
A footnote about ‘Oh, oh, zero’

Don Knuth

I can’t resist adding a few comments to Chuck Bigelow’s wonderful essay about the history of zero-versus-oh in *TUGboat* **34:2** (2013), 168–181.

As an associate editor of ACM’s *Journal and Communications* during the 1960s, and as a prospective author, I’d been giving some thought to the fact that new concepts arising in computer science were calling for new typographic conventions. In particular, I corresponded with Myrtle Kellington, who was responsible for typography in all of ACM’s publications. (Computer scientists have Myrtle to thank for the now-universal style in which computer programs have long been typeset with a pleasant mixture of roman, bold, and italic. She introduced this style when she masterminded the publication of the Algol 60 Report, first with roman and bold only [1] and later with italic too [2].)

I applauded her work on formatted algorithms, but I wasn’t happy with the appearance of various papers about machine input and output, in cases where a monospaced font would have improved the exposition. When she learned of my concerns she wrote to me on 10 February 1966:

Whenever the vertical alignment is a requirement the printer uses the only typeface he has where this applies. It is called “Typewriter,” is machine-set, is available in four sizes, and looks like the old-fashioned typewriter style. A sample is enclosed.

Actually the sample she sent had three different fonts, and it didn’t exhibit the full character sets. Those fonts (all to be used on Monotype machines) were: Typewriter No. 74 (eight point size); Remington No. 70 (ten point and twelve point sizes); and Remington No. 17 (eleven point size), which was somewhat darker. Only No. 17 was available for machine setting; the other styles needed to be inserted by hand.

I replied on 14 February 1966—evidently the U.S. Postal service was quite efficient in those days!—as follows:

Regarding my request for a special type-font, I believe the 8 point “Typewriter No. 74” will do very nicely (assuming there are commas, parentheses, and the usual other special characters found on a typewriter). The other styles are also adequate but not as good. If possible it would be preferable to have a more squarish capital letter O so it can’t be mistaken for zero. I don’t know how expensive it

is to make up new characters one at a time; I realize a whole new font can be very costly. The special characters will no doubt be the major problem.

As examples of printed material *using* this style, I can only say unfortunately I don't know of any, except Addison-Wesley is doing it for me in the forthcoming set of books I'm writing. I think ACM should "pioneer" in this. The best example I can give you is point out sections of the last *Communications of the ACM* issue (January '66) which would have been much improved if set in a "typewriter" style:

Page 5, right column, the "all caps" words.

Page 6, "all caps" words and displayed programs.

Page 9, Appendix C. Possibly appendix B also.

Page 30, the tables, if type set, would be candidates for fixed width, although in this case the line engravings were quite adequate as a substitute.

Page 31, the words in all caps.

Page 32 ff. The FORTRAN and COBOL program example.

Pages 36–37, the capitalized "machine response" sentences.

Page 41 ff. All-caps words except perhaps ELIZA.

In general, FORTRAN and COBOL and assembly language programs and references to symbols within such programs would look much better in a fixed-width style. Even an 8 point type would be satisfactory here for appearance sake in the midst of the normal 12 point type (it would look like "small caps"), although perhaps there would be some trouble from the monotype side in such mixing of sizes, I don't know. You already have good formats for printing ALGOL programs, and that needs no change; but these others look quite unreadable by comparison, particularly things like Appendix C on page 9.

The distinction between "oh" and "zero" is reasonably important. On page 6 I see the word "TO" which should have really been tee-zero.

I would like to continue discussing this with you by letter. Can you tell me what special characters are available, how difficult

it is to mix 8 point in with 12 point text, etc.? I will be very glad to mark up all papers that go through my desk in a special way to indicate what parts should be in this fixed-width style.

Unfortunately, Myrtle's reply (dated 17 February) shot all these ideas down:

Dear Don: I am grateful for your interest in the printing aspects . . . However, I had not realized from your earlier letter that this new or special typeface with fixed-width characters would have to be used for words run in the text — not displayed, that is. What I am trying to say is that I had assumed that use of the Typewriter typeface would be only for certain special displayed sequences or programs.

Incorporating the Typewriter, or any other typeface, for isolated words or groups of words into the customary text would lead to prohibitive costs.

Let me tell you what can be done staying within machine-set composition, which is the most economical form of typesetting in letter press composition, and perhaps we can evolve a plan that would achieve what you are after, or at least partially.

The typeface used for the text in *CACM* is Modern No. 8, referred to as #8. This is available in all sizes both in roman and italics. . . . Along with #8 we can have, for machine setting that is, one other typeface. The one we use is Bodoni boldface, called #275. . . . To summarize, we can intermix in one keyboard operation for machine setting the following typefaces and sizes:

In size 10 on 12,

#8, in roman: all caps; csc (caps and small caps); sc (all small caps); and clc (caps and lowercase).

