☞ Fonty

The Tao of Fonts

Włodzimierz Bzyl

Typographical journey

Throughout history people used symbols to visually encode thoughts and feelings. The oldest example I was able to find in the literature is an inscription from La Pasiega cave.

Fig. 1. Inscription from La Pasiega cave (Spain, ca. 10000 B.C)

Unfortunately we don't know how to decode this inscription. So, we don't know exactly what it means — we can only guess. The meaning of these symbols was lost in the past. Symbols are also found on pikes and bows. It is assumed that they are some kind of owner's signature or that they have a magical value, this means that they bring luck to the owner etc. Nowadays, we still use symbols in similar way. Probably everyone at least once received a mail overprinted with "CONFIDENTIAL – you were chosen to be rich. You can take part in our lottery-drawing. Only thing you have to do is to subscribe our magazine."

Fig. 2. 'Magical' stamps

Pictographs, ideograms, alphabets were written and reproduced on papyrus, stones, wood, clay tablets, paper, computer displays and different techniques were used to write, carve, cut, and print symbols.

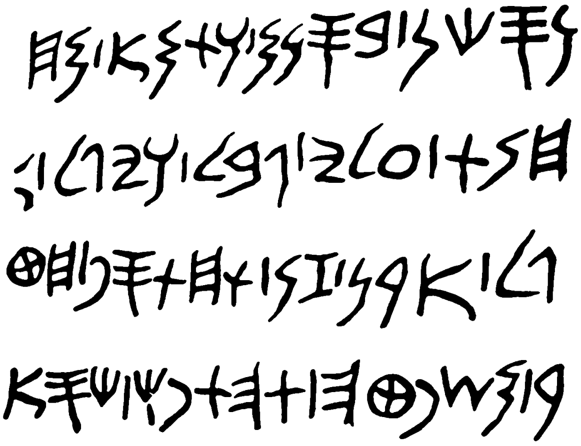


Fig. 3. Phoenician inscription (Byblos, 1100 B.C.)

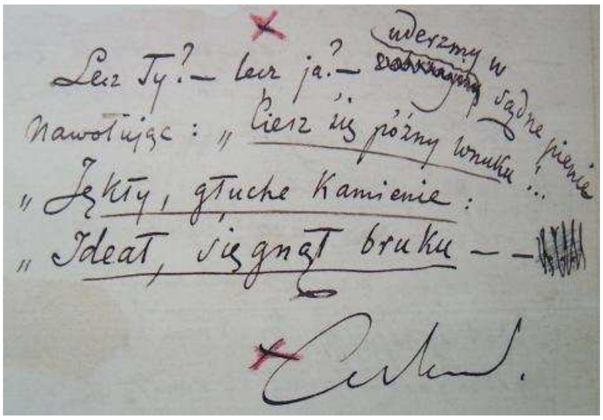


Fig. 4. Cyprian K. Norwid, *Vade-mecum* (manuscript in Polish, 1865–1866)

Handwriting used to be the most common technique. Phoenicians invented alphabetic font which is precursor of Greek, Latin, Cyrillic fonts. The inscription on Fig. 3 was originally written on papyrus. Handwriting can reveal the author's personality and that makes this technique interesting. Wouldn't it be nice to have a psychological portrait of the author?

Fig. 4 shows a fragment of a poem written by probably the greatest Polish poet Cyprian Kamil Norwid. This poem seems to move every Pole who reads it. I think that the same poem printed along with several others in Computer Modern, Garamond, Times, or Palatino would not have that impact on readers. So, maybe this is why I don't like reading poems which all look the same after a while.

: But my doctor should use Computer Modern on prescriptions. I would be very grateful if he did.

Fig. 5. *Calendarium Parisiense* (manuscript Latin, ca. 1425)

But the main problem caused by handwriting (and computer typesetting too) is an appearance of overfull and underfull lines. These, in some cases, can not be eliminated. Fig. 5 shows the earliest example I was able to find.

Another technique is cutting in stone. The shapes in Fig. 6 are more regular, partly because bigger letters were sketched beforehand.

