The `ShorTeX` package

Trevor Campbell, Jonathan Huggins, and Jeff Negrea

Updated March 21, 2024

Abstract

The purpose of the ShorTeX (meta)package is to make the process of typesetting typical mathematical documents in LaTeX more efficient, and the resulting code easier to read. It achieves this by (1) providing an extensive, internally consistent, and easy to learn set of macro shorthands and custom commands, and (2) incorporating a set of packages that are dedicated to reducing manual coding effort.

Contents

1 Usage and package options

Put a copy of `shortex.sty` in the folder alongside your other document files, and include ShorTeX in the document by adding `\usepackage{shortex}` to the preamble. **Do not install ShorTeX system-wide;** this package has not yet reached a stable version 1.0, and we are updating things regularly without any guarantee of backwards compatibility. **You must compile your document 4 times when using ShorTeX** to ensure that equation numbers and references update properly.

ShorTeX will include and configure many common packages for you (e.g., `graphicx`, `subcaption`, `hyperref`, `algorithm`, `algpseudocode`, `amsmath`, among others), so you do not need to explicitly include and set these up yourself. If you are writing a document that must use a specific style file (e.g., for a conference or journal) that itself includes some of these packages, we recommend editing those style files to remove the package imports.

The ShorTeX package has a few options:

`manualnumbering` Do not include `autonum.sty`. This disables automatic equation numbering.

`blackhypersetup` Switch hyperlinks, citations, references, etc. to be typeset in black font. The default is dark blue.

`draft` Enable `graphicx` draft mode (with placeholder figures).

`minimal` Disable common font style/accent combinations (see Section 5 for details).

2 Environments

LaTeX documents often include a lot of verbose code related to creating environments (`\begin{blah}...\end{blah}`). ShorTeX provides a set of shortened macros for common environments. Each shortened begin and end command starts with `\b...` and `\e...`, respectively. Note that all theorem-like environments (theorem, lemma, proposition, etc.) are numbered by default; unnumbered versions can be obtained by appending a `u`. For example, `\bthmu...\ethmu` creates an unnumbered theorem environment, while `\blemu...\elemu` creates an unnumbered lemma environment.

Environment	Syntax
abstract	`\babs...\eabs`
itemize	`\bitem...\eitem`
enumerate	`\benum...\eenum`
description	`\bdesc...\edesc`
algorithm	`\balg...\ealg`
algorithmic	`\balgc...\ealgc`
table	`\btab...\etab`
subtable	`\bsubtab...\esubtab`
tabular	`\btabr...\etabr`
figure	`\bfig...\efig`
figure*	`\bfigs...\efigs`
subfigure	`\bsubfig...\esubfig`
center	`\bcent...\ecent`
align	`\[...\]`
inline math	`$...$`
Note: These are numbered theorem-like environments. For unnumbered, append a `u`: e.g., `bthmu...ethmu`.	
theorem	`\bthm...\ethm`
lemma	`\blem...\elem`
proposition	`\bprop...\eprop`
corollary	`\bcor...\ecor`
conjecture	`\bconj...\econj`
definition	`\bdef...\edef`
assumption	`\bassump...\eassump`
example	`\bexa...\eexa`
remark	`\brmk...\ermk`
fact	`\bfact...\efact`
exercise	`\bexer...\eexer`
proof	`\bprf...\eprf`
proofof	`\bprfof{\cref{theorem_label}}...\eprfof`
matrix	`\bmat...\emat`
bmatrix	`\bbmat...\ebmat`
pmatrix	`\bpmat...\epmat`

3 Delimiters

Mathematics in LaTeX often includes quite a few delimiters (parentheses, brackets, curly brackets, etc.). A very common usage of these involves the `\left...\right` commands for automatic sizing. One can also use `\bigl...\bigr`, `\Bigl...\Bigr`, `\biggl...\biggr`, `\Biggl...\Biggr` to control sizing manually. ShorTeX creates shorthands for these.

Description	Syntax
automatic	`\lt...\rt`
big	`\lb...\rb`
Big	`\lB...\rB`
bigg	`\lbg...\rbg`
Bigg	`\lBg...\rBg`

These can be applied to all the usual delimiter characters. The following tables demonstrate usage for automatically sized delimiters.

Description	Example	Text style	Display style
parentheses	`\lt(\frac{x}{y}\rt)`	$\left(\frac{x}{y}\right)$	$\left(\frac{x}{y}\right)$
curly brackets	`\lt\{\frac{x}{y}\rt\}`	$\left\{\frac{x}{y}\right\}$	$\left\{\frac{x}{y}\right\}$
square brackets	`\lt[frac{x}{y}\rt]`	$\left[\frac{x}{y}\right]$	$\left[\frac{x}{y}\right]$
pipes	`\lt|frac{x}{y}\rt|`	$\left\lvert\frac{x}{y}\right\rvert$	$\left\lvert\frac{x}{y}\right\rvert$
double pipes	`\lt\\|frac{x}{y}\rt\\|`	$\left\lVert\frac{x}{y}\right\rVert$	$\left\lVert\frac{x}{y}\right\rVert$
angle brackets	`\lt<frac{x}{y}\rt>`	$\left\langle\frac{x}{y}\right\rangle$	$\left\langle\frac{x}{y}\right\rangle$

4 Greek characters and variants

ShorTeX defines shorthands for Greek characters and variants. The syntax is identical to standard LaTeX for letters with three or fewer characters, and is reduced to three characters for those with more than three. Variants are obtained by preceding the usual command with `\v...`.

