

gfncmd.sty 第 0.1 版

@bd_gfngfn

2014 年 8 月 4 日

概要

gfncmd.sty は、頻出する記法を短縮して入力効率を上げることは勿論、加えてソースコードの保守性を高めることも意図して制作したスタイルファイルです。L^AT_EX はそのソフトウェアの設計上文書を単なる字形の排列として組むことを許容してしまいますが、字形としてだけではなく意味や構造を明示して書き、後で容易に改変できるようにしよう、というのがこのスタイルファイルの趣旨なのです。したがって、新しい機能を追加するというよりは既存の機能を“整頓する”役割を想定したスタイルファイルとなっています。

意味や構造を明示して書くことによるわかりやすい恩恵としては、記法に複数の流儀のあるものが後から容易に変更できることなどが挙げられます。このため、複数人で書いた原稿の記法の流儀を後で統一したり、他人の書いた原稿のコードに好きなオプションをつけてコンパイルすることで自分なりの読みやすい文書を組んだりすることが実現できるのです。

記法の流儀は、オプション指定用の制御綴により切り替えることができます。オプション指定用の制御綴を書いておくとその地点以降が指定に従いますが、保守性の観点から考えて文書の途中で切り替えることはできるだけ避け、プリアンブルに 1 箇所だけ記しておくのが理想的です。

コマンド

括弧類

演算順序の括弧

$\backslash displaystyle \backslash paren{\backslash lambda z.z \backslash paren{\backslash lambda x.x$	
入力：	$\backslash paren{\backslash paren{\backslash lambda y.yz} \backslash paren{\backslash lambda y.y}} \backslash paren{\backslash lambda u.u} \backslash paren{\backslash lambda t.t} \$$
$\backslash usenormalparen \text{ (標準)} \quad (\lambda z.z (\lambda x.x ((\lambda y.yz) (\lambda y.y))) (\lambda u.u)) (\lambda t.t)$	
出力：	$\backslash useexpandingparen \quad \left(\lambda z.z \left(\lambda x.x \left((\lambda y.yz) (\lambda y.y) \right) \right) (\lambda u.u) \right) (\lambda t.t)$

演算順序を指定する括弧です（関数の引数の括弧はこれとは区別します．後述）．2つのオプションがありますが，いずれも以下のように中身に合わせて大きさが変動します．

$$(a+b) \left(\frac{1}{2}c + \sqrt{d} \right)$$

`\useexpandingparen` オプションは，括弧の階層が深くなるにつれて外側の括弧の大きさが自動で大きくなるように設計されています． λ 項のように階層の複雑な式の表示では`\usenormalparen` オプションに比べ階層が見やすくなり，シンタックスハイライトの役目を期待できます．しかし標準の \LaTeX では括弧が細かい段階で調整されないのて，必ずしも整って見えるとは限りません．

関数引数の括弧

入力：	<code>\$\$</code>	
	<code>\usenormalfprn</code> （標準）	$y = f(x)$
出力：	<code>\usesmallfprn</code>	$y = f(x)$
	<code>\usescriptfprn</code>	$y = f(x)$

関数引数用の括弧です．一応`\usesmallfprn`，`\usescriptfprn`という引数を小さくするオプションを用意していますが，ほとんど使うことはないと思われます．

集合

自然数，整数，有理数，実数，複素数

入力：	<code>\$\$\setC, \setH, \setN, \setNp, \setP, \setQ, \setR, \setZ\$</code>	
	<code>\usemathbfset</code> （標準）	$\mathbf{C}, \mathbf{H}, \mathbf{N}, \mathbf{N}^+, \mathbf{P}, \mathbf{Q}, \mathbf{R}, \mathbf{Z}$
出力：	<code>\usebmset</code>	$\boldsymbol{C}, \boldsymbol{H}, \boldsymbol{N}, \boldsymbol{N}^+, \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{Z}$
	<code>\usemathbbset</code>	$\mathbb{C}, \mathbb{H}, \mathbb{N}, \mathbb{N}^+, \mathbb{P}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}$
併用：	<code>amssymb</code>	

\mathbf{N}^+ は広く使われているわけではありませんが，正整数を指すための簡潔で便利な記法のためコマンドとして導入しています．したがって \mathbf{N} は非負整数の意として扱うことを想定しています．なお，`\usemathbbset`で指定される黑板太字体に関しては「手書き字形なので印刷字体としては不適」という主張がよく見られ，あまり積極的に使用することは推奨しません．

行列 `\setMatrix{\#1}{\#2}{\#3}`

入力：	$\$A \in \text{\setMatrix{m}{n}{K}}\$$
	$\text{\usemathbfsetMatrix}$ (標準) $A \in \mathbf{M}_{m,n}(K)$
	\usebmsetMatrix $A \in \mathbf{M}_{m,n}(K)$
出力：	$\text{\useromansetMatrix}$ $A \in M_{m,n}(K)$
	$\text{\useitalicsetMatrix}$ $A \in M_{m,n}(K)$
	$\text{\usetimessetMatrix}$ $A \in K^{m \times n}$

冪集合 $\text{\Powerset}{\#1}$

入力：	$\$\text{\Powerset}{S}\$$
	\useromanPowerset (標準) $\text{Pow } S$
出力：	$\text{\usebinaryPowerset}$ 2^S
	\usefrakPowerset $\mathfrak{P}S$
併用：	<code>amsmath</code> , <code>amssymb</code>

微分

微分記号 \ordd

入力：	$\$\text{\ordd } U = T \text{\ordd } S - P \text{\ordd } V\$$
	\useromanordd (標準) $dU = TdS - PdV$
出力：	\useitalicordd $dU = TdS - PdV$