This picture shows the earliest example of serifs ever found. The serifs are functioning here as a way of finishing letters, which otherwise would be irregular and wiggly ended. Nowadays, we think that serif fonts are easier and quicker to read. This is generally true, because letters without serifs in some cases looks similar to each other, for example: `I– l– 1`.

Fig. 7 shows the 32nd page of one of the first printed books in Poland. This book is very important, because it contains designs for Polish

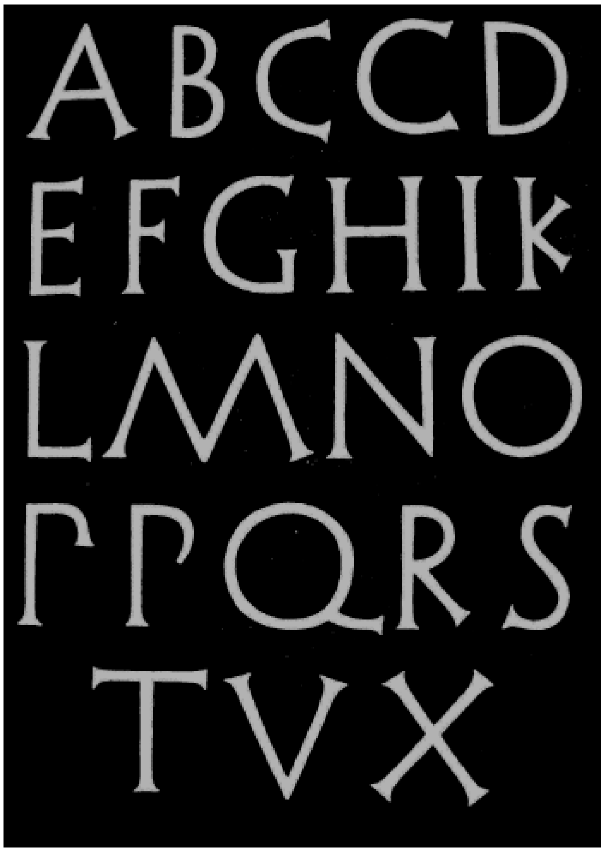


Fig. 6. Rome, (ca. 200 B.C.)

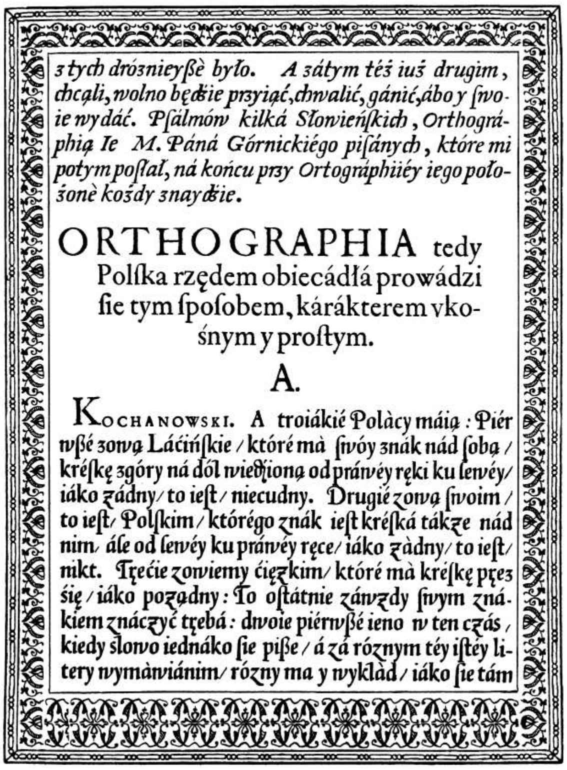
diacritics: aogonek, eogonek etc. The borders were cut in wood and letters in metal. The printed letters are much smaller than curved ones, so cutting them was a real challenge. Even tiny serifs are present. This technique was perfected over the years. The results are seen on Fig. 8. Here borders were cut in wood and the type was cut in metal too.