Letter	Syntax	Symbol	Variant Syntax	Variant Symbol
alpha	`\apa,A`	α, A		
beta	`\bta,B`	β, B		
gamma	`\gma,\Gma`	γ, Γ		
delta	`\dta,\Dta`	δ, Δ		
epsilon	`\eps,E`	ϵ, E	`\veps,E`	ε
zeta	`\zta,Z`	ζ, Z		
eta	`\eta,H`	η, H		
theta	`\tta,\Tta`	θ, Θ	`\vtta`	ϑ
iota	`\ita,I`	ι, I		
kappa	`\kpa,K`	κ, K	`\vkpa`	$\varkappa$
lambda	`\lda,\Lda`	λ, Λ		
mu	`\mu,M`	μ, M		
nu	`\nu,N`	ν, N		
xi	`\xi,\Xi`	ξ, Ξ		
omicron	`o,O`	o, O		
pi	`\pi,\Pi`	π, Π	`\vpi`	ϖ
rho	`\rho,P`	ρ, P	`\vrho`	ϱ
sigma	`\sga,\Sga`	σ, Σ	`\vsga`	ς
tau	`\tau,T`	τ, T		
upsilon	`\ups,\Ups`	υ, Υ		
phi	`\phi,\Phi`	ϕ, Φ	`\vphi`	φ
chi	`\chi,X`	χ, X		
psi	`\psi,\Psi`	ψ, Ψ		
omega	`\oga,\Oga`	ω, Ω		

5 Font styles and accents

Applying accents (e.g., hats $\widehat{a}$, tildes $\widetilde{a}$, bars $\bar{a}$) and changing fonts (e.g., double-stroke $\mathbb{A}$, caligraphic $\mathcal{A}$, and bold $\boldsymbol{A}$) is quite cumbersome in standard LaTeX. For example, the code to make a tilde caligraphic A, $\widetilde{\mathcal{A}}$ is `\widetilde{\mathcal{A}}`. By itself that code is not too bad, but many such characters in a large mathematical expression results in unreadable code.

ShorTeX defines an efficient syntax for changing fonts and applying accents to characters. The syntax takes the form `\s[modifiers]character`, where `modifiers` is a set of single characters that represent font/accent modifications to `character`. For example, the code for tilde caligraphic A is `\s[tc]A` where `t` represents "tilde," `c` represents "caligraphic," and `A` is the character to typeset.

Note: modifiers are applied in the reverse of the order in which they appear; the modifier furthest to the right is applied first. This matches the order that the corresponding commands would appear in TeX code.

Style/Accent	Modifier	Example	Typeset Example
caligraphic (`mathcal`)	`c`	`\s[c]A`	$\mathcal{A}$
bold (`mathbf`)	`k`	`\s[k]A`	$\boldsymbol{A}$
doublestroke (`mathbb`)	`d`	`\s[d]A`	$\mathbb{A}$
hat (`widehat`)	`h`	`\s[h]A`	$\widehat{A}$
tilde (`widetilde`)	`t`	`\s[t]A`	$\widetilde{A}$
bar (`widebar`)	`b`	`\s[b]A`	$\bar{A}$

These style modifiers can be combined; the underlying code is flexible enough that it will happily produce a wide variety of combinations, including those that aren't very sensible.

Style/Accent	Modifier	Example	Typeset Example
caligraphic tilde	`ct`	`\s[ct]A`	$\widetilde{\mathcal{A}}$
bold hat	`kh`	`\s[kh]A`	$\widehat{\boldsymbol{A}}$
hat tilde	`ht`	`\s[ht]A`	$\widetilde{\widehat{A}}$
tilde hat	`th`	`\s[th]A`	$\widehat{\widetilde{A}}$

We can avoid typing `[]` for commonly used patterns by parsing the font style string in advance. For example, if we use "bold hat" symbols frequently, we might want to use commands like `\skh...` instead of `\s[kh]...`. We can accomplish this using the `\parsefontstylestrings` command, with syntax

```
\parsefontstylestrings{{<fstr1>}{<fstr2}...}{<alphabet>}
```

For example, to define "bold hat" and "caligraphic hat" styles for the characters A, B, C, and D, we would use the command

```
\parsefontstylestrings{{kh}{ch}}{ABCD}
```