11　証明の形式化 2――言語

前回は命題を記号であらわすことについて説明しました。復習として，次の命題（ちなみにいずれも真）がどのように記号化されるか，あらためて考えてみましょう。

- 自然数に関する命題
  (a)　$1+1=2$ である。　(b)　5 は奇数である。　(c)　最大の自然数は存在しない。
- 実数に関する命題
  (d)　$(-1)^2=1$ である。　(e)　正の実数の逆数は正である。
  (f)　$x^3=2$ をみたす実数 x は一つしかない。

ところで，「通常の数学で扱う命題とは，基本的な命題から $\wedge$，$\vee$，$\neg$，$\to$，$\forall$，$\exists$ を使って作れるような命題だ」ということを前回いったのですが，ここで「基本的な命題とは何か」という疑問が生じます。この疑問を解消するために，「言語」の概念を紹介しておくことにします。

言語とは

どんな数学的対象について語りたいのかということに応じて，命題を書きあらわすために使いたい記号や表現もかわってきます。

たとえば自然数に関する命題なら，個々の自然数 0，1，2，……，および自然数をあらわす変数のほかに，$+$，$-$，$\times$，$=$，$<$，$>$ といった記号がほしいでしょう。もっと複雑な概念をあらわす記号も用意しておきたいと考える人もいるかもしれません。実数に関する命題とか，ユークリッド平面の幾何学に関する命題であれば，また別の記号や表現が必要になってきます。

そこで，どんな数学的対象に関する命題を書きたいのかを念頭において，使ってよい文字や語彙を決めます――それを**言語**といいます。下記の方針をとることにします。

- 言語は「定数記号」，「関数記号」，「述語記号」からなるものとします。変数記号は言語の概念には含めず，どんな言語を採用する場合も，a，b，c，……とか x，x'，x''，……など，いくらでも多くの変数記号が供給され，使用できるのだということにしておきます。
- 記号を増やすことには利点もありますが，論理の分析という観点からは好ましくありません。ここでは，記号はなるべく精選することにします。

すると数学の命題は，「ある言語を用いて一定の構文規則にしたがって作られた記号列」としてあらわされることになります。これは純粋な記号列であって，そこに「意味」はありません。意味を剥ぎとられた記号列を考えていることを強調するために，上記のような記号列のことを**閉論理式**とよぶことがあります。

自然数に関する命題を書きあらわすための言語

たとえば，次の言語 $\mathcal{L}_{\mathrm{A}}$ を使えば，自然数に関する命題を書きあらわすことができそうです。

言語 $\mathcal{L}_{\mathrm{A}}$

- 定数記号：0，1，2，3，……
- 関数記号：+，×
- 述語記号：=，<

+ が「関数記号」であるというのは，「入力された 2 つの自然数に対し，それらの和を出力する」という 2 変数関数をあらわしているからです。関数の一般的記法では $+(x, y)$ と書くべきところを，慣習により $x + y$ と書いてもよいものと約束します。× も同様です。

「述語記号」は何らかの主張をあらわすのに用いる記号です。こちらも，$=(x, y)$ と書くほうが理に適っている（かもしれない）ところを，慣習により $x = y$ と書くのだと考えます。

0，1，2，3，……という定数記号に関しては，「次の自然数」をあらわす 1 変数関数記号 suc を用意して，0 以外は suc(0)，suc(suc(0))，……と書くことも考えられるでしょう。したがって，言語 $\mathcal{L}_{\mathrm{A}}$ を修正した次の言語 $\mathcal{L}'_{\mathrm{A}}$ を採用するのも悪くない考えです。

言語 $\mathcal{L}'_{\mathrm{A}}$

- 定数記号：0
- 関数記号：suc，+，×
- 述語記号：=，<

問題

1. 命題 (a)，(b)，(c) を，言語 $\mathcal{L}_{\mathrm{A}}$ にもとづく閉論理式としてあらわしてください。

2. 実数に関する命題を書きあらわすための言語を考案して，命題 (d)，(e)，(f) を，それにもとづく閉論理式としてあらわしてください。

次回予告

ここまで，命題を記号列として（閉論理式として）あらわす方法を整理してきました。それをふまえて次回は，証明を記号列の操作としてとらえること（これが「証明の形式化」です）の説明に入ります。いくつかの方法がありますが，ゲンツェンの「自然演繹」を採用することにします。