

The *alphabet* package

Günter Milde

June 4, 2013

With the *alphabet* package, you can write Greek letters “by name” or as literal Unicode characters in both, “text” and “math” mode. The mode determines whether the letters are taken from the text or math font. Just like Latin letters, the Greek counterparts are by default italic in math mode (capital letters upright without *fixmath* or *isomath*) and upright in text:

Text: $\Lambda \Gamma \iota \gamma$, emphasized text: $L \Gamma l \gamma$, mathematics: $L \Gamma l \gamma$

This makes it easy to 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 $A = \pi r^2$.

Tests and examples

Greek alphabet

Greek letters via Latin transscription in LGR font encoding:

$\Lambda \ B \ \Gamma \ \Delta \ E \ Z \ H \ \Theta \ I \ K \ \Lambda \ M \ N \ \Xi \ O \ \Pi \ P \ \Sigma \ T \ \Upsilon \ \Phi \ X \ \Psi \ \Omega$
 $\alpha \ \beta \ \gamma \ \delta \ \epsilon \ \zeta \ \eta \ \vartheta \ \iota \ \kappa \ \lambda \ \mu \ \nu \ \xi \ \omicron \ \pi \ \rho \ \varsigma \ \tau \ \upsilon \ \varphi \ \chi \ \psi \ \omega$

Greek letters via default macros in LGR font encoding:

$\Lambda \ B \ \Gamma \ \Delta \ E \ Z \ H \ \Theta \ I \ K \ \Lambda \ M \ N \ \Xi \ O \ \Pi \ P \ \Sigma \ T \ \Upsilon \ \Phi \ X \ \Psi \ \Omega$
 $\alpha \ \beta \ \gamma \ \delta \ \epsilon \ \zeta \ \eta \ \vartheta \ \iota \ \kappa \ \lambda \ \mu \ \nu \ \xi \ \omicron \ \pi \ \rho \ \sigma \ \varsigma \ \tau \ \upsilon \ \varphi \ \chi \ \psi \ \omega$
 $\text{\textsf{F}} \text{\textsf{F}} \ \text{\textsf{f}} \ \text{\textsf{f}} \ \text{\textsf{I}} \ \text{\textsf{I}} \ \text{\textsf{O}} \ \text{\textsf{O}} \ \text{\textsf{T}} \ \text{\textsf{T}} \ \text{\textsf{X}} \ \text{\textsf{X}} \ \text{\textsf{Y}}$

Greek letters via default macros in other font encoding (here T1):

$\Lambda \ B \ \Gamma \ \Delta \ E \ Z \ H \ \Theta \ I \ K \ \Lambda \ M \ N \ \Xi \ O \ \Pi \ P \ \Sigma \ T \ \Upsilon \ \Phi \ X \ \Psi \ \Omega$
 $\alpha \ \beta \ \gamma \ \delta \ \epsilon \ \zeta \ \eta \ \vartheta \ \iota \ \kappa \ \lambda \ \mu \ \nu \ \xi \ \omicron \ \pi \ \rho \ \sigma \ \varsigma \ \tau \ \upsilon \ \varphi \ \chi \ \psi \ \omega$
 $\text{\textsf{F}} \text{\textsf{F}} \ \text{\textsf{f}} \ \text{\textsf{f}} \ \text{\textsf{I}} \ \text{\textsf{I}} \ \text{\textsf{O}} \ \text{\textsf{O}} \ \text{\textsf{T}} \ \text{\textsf{T}} \ \text{\textsf{X}} \ \text{\textsf{X}} \ \text{\textsf{Y}}$

Greek letters via Unicode input in T1 font encoding:

Α Β Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω
 α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ ς τ υ φ χ ψ ω
 Ϝ ϝ Ϟ ϟ Ϡ ϡ Ϣ ϣ Ϥ ϥ Ϧ ϧ Ϩ ϩ

Greek letters in math (with *fixmath*): (There are no math macros for Greek letters which exist with similar shape in the Latin alphabet.)

$\Gamma \Delta \Theta \Lambda \Xi \Pi \Sigma \Upsilon \Phi \Psi \Omega$
 $\alpha \beta \gamma \delta \epsilon \zeta \eta \theta \iota \kappa \lambda \mu \nu \xi \pi \rho \sigma \varsigma \tau \upsilon \phi \chi \psi \omega \varphi$

Unicode input in math mode:

$\Gamma \Delta \Theta \Lambda \Xi \Pi \Sigma \Upsilon \Phi \Psi \Omega$
 $\alpha \beta \gamma \delta \epsilon \zeta \eta \theta \iota \kappa \lambda \mu \nu \xi \pi \rho \sigma \varsigma \tau \upsilon \phi \chi \psi \omega \varphi$

Έλληνικά (Έλληνικά) in PDF strings

With the `alphabet` 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 data, as `hyperref`'s PU encoding (currently) does not support polytonic Greek (dasia dropped at the start of the heading in the PDF toc).

The generic macros result in warnings like

Package `hyperref` Warning: Token not allowed in a PDF string (Unicode):
(`hyperref`) removing ‘`\TextOrMath`’ on input line 109.

Package `hyperref` Warning: Token not allowed in a PDF string (Unicode):
(`hyperref`) removing ‘`\mathlambda`’ on input line 109.

that can safely be ignored.

Greek in math $\sin^2 \alpha$

PDF strings do not know math mode. The content of a formula or equation is evaluated in text mode with non-valid commands discarded. This works for simple formulas (but not for super-/subscripts). With the `alphabet` package, it works also for Greek letters (with the abovementioned warnings from `hyperref`).

In the main document, Greek continues to work as usual:

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

Literal Unicode characters in mathematics are currently not supported.

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:

$\acute{\alpha}$ $\acute{\epsilon}$ $\acute{\iota}$ $\grave{\eta}$ $\acute{\omicron}$ $\acute{\upsilon}$ $\acute{\omega}$
 $\grave{\alpha}$ $\grave{\epsilon}$ $\grave{\iota}$ $\grave{\eta}$ $\grave{\omicron}$ $\grave{\upsilon}$ $\grave{\omega}$
 $\acute{\alpha}$ $\acute{\epsilon}$ $\acute{\iota}$ $\acute{\eta}$ $\acute{\omicron}$ $\acute{\upsilon}$ $\acute{\omega}$

Limitations

If the current font encoding is not LGR, the same limitation like with *textalpha* apply:

- Composition of diacritics (like $\backslash>\backslash'$) fails:

$\acute{\alpha}$ $\acute{\epsilon}$ $\acute{\iota}$ $\acute{\eta}$ $\acute{\omicron}$ $\acute{\upsilon}$ $\acute{\omega}$

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): $\acute{\alpha}$ $\acute{\epsilon}$ $\acute{\iota}$ $\acute{\eta}$ $\acute{\omicron}$ $\acute{\upsilon}$ $\acute{\omega}$ (LGR) vs. $\acute{\alpha}$ (T1)

- MakeUppercase fails with composite diacritics in other font encodings.
- There is no kerning between Greek letters, if the font encoding is not LGR: compare $\acute{\alpha}\acute{\epsilon}$ (LGR) to $\acute{\alpha}\acute{\epsilon}$ (T1). Because of this (and for proper hyphenation), use of the Babel package and correct language setting is recommended for Greek quotes.

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