

文書に Maintenance 性を

東大 TeX 愛好会

Week6

概要

Week2 でも話題になりましたが、「文書のメンテナンス性の高さ」という点において L^AT_EX は優れています。このことを実感できるわかりやすい例を 2 つ取り上げてみたいと思います。

第 I 部

「順列・組み合わせ」の記号

中学や高校の数学で必ずでてくるのが、順列や組み合わせという概念です。授業で一度は

- n 個の元から k 個を選んで並べる順列の総数は ${}_nP_k$ と表す
- n 個の元から k 個を選ぶ組み合わせの総数は ${}_nC_k$ と表す

のように習ったことと思います。しかし、意外と知られていないようですが、この記号は国際的に通用するわけではありません。あまり参照元として好ましくないかもしれませんが、一般の人がよく使用している記号を垣間みることができる Wikipedia を見てみると、このことが実感できます（以下、選ぶ個数を表す k の代わりに m や r が混在しますが、これは Wikipedia の記述が言語によって統一されていないことに由来していますので、気にしないでください）。

日本語版 Wikipedia「順列」では、順列の総数を表す記号として ${}_nP_k$, nP_k , $P_{n,k}$, $P(n,k)$ が挙げられています。同様に、「組合せ」では、選び方の総数を表す記号として ${}_nC_k$, $C(n,k)$ が挙げられています。英語の Permutation, Combination を基にしているので、一見世界共通のように誤解されがちですが、他の言語も見てみるとどうでしょう。

英語 (Permutation)	P_k^n , ${}_nP_k$, nP_k , $P_{n,k}$, $P(n,k)$
フランス語 (Arrangement)	A_n^k
イタリア語 (Disposizione)	$D_{n,k}$
中国語	$P(n,k)$, A_r^n
韓国語	$P(n,r)$

順列については各言語の頭文字が使われ、一見して理解できないものもありますし、組み合わせについても、よく見ると指数の上下が逆になっている例も見受けられ、紛らわしいのが実情です。

確かに、これらの概念の拡張（例えば組み合わせに対応する「二項係数」）には比較的通用する記号が存在し、対応する L^AT_EX マクロが定義されていることもあります。しかし、標準の L^AT_EX には、日本の初等数学で

英語 (Combination)	$C_k^n, {}_nC_k, {}^nC_k, C_{n,k}, C(n,k)$
フランス語 (Combinaison)	C_n^k
イタリア語 (Combinazione)	$C_{n,k}$
中国語	$C(n,k), C_r^n$
韓国語	${}_nC_k, {}^nC_k, C(n,r)$

よく使われる記法を簡略化するマクロがありません。仮に「長文の中で何度も C_k^n という記号を使っていたのに、急遽 $\mathrm{C}(n,k)$ に変更しなければならなくなった！」という場合、どうすればよいでしょう。下付き文字や括弧といったありふれた記法だけを用いるため、ベタ書きしてしまうと探し出すのに一苦労です。

P や C を用いた記号だけでもいろいろな流儀があることを考えると、こういう場合は初めに L^AT_EX マクロを定義してマークアップするのが非常に重要です。

```
%% definition of \tlcomb{}{} for combination
\newcommand{\tlcomb}[2]{%
  \ensuremath{{}_{#1}\mathrm{C}_{#2}}%    definition 1
% \ensuremath{\mathrm{C}(\#1,\#2)}%      definition 2
% \ensuremath{{}^{#1}\mathrm{C}_{#2}}%    definition 3
}
%% definition of \tlperm{}{} for permutation
\newcommand{\tlperm}[2]{%
  \ensuremath{{}_{#1}\mathrm{P}_{#2}}%    definition 1
% \ensuremath{\mathrm{P}(\#1,\#2)}%      definition 2
% \ensuremath{{}^{#1}\mathrm{P}_{#2}}%    definition 3
}
```

このマクロでは、本文中と数式環境中の両方で数式とみなされるように、`\ensuremath` の中に入れて定義しました。もし `\ensuremath` に入れずに定義していたならば、本文中で使用した場合に `_`, `^`, `\mathrm` が

```
! Missing $ inserted.
! LaTeX Error: \mathrm allowed only in math mode.
```

というエラーになるはずです。

第 II 部

「ベクトルの内積」の記号

こちらもマークアップが非常に有効な例です。高校数学で学ぶベクトルの内積ですが、2 つのベクトルの内積の記法として、分野によってさまざまな流儀が考えられます。この場合も、ある記法から別の記法に置き換えるのはベタ書きでは困難です。そこで、マクロの出番です。

```

%%% definition of \tlinner{}{} for inner product
\newcommand{\tlinner}[2]{%
  \vec{#1} \cdot \vec{#2}%          definition 1
% \left( \vec{#1}, \vec{#2} \right)% definition 2
% \left\langle \vec{#1}, \vec{#2} \right\rangle% definition 3
}

```

一見`\cdot`を使うほうが入力が楽だと思うかもしれませんが、メンテナンス性という点では「その`\cdot`が内積を表すのか、それとも積の記号なのか」がわからないという問題があります。マークアップすることで、`\tlinner` の定義さえ変更すれば、文書にでてくるすべての内積の記号を変更できるという安心感が生まれます。

第 III 部

終わりに

今回のマクロは、少し \LaTeX に慣れた方ならすぐに中身を理解できるほど単純なものだと思います。マクロは難しいものと思って敬遠しているとすれば、それはマークアップとしての \LaTeX のメリットを十分活用できていないことになります。

今週のマクロでは会員がさまざまなレベルのマクロを披露しますが、本記事をきっかけにより広く \LaTeX の良さが伝われば幸いです。

```

\documentclass{jsarticle}
%%% definition of \tlcomb{}{} for combination
\newcommand{\tlcomb}[2]{%
  \ensuremath{{}_{#1}\mathrm{C}_{#2}}% definition 1
% \ensuremath{\mathrm{C}(\#1,\#2)}% definition 2
% \ensuremath{{}^{#1}\mathrm{C}_{#2}}% definition 3
}
%%% definition of \tlperm{}{} for permutation
\newcommand{\tlperm}[2]{%
  \ensuremath{{}_{#1}\mathrm{P}_{#2}}% definition 1
% \ensuremath{\mathrm{P}(\#1,\#2)}% definition 2
% \ensuremath{{}^{#1}\mathrm{P}_{#2}}% definition 3
}
%%% definition of \tlinner{}{} for inner product
\newcommand{\tlinner}[2]{%
  \vec{#1}\cdot\vec{#2}%          definition 1
% \left( \vec{#1}, \vec{#2} \right)% definition 2

```

```

% \left\langle \vec{#1}, \vec{#2} \right\rangle definition 3
}
\begin{document}
異なる 6 枚のカードから 3 枚を選ぶ組み合わせの総数は  $\text{tlcomb}\{6\}\{3\}$  通りです。
異なる 12 枚のカードから 5 枚を選んで並べる順列の総数は  $\text{tlperm}\{12\}\{5\}$  通りです。数式中でも使
えます：
\[
\text{tlcomb}\{6\}\{3\} = \frac{6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1}{3!} = 20
\]

2 本のベクトル  $\vec{a}$ ,  $\vec{b}$  が次のように与えられているとき、それらの内積
 $\text{tlinner}\{a\}\{b\}$  を求めなさい。…
\end{document}

```