

The *alphabet* package

Günter Milde

February 5, 2016

Abstract

The *alphabet* package makes the standard macros for Greek letters in mathematical mode also available in text mode. This way, you can input Greek letters “by name” everywhere in the document. The mode determines whether the characters are taken from the text or math font.

With 8-bit TeX and *greek-inputenc*, literal Unicode characters can also be used in mathematical mode.

1 Requirements

The *alphabet* package depends on *textalpha* (both are part of *greek-fontenc*).

It can be used under 8-bit TeX as well as XeTeX/LuaTeX (this document is typeset with 8-bit TeX). Depending on the converter and fonts, different [limitations](#) apply.

The package conflicts with *unicode-math*.

2 Usage

Load this package in the preamble of your document (after font and math setup) with

```
\usepackage{alphabet}
```

or, if you want to merge symbol variants, with the [normalize-symbols](#) option:

```
\usepackage[normalize-symbols]{alphabet}
```

See the source of this document [alphabet-doc.tex](#) for a setup and usage example.

Now you can write a single Greek symbol (like Ψ or μ) or a $\lambda\omicron\gamma\omicron\varsigma$ in non-Greek text as well as ISO-conforming formulas with upright symbols for constants like $A = \pi r^2$ (instead of $A = \pi r^2$).¹

¹The [isomath documentation](#) describes more alternatives for upright Greek symbols in math mode.

Just like Latin letters, the Greek counterparts are by default italic in math mode² and upright in text:

Text: L Γ l γ, mathematics: *L Γ l γ*

2.1 option `normalize-symbols`

The `normalize-symbols` option is passed to the [alphabet](#) package.³ It merges “letters” and “symbol” variants of some Greek letters to the “letter” character:

This text uses both variants for beta ($\beta|\beta$), theta ($\vartheta|\vartheta$), phi ($\varphi|\varphi$), pi ($\pi|\pi$), kappa ($\kappa|\kappa$), rho ($\rho|\rho$), Theta ($\Theta|\Theta$), and epsilon ($\varepsilon|\varepsilon$) in the LaTeX source.

Without this option, the symbol variant characters cannot be used in text, because they are not supported by 8-bit Greek fonts (LGR encoding).

The `normalize-symbols` option is ignored, if you compile the document with XeTeX or LuaTeX using Unicode fonts. (This document is compiled using 8-bit TeX.)

Attention: Be careful in cases where the distinction between the symbol variants may be important (e.g. in a mathematical or scientific context). Use XeTeX/LuaTeX with Unicode fonts or the respective characters in mathematical mode (e.g. π vs. ϖ).

2.2 symbol variants

Mathematical notation uses variant shapes of some Greek letters as additional symbols. The variations have no syntactic meaning in Greek text and text fonts may use the variant shapes in place of the “regular” ones as a stylistic choice.

Unicode defines separate code points for the symbol variants. TeX supports some of the variant shape symbols in mathematical mode, but its concept of “standard” vs. “variant” symbols differs from the distinction between “GREEK LETTER ...” vs. “GREEK ... SYMBOL” in the Unicode standard. See [greek-euenc-doc](#).

The [alphabet](#) package defines generic macros for these variants that are short forms of the set defined in `greek-euenc.def`:

`\<name>` selects the Unicode GREEK LETTER ... variant,
`\<name>symbol` selects the Unicode GREEK ... SYMBOL variant,
`\var<name>` selects the variant shape according to TeX’ mathematical mode

See Table 1 for the full list.

²Capital Greek letters are upright in TeX unless a package selects the “ISO” math-style. See the [isomath documentation](#) for a detailed discussion of math-styles.

³The `normalize-symbols` option was added in version 0.13 (2015-08-03). Unicode input of the symbol variants requires at least version 1.6 (2015-08-05) of [greek-inputenc](#).

3 Limitations

With 8-bit TeX, the limitations described in the [textalpha documentation](#) apply. See also the tests in section [8 bit limitations](#).

