Where's the Greek Shift Key?

S. A. Fulling
Mathematics Department, Texas A&M University, College Station, TX 77843
409-845-2237. Internet: fulling@sarastro.tamu.edu Bitnet: saf8613@tamstar

Abstract

Experienced typists of mathematical formulas soon tire of typing out the names of Greek letters; they adopt short macros, such as `\za` for `\alpha`. A standard correspondence between Greek and Latin letters is proposed, in which phonetic resemblance is given precedence over typographical. The resulting macros enable any Greek letter to be typed in three keystrokes (or two, if `\z` is assigned to a function key).

The title is one of the first questions I asked when I began using TeX in 1985. I couldn't believe that one had to write out the names of the Greek letters in mathematical formulas. Later I learned to appreciate TeX's design, where control sequences are given self-explanatory, comprehensible names, and it is the user's responsibility to speed things up by giving shorter names to the most frequently used ones.

Ideally, our keyboards would have a Greek shift key, so that holding down that key and typing `a` would enter `\alpha`. Lacking that, we could just `\define` `\a` as `\alpha`, and similarly for other letters — except that there are already some single-letter control sequences in `plain` TeX (e.g., `\b`, `\d`, `\H`, `\i`, ...). A check of *The TeXbook*'s index reveals, however, that there are no *two-letter* control sequences beginning with `\z`. Thus we can `\define` `\za` as `\alpha`, and so on, enabling any Greek letter to be typed in three keystrokes as soon as a correspondence between Greek and Latin letters is established.

In fact, the keystrokes can be reduced to two if `\z` can be assigned to a single key. This is even better than a shift key! One situation where this can be done is an IBM PC or compatible running a keyboard macro program such as SuperKey. My solution (which will not fit everyone's typing style) is to program the `5` key in the numeric keypad to enter `\z`. (This is the only numeric key that does not already have a cursor or editing function assigned to it.)

All this discussion is preliminary to the main problem: There is no well-defined correspondence between Greek and Latin letters. Indeed, there are 26 Latin letters and only 24 Greek ones. Clearly, α should be assigned to `a` and β to `b`, but should γ be represented by `c` or by `g`? I would choose the latter; but then what should we do with letters like θ and ψ, which have no Latin equivalents?

Looking at the *Symbol 12* type element ("math golf ball") of my old IBM Selectric typewriter, I find α and β treated as expected, but then γ is assigned to `q`, δ to `w`, θ to `r`, ϕ to `d`, and so on. I have never taken a course in Greek, but I think that every reasonably well educated person can see that something is wrong here.[1] Let us try to construct a correspondence that is less arbitrary.

Of course, ultimately such a correspondence *is* arbitrary, in the sense that some design decisions are necessary. My first decision has already been hinted at: Instead of simply matching up the alphabets in order ($\gamma \mapsto$ `c`), we should try to assign each Greek letter to the most closely corresponding Latin letter ($\gamma \mapsto$ `g`). With this understanding, the following Greek letters have obvious Latin equivalents: α, β, γ, δ, ϵ, ζ, ι, κ, λ, μ, ν, o, σ, τ, υ, ϕ. This is 16 of the 24.

Beyond this point, however, there is room for controversy. There is a temptation to assign ρ to `p`, since the letters look alike. But I prefer to give phonetic resemblance precedence over typographical resemblance; ρ is `r`, and `p` is π. For the same reason, `x` represents ξ, not χ. After all, χ is nicely handled by `q`, a phonetically similar and otherwise useless letter. However, typographical resemblance takes over for η ($\mapsto$ `h`) and ω ($\mapsto$ `w`), because their closest phonetic counterparts have already been used for ϵ and o.

---

[1] Unfortunately, this irrational transliteration scheme is "common in most word processing systems", according to Hoover [1989, page 557 (cf. Figure 4)].

Finally, it is stretching the truth only slightly to say that there *is* a Latin phonetic equivalent to θ: There was an old English letter "thorn", which resembled a y and is still often represented by y on the signs of small businesses with names of the form "Ye Olde ...".

The only Greek letter left is ψ. It is natural to assign it to c, since both are somewhat superfluous sibilants. The two unused Latin letters are j and v.

