

TEXの数式の括弧を楽に書こう！

経瑠姉栞羅

<http://herunee-dollar.web.fc2.com/>

herunee-dollar@gmail.com

2010年3月29日

1 TEXへの小さな不満

TEXで数式を書いていると、括弧を書くのが面倒くさくなります。わざわざ`\left(`とか`\right)`とか書かなければいけないからです。別に

$$(x) \tag{1}$$

のようなケースなら問題ありません。ルートや指数なら

$$(\sqrt{x}) + (x^2) \tag{2}$$

となります。ぎりぎりセーフでしょうか？しかし分数になると

$$5 \times \left(\frac{x}{2}\right) + 2 \times \left(\frac{x}{3}\right) \tag{3}$$

となってしまいます。これは

$$5 \times \left(\frac{x}{2}\right) + 2 \times \left(\frac{x}{3}\right) \tag{4}$$

と書くべきですね。

しかし、いちいち`\left(`とか`\right)`とか書くのは面倒です。かと言って、文章を書き終わってから最後に(や)を`\left(`や`\right)`で置換すると、数式が急に読みにくくなってしまいます。数式は元のままで、プリアンブルに

```
\replace{{}{\left{}} %実際にはこんな命令ありません！
```

と一行書けば全て置換できるとか、そういう置換コマンドってないんでしょうか？何かあると思うんですが…。ちなみに

```
\def{\left{}} %この命令は誤りです！
```

とか

```
\renewcommand{{}{\left{}} %この命令も誤りです！
```

とかはアウトでした。

また、仮に置換コマンドがあったとしても、`\newcommand` のオプションの [1] の括弧などは置換してほしくないですね。それに、文章中の (括弧) が置換されてしまっただけは、数式じゃないのに `\left(括弧\right)` とか書くな！なんて言われてしまいます。ですから例えば

```
\replace[align]{(}{\left(} %こんな命令ありません！
```

的なコマンドがあればいいのです。つまり、ある環境の中にある括弧に対して置換するという命令を作りたいのです。

このようなものが $\text{T}_\text{E}\text{X}$ の内部で出来ればいいのですが、私は $\text{T}_\text{E}\text{X}$ のマクロはよく分かりません。そこで、bat ファイルで置換してやろうと思いました。bat ファイルでどう便利になるのかを説明します。

2 使用例

$\text{T}_\text{E}\text{X}$ ファイルに

```
\begin{align}
(1+\frac{1}{6})\times 4=\frac{14}{3}
\end{align}
```

と書いて、bat ファイルで処理する (詳細は後述) と

$$\left(1 + \frac{1}{6}\right) \times 4 = \frac{14}{3} \quad (5)$$

というように、括弧の大きさが `\left(\dots\right)` というように書いたのと同じになります。ちなみに [ブラケット]、{ブレース} も、

```
\begin{align}
\{2\times[3\times(4+\frac{1}{5})+\frac{1}{6}]+\frac{1}{7}\}
\end{align}
```

と書けば

$$\left\{2 \times \left[3 \times \left(4 + \frac{1}{5}\right) + \frac{1}{6}\right] + \frac{1}{7}\right\} \quad (6)$$

となります。もちろん、括弧は左右で対応させなければいけません。数式の途中で改行する場合は `\right.` と `\left.` と書くのをお忘れなく。えっと...便利...でしょ? 人によってはお節介機能なのかもしれませんが。とりあえず align 環境の中の括弧のみ置換します。いまだ `equation` や `eqnarray` なんて使う人はいないでしょうが、`multline` ぐらいなら私もたまに使います。align 環境以外の環境内の括弧も置換したいという方は、bat ファイルを編集して下さい (後述)。

どうしても align 環境の中で、大きさを変えたくない! なんて状況にある方のために `\normal(と\normal)` を用意しました (そのような状況になったことは私はありません)。

ただ、この文章は同じ bat ファイルで処理しているので、最初の三つの式や $\text{T}_{\text{E}}\text{X}$ ソースの部分には `\normal` を用いています。普通の人にはこんなことはないでしょう)。置換して欲しくない場合には使って下さい。同様に `\normal[や\normal]`、`\normal\{、\normal\}` が使えます。コマンドを整理すると次のようになります。