With XeTeX/LuaTeX and Unicode fonts, literal Unicode characters cannot be used in formulas (the log file reports missing characters) This is a generic TeX limitation which *alphabetalpha* overcomes if used under 8-bit TeX. Under XeTeX/LuaTeX it may be lifted using the *unicode-math* package, but unfortunately, *alphabetalpha* conflicts with *unicode-math*.

4 Tests and examples

4.1 Greek alphabet

Greek letters via generic “name” macros without language/font-encoding switch:

A B Γ Δ E Z H Θ I K Λ M N Ξ O Π P Σ T Υ Φ X Ψ Ω
α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω
ϕ F ϕ ⁴ ι ϑ ϱ ϒ Ϝ ϝ Ϟ ϟ Ϡ ϡ Ϣ ϣ Ϥ ϥ Ϧ ϧ Ϩ ϩ Ϫ ϫ Ϭ ϭ Ϯ ϯ ϰ ϱ ϲ ϳ ϴ ϵ ϶ Ϸ ϸ Ϲ Ϻ ϻ ϼ Ͻ Ͼ Ͽ Ͽ

Greek letters via Unicode input without language/font-encoding switch:

A B Γ Δ E Z H Θ I K Λ M N Ξ O Π P Σ T Υ Φ X Ψ Ω
α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω
ϕ F ϕ ι ϑ ϱ ϒ Ϝ ϝ Ϟ ϟ Ϡ ϡ Ϣ ϣ Ϥ ϥ Ϧ ϧ Ϩ ϩ Ϫ ϫ Ϭ ϭ Ϯ ϯ ϰ ϱ ϲ ϳ ϴ ϵ ϶ Ϸ ϸ Ϲ Ϻ ϻ ϼ Ͻ Ͼ Ͽ Ͽ

4.2 Diacritics

Accent macros are set up for use with the generic macros by definition of “TextComposite” commands.

Diacritics (except the dialytika) should be placed before capital letters and dropped with MakeUppercase:

ά έ ί ή ῶ ῶ ῶ
Ά Έ Ί Η Ὠ Ὠ Ὠ
Α Ε Ϊ Η Ο Υ Ω

4.3 ‘Ελληνικά (‘Ελληνικά) in PDF strings

With the *alphabetalpha* package, you get Greek letters in both, the document body and PDF metadata generated by `hyperref` if the input uses Unicode literals or macros. Wrapping in `\ensuregreek` ensures the right placement of the accents and breathings (before, not above capital letters). With LICR input (accent macros + symbol macros), non-standard diacritics are missing in the PDF

⁴There is no separate Unicode code point for a stigma variant symbol, `\varstigma` is not defined with Xe/LuaTeX and similar to `\stigma` in some fonts.

data, as hyperref’s PU encoding currently does not support polytonic Greek. (Here, the dasia is dropped at the start of the word in parentheses in the PDF toc. The warning “Glyph not defined in PU encoding, removing ‘\<’ on input line 145.” is written to the log.)

Warnings like

```
Token not allowed in a PDF string (Unicode):
removing ‘\textbackslash TextOrMath ’ on input line 145.
```

can be ignored or avoided with the preamble code:

```
\pdfstringdefDisableCommands{\let\TextOrMath\@firstoftwo}
```

4.4 Greek in math $\Gamma = \sin \alpha / \cos \beta$

In the main document, Greek in math mode should work as usual:

$$\Gamma = \frac{\sin \alpha}{\cos \beta}.$$

Greek letters and symbols in math input as macro (there are no math macros for Greek letters wich exist with similar shape in the Latin alphabet):

```
Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
α β γ δ ε ζ η θ ι κ λ μ ν ξ π ρ σ τ υ φ χ ψ ω
ϑ ϕ ϗ ϝ Ϟ ϟ
```

PDF strings do not know math mode. The content of a formula or equation is evaluated in text mode with non-valid commands discarded (and warnings written to the log). This works resonably well for simple formulas (but not, e.g., for super-/subscripts). With the *alphabet* package, it works also for Greek letters.

