

TEX でマクロ

栗野 俊一*

Ver. 0.01 (2012/06/12 版)

1 TEX でマクロ

1.1 長い数式

TEX で長い式を入力する事を考えます。例えば、数列 a_n などはどうでしょうか。

数列 a_n の例 (TEX) —————

```
\[
\{a_n\} = a_1, a_2, \dots, a_i, \dots
\]
```

数列 a_n の例 (整形後の文章) —————

$$\{a_n\} = a_1, a_2, \dots, a_i, \dots$$

1.2 マクロで入力をサボる

TEX で複雑な式を何度も入力するのは大変^{*1} です。

そこで、次のように複雑な数式には名前をつけてしまいましょう。

以下では、長い数式に、`\SeqA` という名前を付けています。名前を付けるには、`\newcommand` を使います。

`\newcommand` の後ろの一つ目のプレース^{*2} に囲まれた所^{*3} には新しい名前 (マクロ名)^{*4} を、二つ目には、名前を付ける数式^{*5} を指定します。

* 日本大学理工学部数学科 講師 (kurino@math.cst.nihon-u.ac.jp)

*1 「何が大変か？」というと、別に「長い物を入力する事」が大変なわけではなく、「その長い物が等しいという事を保証する事 (等値性)」が大変なのです。マクロを使えば、「同じ内容を表現する場合は同じマクロを使う」事で、簡単に等値性が保証できます。

*2 「{」と「}」の事。

*3 これを「引数 (ひきすう)」とよびます。

*4 マクロ名には、 \yen で始まる英文字の並び (数字は使えない) を指定します。最初の文字を大文字にしておけば、標準のマクロと区別する事ができます。

*5 もちろん、数式に限定する必要はなく、TEX で利用できる普通の表現 (全てではなく制限がありますが、まあ、良く利用するものは大概大丈夫..) はなんでも、書込む事ができます。

この`\newcommand`の後^{*6}からは、この名前で式を参照する事ができるようになります。

この新しく、「式の代わりに利用できるようになった名前」を「マクロ」とよびます。

数列 a_n をマクロで定義 (TeX)

```
\newcommand{\SeqA}{\{a_n\} = a_1, a_2, \dots, a_i, \dots} % 数式に名前を付けた  
\[  
\SeqA % 名前で数式を呼び出す  
\]
```

二度目からが便利になる。

```
\[  
\SeqA % 名前で数式を呼び出す  
\]
```

数列 a_n をマクロで定義 (整形後の文章)

$$\{a_n\} = a_1, a_2, \dots, a_i, \dots$$

二度目からが便利になる。

$$\{a_n\} = a_1, a_2, \dots, a_i, \dots$$

1.3 似た式なら引数でまとめる

今度は、数列 b_n を考えましょう。数列 a_n 同様にマクロが定義できます。

数列 b_n もマクロで定義 (TeX)

```
\newcommand{\SeqA}{\{a_n\} = a_1, a_2, \dots, a_i, \dots}  
\newcommand{\SeqB}{\{b_n\} = b_1, b_2, \dots, b_i, \dots}  
\[  
\SeqA  
\]  
  
\[  
\SeqB  
\]
```

^{*6} なので、`\newcommand` はプリアンブルに置くのが普通。

数列 b_n もマクロで定義 (整形後の文章)

$$\{a_n\} = a_1, a_2, \dots, a_i, \dots$$

$$\{b_n\} = b_1, b_2, \dots, b_i, \dots$$

しかし、`\SeqA` も `\SeqB` ほとんど同じで、 a が b に変わっただけです。
そこで、このような場合は、違った所を番号 (#1) を付け替えたものにします。

引数付きマクロ (TeX)

```
\newcommand{\Seq}[1]{\{#1_n\} = #1_1, #1_2, \dots, #1_i, \dots} % a,b が #1 になっている
\[
\Seq{a} % a を引数に指定
\]

\[
\Seq{b} % b を引数に指定
\]
```

引数付きマクロ (整形後の文章)

$$\{a_n\} = a_1, a_2, \dots, a_i, \dots$$

$$\{b_n\} = b_1, b_2, \dots, b_i, \dots$$

ついでですから、添字 (i) も指定できるようにしましょう。

複数の引数付きマクロ (TeX)

```
\newcommand{\SeqI}[2]{\{#1_n\} = #1_1, #1_2, \dots, #1_{#2}, \dots}
\[
\SeqI{a}{i} % a,i を引数に指定
\]

\[
\SeqI{b}{j} % b,j を引数に指定
\]
```

複数の引数付きマクロ (整形後の文章)

$$\{a_n\} = a_1, a_2, \dots, a_i, \dots$$

$$\{b_n\} = b_1, b_2, \dots, b_j, \dots$$

2 いつも使うマクロは別ファイルを分けて input する

気に入ったマクロは、何度も使い回す事になるとおもいます。その場合は、そのマクロをファイルに保存しておいて、それを読み込むようにしておくとう便利です。

以下の例では、マクロ `\SeqI` の定義が、ファイル `macro.tex` の中に記述されているという想定です。

その場合には、`\input` を利用して、そのファイルの内容を、読み込む事ができます。

マクロ定義をファイルに追い出す (TEX)

```
\input{mymacro.tex} % \SeqI の定義はこの中にあるものとする
                    % input は通常、プリアブル (\begin{document} の前) に置く
\[
\SeqI{a}{i} % \SeqI を利用
\]
```

マクロ定義をファイルに追い出す (整形後の文章)

$$\{a_n\} = a_1, a_2, \dots, a_i, \dots$$