通常	→	私の提案 (ただし align 環境内のみ)
(,)	→	<code>\normal(,\normal)</code>
<code>\left(,\right)</code>	→	<code>(,)</code>
[,]	→	<code>\normal[,\normal]</code>
<code>\left[,\right)</code>	→	<code>[,]</code>
<code>\{,\}</code>	→	<code>\normal\{,\normal\}</code>
<code>\left\{,\right\}</code>	→	<code>\{,\}</code>

如何でしょうか？

bat ファイルで自動で置換したい!と思ったのですが、コマンドプロンプト自体ほとんど使うことがなく、これだけで置換する方法がよく分かりませんでした。そこで、さらに使ったことのない Perl を使いました (おいおい...)

3 ソースコード

bat ファイルの中身です。文章の幅の関係で適当なところで改行していますが、インデントを下げているところは実際には改行ではありません。つまり、8 行で記述されています。

```
perl -p -e "s/\n/kaigyou_temp/g" hoge.tex |
perl -p -e "s/(?:(?<=begin{align}))(.*)?(?:(?=end{align}))
/($tmp=$1)=~s!(\(|\|\\\{)!\\left$2!g; $tmp/egs;" |
perl -p -e "s/(?:(?<=begin{align}))(.*)?(?:(?=end{align}))
/($tmp=$1)=~s!(\)|\|\\\})!\\right$2!g; $tmp/egs;" |
perl -p -e "s/\\normal\\((left|right))/g" |
perl -p -e "s/kaigyou_temp/\n/g" > hoge_temp.tex
platex hoge_temp.tex
platex hoge_temp.tex
del hoge.pdf
dvi2pdf hoge_temp.dvi
del hoge_temp.aux hoge_temp.log
hoge_temp.dvi hoge_temp.out hoge_temp.tex
rename hoge_temp.pdf hoge.pdf
hoge.pdf
```

二行目以降は大丈夫だと思います。dvips などをお使いの方は適当に書き換えて下さい。

一行目はややこしすぎて何が何だか分かりませんね。hoge.tex に対して Perl で置換を 5 回行って hoge_temp.tex に保存しています。

まず初めに改行を適当な文字列 (kaigyoub_temp) で置き換えています。改行があると色々面倒くさいからです。その次に align 環境内の括弧を置換していきます。そして \normal 括弧を適切に処理します。align 環境内で書かれた \normal (はそれまでの置換で \normal\left(となっていますので、\normal\left を取ってやればいいのです。最後に改行を戻しています。

括弧の置換ですが、正規表現を用いています。といっても私はよく分かりません。まずは

```
(?:(<=begin{align}))(.*)?:(<=end{align}))
```

で、align 環境の中身を \$1 に入れています。その部分に対して...置換を...行って...? まあここら辺は私の勉強不足を認めます。とりあえずよく分からないけどどうまくいく的な状態です。すみません。

align 環境以外をお使いの方は、align の所を書き換えるか、書き換えた処理をつけたして下さい。私は multiline も使うので

```
perl -p -e "s/\n/kaigyoub_temp/g" hoge.tex |
perl -p -e "s/(?:(<=begin{align}))(.*)?:(<=end{align}))
/($tmp=$1)=~s!(\(|\|[\|\\\})!\|left$2!g; $tmp/egs;" |
perl -p -e "s/(?:(<=begin{align}))(.*)?:(<=end{align}))
/($tmp=$1)=~s!(\(|\|[\|\\\})!\|right$2!g; $tmp/egs;" |
perl -p -e "s/(?:(<=begin{multiline}))(.*)?:(<=end{multiline}))
/($tmp=$1)=~s!(\(|\|[\|\\\})!\|left$2!g; $tmp/egs;" |
perl -p -e "s/(?:(<=begin{multiline}))(.*)?:(<=end{multiline}))
/($tmp=$1)=~s!(\(|\|[\|\\\})!\|right$2!g; $tmp/egs;" |
perl -p -e "s/\\normal\\((left|right))/g" |
perl -p -e "s/kaigyoub_temp/\n/g" > hoge_temp.tex
```

