 可換群

(3) $\forall x, y, z \in K$:

$$x(y + z) = xy + xz, (x + y)z = xz + yz$$

「群」の公理

$G = (G, *, e)$ が 群 とは、

(1) $* : G \times G \longrightarrow G$: 二項演算

(2) $\forall x, y, z \in G : x * (y * z) = (x * y) * z$

(3) $\forall x \in G : e * x = x * e = x$

(4) $\forall x \in G : \exists y \in G : x * y = y * x = e$

更に次も満たすとき 可換群 と言う。

(5) $\forall x, y \in G : x * y = y * x$