#8, in italics: all caps, and clc.

#275, in roman: all caps, and clc.

#275, in ital: all caps, and clc.

In 8 on 10, the same range.

But the two sizes, 8 pt and 10 pt, cannot be intermixed on the same line, I am sure you know — unless of course hand work is involved.

Thus for all the examples you mentioned in your letter, one could not have the Type-writer typeface without an entire special hand operation to drop all those words in after the regular text had been set by machine in #8 and #275.

... Actually we are giving considerable attention to changing printers, originally motivated by saving money, but now by many other considerations such as automated typesetting or being prepared for a fully automated operation all the way along the line, as well as quality of the printing. With certain of these new cold type and automated composition processes, one can intersperse several typefaces and adjust the spacings almost at whim. And your request would be no problem.

I will bear all this in mind as we carry on our interviews and observe demonstrations.

(Indeed, ACM did eventually change to “cold type” printing, and it turned out to be a mistake — although I’m sure they tried valiantly to work with the available vendors during those years. Decent mathematical typesetting was becoming a lost art; and that, of course, is why I was motivated to develop \TeX some years later. A downward spiral of decreasing typographic quality in *Communications of the ACM* began with their issue of March 1971; various examples of fixed-width type can incidentally be found in that issue, all of which were poorly reproduced from line-printer output. The *Journal of the ACM* began to suffer the same fate in October 1976.)

Meanwhile, as indicated in my letter to Myrtle, I had been having much better responses from Addison-Wesley, as they were preparing to publish *The Art of Computer Programming*. Addison-Wesley had been founded by two printers who were interested in producing good textbooks; and they became the only scientific publisher with their own in-house composition facility, at least in America. All of their typesetting was done under the direction of an old-timer named Hans Wolf. At my request, Hans had figured out a clever way to adapt his Monotype keyboards and casters so that machine setting with Remington #17 could actually be mixed together with the normal roman, italic, bold, bold italic, and math symbols. (Indeed, I’m pretty sure that this had never been done before, because Hans had originally told me — as Myrtle was to do later — that such a thing would be impossible.) This mixing could be done either in 10-point type on a 12-point base, or 9-point type on an 11-point base.

Furthermore, Hans and Addison-Wesley agreed to have a special glyph cut for me, a “squarish” version of the uppercase O, compatible with the existing Remington font. The font also included a new special character like ‘ \square ’ to indicate a blank space.

Therefore the publication of Volume 1 of *TAOCP* in January 1968 was actually the debut of a brand-new typographic style for computer science, featuring typewriter style blended freely with ordinary text in appropriate places. The new ‘O’ didn’t quite align properly with the other letters at the baseline; but I didn’t actually notice that glitch at the time, because I was so happy to have ‘O’ instead of ‘0’.

Why did I ask Addison-Wesley for a squarish Oh? I was almost surely influenced primarily by the dot-matrix font used by keypunch machines in those days. Look, for example, at the illustration of a punched card on page 148 of the original 1968 edition of my book, or on page 152 of the current edition. Bob Bemer’s article [3, page 516] also shows it as IBM’s recommended corporate practice for distinguishing Oh from zero on keypunches, as of 1964. On typewriters, IBM recommended a wide Oh and a narrow zero, “except for the stylized fonts for OCR and MICR.”

When Volume 2 of *The Art of Computer Programming* came out in 1981, with glyphs now drawn by METAFONT, of course I retained my beloved ‘O’. And the ‘Q’ became squarish too at this time (although with a loop at the bottom instead of a crossbar).

Alas, however, Chuck’s essay demonstrates that I’m still standing alone in this respect: None of the nine monospaced typefaces in his Fig. 9 have anything like an Oh that I would want to use. (Nowhere did I see a really satisfactory Oh in Chuck’s discussion — until I came to Karl Berry’s production notes at the end, and Karl’s reference to `ZeroFontOT.otf`.) I herewith submit a humble request to have squarish O and Q available as alternates in the next edition of Lucida Console.

References

- [1] Peter Naur et al., “Report on the Algorithmic Language ALGOL 60,” *Communications of the ACM* **3** (1960), 299–314.
- [2] Peter Naur et al., “Revised report on the Algorithmic Language ALGOL 60,” *Communications of the ACM* **6** (1963), 1–17.
- [3] R. W. Bemer, “Toward standards for handwritten zero and oh,” *Communications of the ACM* **10** (1967), 513–518.

◇ Don Knuth
<http://www-cs-staff.stanford.edu/~uno>