I have been using macros implementing this transliteration scheme for over four years. After submitting the abstract for this presentation, I learned that similar systems had already been introduced by Silvio Levy [1988] for typesetting Greek *text*, and by John Collins [1990] in a TEX-compatible WYSIWYG editor for PCs. To my great relief, their choices[2] are almost identical to mine:

Latin	*Fulling*	*Collins*	*Levy*
c	ψ	ψ	[ς]
j	unused	unused	θ
v	unused	θ	unused
y	θ	unused	ψ

Clearly we are close to a consensus; let us all work to establish one.

In addition to lower-case letters, one will define macros for the capital Greek letters when necessary (e.g., \zC for Ψ). But there are 13 capital letters plus the lower-case omicron that are typographically identical with Latin letters. (E.g., we do not need \zQ for X, the capital chi.) The Latin letters thereby freed, along with both cases of j and v and the 10 numerals, can be used with \z to define additional macros at the user's discretion. (E.g., \zQ might be \defined as \subseteq.) In effect, \z acts (almost) as a second escape character. Examples are listed below.

Listing

The foregoing discussion is summarized and implemented by these macros:

```
    %% greek.mac %%
\def\za{\alpha}               % alpha   -> A
\def\zb{\beta}                % beta    -> B
\def\zc{\psi}                 % psi     -> C
    \def\zC{\Psi}
\def\zd{\delta}               % delta   -> D
    \def\zD{\Delta}
\def\ze{\epsilon}             % epsilon -> E
\def\zf{\phi}                 % phi     -> F
    \def\zF{\Phi}
\def\zg{\gamma}               % gamma   -> G
    \def\zG{\Gamma}
\def\zh{\eta}                 % eta     -> H
\def\zi{\iota}                % iota    -> I
%  (available for another purpose) -> J
\def\zk{\kappa}               % kappa   -> K
\def\zl{\lambda}              % lambda  -> L
    \def\zL{\Lambda}
\def\zm{\mu}                  % mu      -> M
\def\zn{\nu}                  % nu      -> N
% (implicitly assigned to omicron) -> O
\def\zp{\pi}                  % pi      -> P
    \def\zP{\Pi}
\def\zq{\chi}                 % chi     -> Q
\def\zr{\rho}                 % rho     -> R
\def\zs{\sigma}               % sigma   -> S
    \def\zS{\Sigma}
\def\zt{\tau}                 % tau     -> T
\def\zu{\upsilon}             % upsilon -> U
    \def\zU{\Upsilon}
%  (available for another purpose) -> V
\def\zw{\omega}               % omega   -> W
    \def\zW{\Omega}
\def\zx{\xi}                  % xi      -> X
    \def\zX{\Xi}
\def\zy{\theta}               % theta   -> Y
    \def\zY{\Theta}
\def\zz{\zeta}                % zeta    -> Z
```

Examples of possible ways to fill up the table are

```
\def\zI{\infty}      \def\zN{\emptyset}
\def\zj{\quad}       \def\zJ{\qquad}
\def\zK{\subset}     \def\zQ{\subseteq}
\def\zo{\oplus}      \def\zO{\otimes}
\def\zv{\partial}    \def\zV{\nabla}
\def\zZ#1{\ifcase#1 {}\or \displaystyle
    \or \textstyle \or \scriptstyle
    \or \scriptscriptstyle \fi}

\def\z#1#2{\ifcase#1 {\overline {#2}} %\z0
    \or{\if #2i {\tilde\imath}
\else\if #2j {\tilde\jmath}   %\z1
\else {\tilde #2} \fi\fi}
    \or{\if #2i {\hat\imath }
\else\if #2j {\hat\jmath}     %\z2
\else {\hat #2} \fi\fi}
 %% \or ...                   % more numerals
    \fi}
```

[2] More precisely, the treatment of σ and ς described here is a slight modification of Levy's, described by Haralambous and Thull [1989].

Bibliography

Collins, John. "ET, Version 1.05." Illinois Institute of Technology, 1990.

Haralambous, Yannis, and Klaus Thull. "Typesetting Modern Greek with 128 Character Codes." *TUGboat* 10(3), pages 354 - 359, 1989.

Hoover, Anita Z. "Using WordPerfect 5.0 to Create TEX and LATEX Documents." *TUGboat* 10(4), pages 549 - 559, 1989.

Levy, Silvio. "Using Greek Fonts with TEX." *TUGboat* 9(1), pages 20 - 24, 1988.