Finally we get to Fig. 9 showing what computers are good at. 'Shapes' drawn by a computer are almost ideal. Typographical embellishments are not present. Instead the letters are colored and the background photo is used to improve typography of the page.

Each new technique starts from the point where the old one ends.

When we make a letter with a computer, mouse is used to put points on 'a imaginary sheet'. At the same time when points are laid down computer joins them with curves. Next, red points and the endings of green lines are adjusted by hand. The computer at the same time, see Fig. 10, redraws the shape. So, a chisel was replaced by a mouse and wood/metal

: I can not imagine books and newspapers carved in stones. Nevertheless, there was on exception: I saw Fred Flinston reading newspapers. But this was long time ago down in Bed Rock.



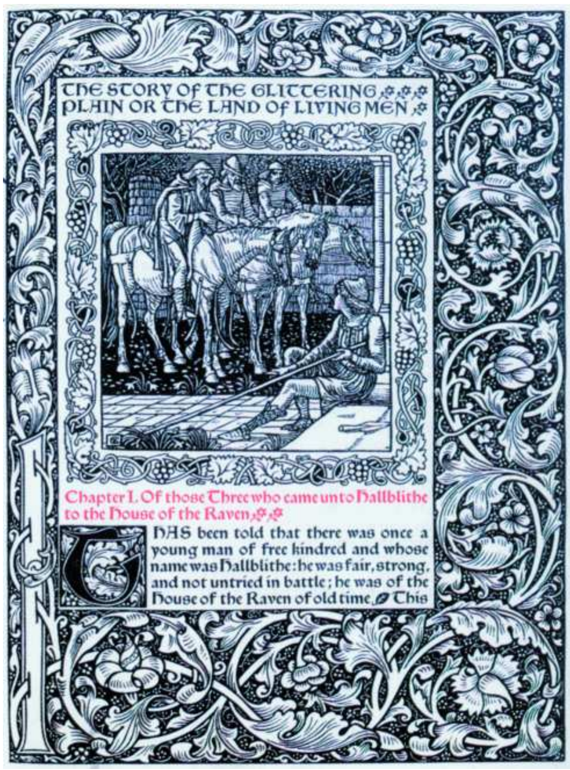
ʒ tych dróʒnieyßè było. A ʒátym téʒ iuʒ drugim, chcąli, wolno będźie przyiąć, chwalić, gánić, ábo y ſwo-ie wydáć. Pſálmów kilká Słowieńſkich, Orthográ-phią Ie M. Páná Górnickiégo piſánych, które mi potym poſłał, ná końcu przy Ortográphiiéy iego poło-ʒonè koʒdy ʒnaydźie.

ORTHOGRAPHIA tedy Polſka rzędem obiecádłá prowádzi ſie tym ſpoſobem, kárákterem vko-śnym y proſtym.

A.

Kochanowski. A troiákié Polàcy máią: Piérwßé ʒową Láćińſkie / które mà ſwóy ʒnák nád ſobą / kréſkę ʒgóry ná dół wiedʒioną od práwéy ręki ku lewéy/ iáko ʒádny/ to ieſt/ niecudny. Drugié ʒową ſwoim / to ieſt/ Polſkim / którégo znák ieſt kréſká tákze nád nim/ ále od lewéy ku práwéy ręce/ iáko zàdny/ to ieſt/ nikt. Tręćie zowiemy ćiężkim/ które mà kréſkę pręʒ śię / iáko pozadny: To oſtátnie zánzdy ſwym zná-kiem znáczyć trebá: dwoie piérwßé ieno w ten czás, kiedy słowo iednáko ſie piße / á zá róznym téy iſtéy li-tery wymàwiánim/ rózny ma y wyklàd / iáko ſie tám

Fig. 7. *New Polish Character* (Jan Januszowski, 1594 Cracow)