4.5 Greek Unicode characters in math (only under 8-bit TeX)

With the `utf8` option of *inputenc* and *greek-inputenc*, literal Greek Unicode characters are supported also in mathematical mode:

$$\Gamma = \frac{\sin \alpha}{\cos \beta}.$$

Greek letters and symbols in math input as Unicode literals:

```
Γ Δ Θ Λ Ξ Π Σ Υ Φ Ψ Ω
α β γ δ ε ζ η θ ι κ λ μ ν ξ π ρ σ τ υ φ χ ψ ω
ϑ ϕ ϗ ϝ Ϟ ϟ
```

The “normal” vs. “variant” shape of phi and epsilon is inverted when using traditional makros or Unicode letter vs. symbol characters respectively. This is to keep backwards compatibility of the math macros as well as consistent input-output mapping for Unicode in text and math. It corresponds to the behaviour of *unicode-math* with the default option `vargreek-shape=TeX`.

4.6 8-bit limitations

These limitations are lifted, if the document is compiled with XeTeX/LuaTeX.

- Composition of diacritics (like `\>\')` fails: $\acute{\alpha} \acute{\epsilon} \grave{\iota} \acute{\eta} \acute{\omega} \acute{\nu} \acute{\omega}$
Simple diacritics and long names (like `\accdasiaoxia`) work in any font encoding, however they do not select precomposed characters (the difference becomes obvious if you drag-and-drop text from the PDF version of this document): $\acute{\alpha} \acute{\alpha}$ ($\Lambda\Gamma\text{P}$) vs. $\acute{\alpha}$ (T1)
- MakeUppercase fails with composite diacritics in other font encodings.
- There is no kerning between Greek letters, if the font encoding does not support Greek: compare $\text{A}\Upsilon\text{A}$ ($\Lambda\Gamma\text{P}$) to $\text{A}\Upsilon\text{A}$ (T1). Because of this (and for proper hyphenation), use of the Babel package and correct language setting is recommended for Greek quotes.

The `\ensuregreek` macro ensures that the argument is typeset with a font encoding supporting Greek. This keeps kerning (if the kerning pair is inside the argument, $\text{A}\Upsilon\text{A}$), and allows combining of accent macros where pre-composed characters are selected ($\acute{\alpha}$).

macro	text	math
<code>\pi</code>	π	π
<code>\varpi</code>	π	ϖ
<code>\pisymbol</code>	π	ϖ
<code>\rho</code>	ρ	ρ
<code>\varrho</code>	ρ	ϱ
<code>\rhosymbol</code>	ρ	ϱ
<code>\theta</code>	ϑ	θ
<code>\vartheta</code>	ϑ	ϑ
<code>\thetasymbol</code>	ϑ	ϑ
<code>\epsilon</code>	ε	ϵ
<code>\varepsilon</code>	ε	ε
<code>\epsilonsymbol</code>	ε	ϵ
<code>\phi</code>	φ	ϕ
<code>\varphi</code>	φ	φ
<code>\phisymbol</code>	φ	ϕ
<code>\beta</code>	β	β
<code>\varbeta</code>	β	\varnothing
<code>\betasymbol</code>	β	\varnothing
<code>\kappa</code>	κ	κ
<code>\varkappa</code>	κ	\varnothing
<code>\kappasymbol</code>	κ	\varnothing
<code>\Theta</code>	Θ	Θ
<code>\varTheta</code>		Θ
<code>\Thetasymbol</code>	Θ	

Table 1: Macros for Greek symbol variants (\varnothing = symbol missing). With 8-bit TeX and the `normalize-symbols` option, the output for both variants in text mode is the same (8-bit Greek text fonts contain only one symbol variant).