\useromanordd はローマン体で、 \useitalicordd はイタリック体で微分記号を出力します。特にオプションを指定しなかった場合は \useromanordd となります。以降のコマンドも同様。

1 階微分 $\text{\dif}{\#1}{\#2}$

入力：	$\$\text{\displaystyle } \text{\dif}{y}{x} + ay = 0\$$
	\useromanordd (標準) $\frac{dy}{dx} + ay = 0$
出力：	\useitalicordd $\frac{dy}{dx} + ay = 0$

2 階微分 $\text{\ddif}{\#1}{\#2}$

入力：	$\displaystyle \ddif{x}{t} + \omega^2 x = 0$
出力：	$\text{\useromanorrd} \quad \frac{d^2 x}{dt^2} + \omega^2 x = 0$ $\text{\useitalicorrd} \quad \frac{d^2 x}{dt^2} + \omega^2 x = 0$

多階微分 $\text{\ndif}{\#1}{\#2}{\#3}$

入力：	$\displaystyle \sum_{i=0}^{\infty} \text{\ndif}{i}{y}{x}$
出力：	$\text{\useromanorrd} \quad \sum_{i=0}^{\infty} a_i \frac{d^i y}{dx^i} = 0$ $\text{\useitalicorrd} \quad \sum_{i=0}^{\infty} a_i \frac{d^i y}{dx^i} = 0$

1 階偏微分 $\text{\pdif}{\#1}{\#2}$

入力：	$\displaystyle \pdif{}{r} = \pdif{x}{r} \pdif{}{x} + \pdif{y}{r} \pdif{}{y} + \pdif{z}{r} \pdif{}{z}$
出力：	$\frac{\partial}{\partial r} = \frac{\partial x}{\partial r} \frac{\partial}{\partial x} + \frac{\partial y}{\partial r} \frac{\partial}{\partial y} + \frac{\partial z}{\partial r} \frac{\partial}{\partial z}$

2 階偏微分 $\text{\pddif}{\#1}{\#2}$

入力：	$\displaystyle \paren{-\frac{\hbar^2}{2m} \pddif{}{x} + \pddif{}{y} + \pddif{}{z} + V} \psi = i \hbar \pdif{\psi}{t}$
出力：	$\left(-\frac{\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) + V \right) \psi = i \hbar \frac{\partial \psi}{\partial t}$

文字・記号

イプシロン \epsilon

`gfn cmd.sty` を読み込むと、自動的に以下のような設定が適用されます。

入力：	\epsilon
出力：	$\text{\usesymbolepsilon} \quad \epsilon$ $\text{\usescriptepsilon} \quad \varepsilon$

同字種異字体を混在させることを極力避けるためにコマンド \epsilon を再定義しています。 \varepsilon

は未定義化されており，記述して実行するとエラーとなります．

ファイル `\phi`

`\epsilon` と同様に，`gfncmd.sty` を読み込むと，自動的に以下のような設定が適用されます．

入力：	<code>\phi</code>	
出力：	<code>\usesymbolphi</code> (標準)	ϕ
	<code>\usescriptphi</code>	φ

すなわち，同字種異字体を混在させることを極力避けるためにコマンド `\phi` を再定義しています．例えば，量子力学で ϕ を波動関数， φ を球面座標の第 3 変数として表記し別字種として区別することがありますが，それは原則として望ましくない，と考えるのがこのスタイルファイルの流儀です．`\varphi` は未定義化されており，記述して実行するとエラーとなります．どうしても ϕ と φ を別字種として区別のようにしたい場合は `\makephisdifferent` と書いておくと同様の $\mathrm{T}_{\mathrm{E}}\mathrm{X}$ と同様に `\phi` と `\varphi` を区別して扱います．

入力：	<code>\phi, \varphi</code>	
出力：	ϕ, φ	

再び `gfncmd.sty` の仕様に戻す場合は `\makephisssame` と書くことで可能です．

転置記号 `\transposed`

入力：	<code>\transposed{\bm{x}}Q\bm{x} + 2 \transposed{\bm{p}} \bm{x}</code>	
	<code>\usetoptransposed</code> (標準)	$\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x} + 2\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x}$
出力：	<code>\useTtransposed</code>	$\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x} + 2\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x}$
	<code>\uselefttransposed</code>	${}^t\boldsymbol{x}Q\boldsymbol{x} + 2{}^t\boldsymbol{p}\boldsymbol{x}$

転置行列を表す記号です．`\transposed` は少々長いので，略した `\trps` もまったく同一の制御綴として使えます．`\uselefttransposed` オプションの場合， $({}^tA)^{-1} = {}^t(A^{-1})$ ですから ${}^tA^{-1}$ のように交換可能な演算の順序を気にせず書いて便利だと思いがちですが，これをそのまま書いたあとに別のオプションに変更すると $A^{\mathrm{T}-1}$ ， $A^{\mathrm{T}-1}$ となってしまっていて醜悪であり，修正する必要が出てしまいます．括弧は

半順序 `\pord`

入力：	$\$x \ \backslash\join \ \backslash\paren{y \ \backslash\meet z} \ \backslash\pord \ \backslash\paren{x \ \backslash\join y} \ \backslash\meet \ \backslash\paren{x \ \backslash\join z}\$$	
出力：	$\backslash\usepreccurlyeq$ (標準)	$x \vee (y \wedge z) \preccurlyeq (x \vee y) \wedge (x \vee z)$
	$\backslash\usepreceq$	$x \vee (y \wedge z) \preceq (x \vee y) \wedge (x \vee z)$
	$\backslash\useleq$	$x \vee (y \wedge z) \leq (x \vee y) \wedge (x \vee z)$
	$\backslash\useleqslant$	$x \vee (y \wedge z) \leqslant (x \vee y) \wedge (x \vee z)$
併用：	amssymb	