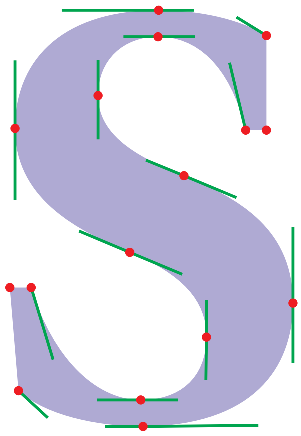
THE STORY OF THE GLITTERING PLAIN OR THE LAND OF LIVING MEN

Chapter I. Of those Three who came unto Hallblithe to the House of the Raven

T HAS been told that there was once a young man of free kindred and whose name was Hallblithe: he was fair, strong, and not untried in battle; he was of the House of the Raven of old time. This

Fig. 8. *The Story of the Glittering Plain* (Kelmscott Press, 1891)

Fig. 9. “Playboy” (Polish edition, 2001)



```
newpath
487 499 moveto
451 499 lineto
423 616 363 661 299 661 curveto
239 661 195 621 195 559 curveto
195 483 278 447 344 420 curveto
435 382 533 331 533 199 curveto
533 95 473 -13 273 -15 curveto
207 -15 108 0 57 47 curveto
42 226 lineto
79 226 lineto
117 101 193 31 269 31 curveto
334 31 382 66 383 140 curveto
383 204 341 249 250 287 curveto
162 324 51 378 51 502 curveto
51 620 132 707 300 707 curveto
361 707 435 695 487 663 curveto
487 499 lineto
closepath
fill
```

Fig. 10

by a computer screen. Therefore we can polish the shape as long as we wish. We can not ruin the shape with the one wrong decision as it can happen in the case of the traditional tools and materials.

One of the limitations of this technique is shown on Fig. 11. Printing a symbol by a computer means putting the imaginary sheet on the screen. This sheet could be expanded, contracted, skewed. We can apply any geometrical transformation to it. Unfortunately, the results are not satisfactory. The stop sign in the third row lost its white border and the inscription is hardly readable. We could repair this if it would be possible to drop the border

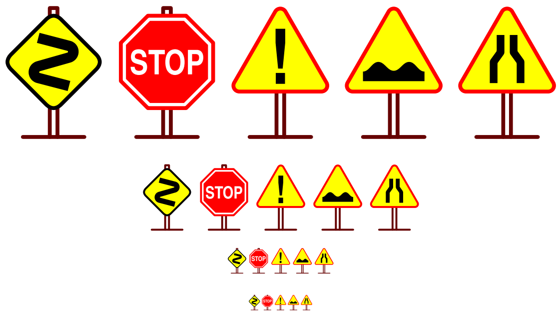


Fig. 11

and scale the inscription less. But this operation is unfeasible, because computer does not know which numbers are responsible for the border and which for the inscription.

So the only way to produce a better font at small sizes is to make it from scratch. Now, some will inevitably use enlarged version of this font instead of unscaled one. Chaos will ensue. Imagine country where each town has slightly different signposts!

Programming fonts

Typographic standards make type more readable. But readability has become a relative concept. The immediacy of personal computers and World Wide Web raised the level of 'typographic literacy'. Computers are used to stretch typographic boundaries [14].

Before we start to explore type with computers, we should ask: *What is the right way to create digitized patterns for printing or displaying?* In this section, I will try to convey some of my excitement about exploratory experiments with the tools created by myself, since I think that I am going in the right direction [see also 3].

To play and explore fonts I use a set of UNIX tools. To this set I added, after conversion to UNIX scripts, DOS batch files forming METATYPE1 package by the JNS team [11].

The language for font programming is METAPOST. To make METAPOST font usable we must convert it into a program understandable by operating system. The choice is PostScript. Mainly because the Ghostscript — free Postscript interpreter is available on every computer platform.