をいつも使っています。

4 使用方法

Perl をインストールして下さい。といっても簡単です。perl.exe と perl510.dll を環境変数の Path の中のどれか (よく分からないという人は dvi2pdf.exe や platex.exe の入っているところ: \tex\bin など) に入れるだけで使えるようになります。bat+tex ファイルをこちらよりダウンロードして、とりあえず hoge.bat を実行してみてください。一発で hoge.pdf が出来れば万歳ですが、まあうまくいかないでしょう。コマンドプロンプトのどこでエラーが出ているかにもよりますが、次のことを確認して下さい。

- 一瞬でコマンドプロンプトの画面が消える → perl.exe をきちんと用意して下さい。
- perl.exe のシステムエラー → 指示された dll を perl.exe と同じフォルダに用意して下さい。
- Perl の処理はうまくいってるけどそれ以降が... → bat ファイルの platex, dvi2pdf を普段お使いのものに書き換えて下さい。

他にどんなトラブルがあるか分かりませんが、メールで文句(環境をきちんと明記したもの)を送っていただければ、もしかしたら解決策を提示できるかもしれません。

5 使用上の注意

先程表を書きましたが、逆に通常のコマンドが使えなくなっています。つまり、align 環境内で`\left(, \right)`は使えません。`\left[`ブラケット`\right]`、`\left\{`ブレース`\right\}`も同様です。これは、hoge.tex に例えば

```
\begin{align}\left(, \right), (, ), \normal(, \normal)\end{align}
```

と書くと、hoge_temp.tex には

```
\begin{align}\left\left(, \right\right), \left(, \right), (, )\end{align}
```

となって出力されているからです。`\left\left`を`\left`に置換する命令を書き足せば解決できますが、あえてこのコードは提示しません。この置換は align 環境内などという面倒くさいことは考えなくてよいので、簡単な命令で書けます。ご自身で(例えば改行を元に戻す置換の前に)付け足して下さい。

何度も言っていますが、hoge.bat は align 環境内ならなんでも置換してしまいます。例えば

```
...\begin{cases}...&...\{\displaystyle \int_0^a...}&...
```

のように、改行`\`の後にすぐ`{`が来ると bat は括弧と勘違いして`\left\{\displaystyle`...となってしまいます。すると、対応する右括弧がないよ!というエラーが出ます。ですので、`\displaystyle`の直前で改行するか、半角スペースを入れるかして下さい。

当然のことと言えば当然なのですが、`\normal(, \normal)`を使うと、普通の方法ではコンパイルできなくなります。普通の方法では`(,)`や`\left(, \right)`が使え、hoge.bat では`(,)`と`\normal(, \normal)`が使えます。つまり両方の方法で共通して使えるのは`(,)`だけとなります。ですので、`\normal(, \normal)`を使うことはお勧めしません。これらを使わなければ大丈夫なのですが、普段 hoge.bat を使っていて、それに慣れてしまうとうっかり TeX のソースファイルを人に渡してしまうかもしれません。そのファイルを渡された人は、確実に hoge.bat は持っていませんので、「なんだこれ、括弧をきちんと書けよ」と言われるでしょう。ひどいケースでは`(... \right.`と書いてあるときはエラーが出ます。ですので、人には hoge.bat が処理した後の TeX ファイル hoge_temp.tex を渡すよう注意して下さい。ただ、「おいおい、`f\left(x\right)`って、ここまで書く必要ないだろ」と言われても私は知りません。

もうすでにお気づきかと思いますが、「align 環境中」というのは正確ではありませんね。verbatim 環境内の`begin{align}`と`end{align}`の間の文字列に対しても置換が有効です。これは align 環境とは言えませんね。細かいことです。

改行を kaigyoyou_temp で置換し、最後にこれを改行に戻しています。つまり、最初から kaigyoyou_temp と書いてあると、これを改行に置き換えてしまいます。文章中ならまだいいですが、コメント中だと困りますね。この文字列は書かないでください。どうしてもという方は(いないと思いますが)、bat ファイル中の kaigyoyou_temp を他の文字列(例えば ichijiteki_na_kaigyoyou とか)にして下さい。

6 おわりに

この bat ファイルによって、数式を書くのが私にとってかなり楽になりました。しかし、単なる置換なのでおそらく T_EX の内部で処理できるでしょうね。どなたか

```
\replace[align]{()}{\left(}
```

を作って頂きたい。あるいは align 環境と括弧の定義だけ異なる、別の数式環境でもいいのですが。