The Linux version of tools consists of METAPOST font libraries and four scripts:

`mkfont1` – converts METAPOST font sources to Type 1 font: this shell script uses programs `mpost`, `t1asm` from the T1UTILS package and `awk`

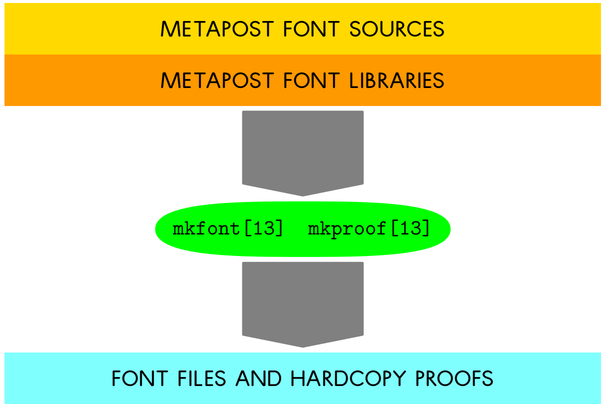


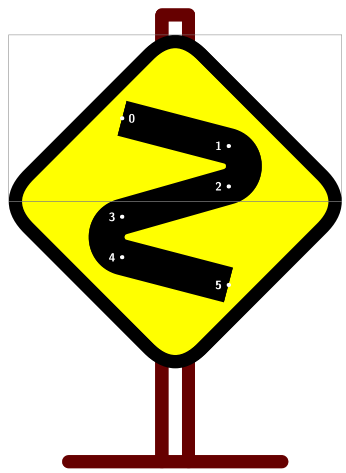
Fig. 12. Fonts programming

`mkfont3` – converts METAPOST font sources to Type 3 font; this perl script calls `mpost`

`mkproof1` and `mkproof3` – these scripts produce hardcopy proofs and are used as debugging tools: they call `mpost` and the `mft` pretty-printer.

In font programming two type of errors appear: bugs in font program and design errors. Bugs are

SIGN-000.MP




```
beginpic(127, 250, 125, 0); "Dangerous␣bend";
    draw post; draw info_signboard;
    clearxy;
    % the dangerous bend
    numeric heavyline; heavyline := 27;
    x5 = w − x0; x5 − x0 = 80; x1 = x2 = x5; x0 = x3 = x4;
    y0 = −y5 = 1/2 h; y1 = −y4 = 1/3 h; y2 = −y3 = 1/11 h;
    pickup pencircle scaled heavyline;
    interim linecap := butt;
    draw z0 -- z1{z1 − z0} .. z2 --- z3 .. z4{z5 − z4} -- z5
        withcolor c.Dangerous Bend;
    labelcolor := white; dotcolor := white;
    labels lft(1, 2, 3, 4, 5); labels rt(0);
endpic;
```

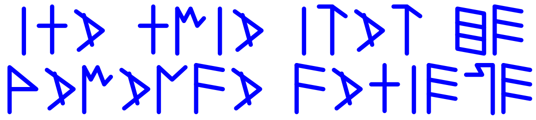
11:57 11 VII 2001 *7*

Fig. 13. Hardcopy proof of the dangerous bend

treated in an ordinary way. To catch design errors I use a hardcopy proofs produced by the `mft` program.

We have tools, so its high time to start programming. Let's start from the beginning.

Phoenician font lacks of vowels. There are three ways of writing with this font. Lines are written from left to right, right to left, or alternatively left to right, right to left, etc. and letters on each second line are reflected vertically. It could be a real challenge to typeset a Phoenician script with TEX.



To [our] Lady Ishtar
This is the holy place

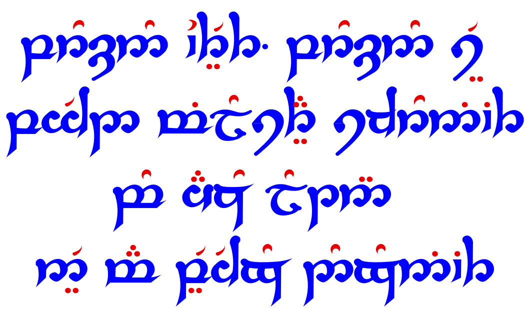
Fig. 14. Phoenician font [23]

TEX could be used to communicate with a STAR TREK crew. We only need to get their font. No problem. There is nothing special about this font except extraordinary shapes of symbols and use of few ligatures.

Listen sons of Kahless!
Listen his daughters!

Fig. 15. Klingon font [21]

We can also send our classic love poems to elves. The elves write vowels over preceding letter unless it is also a vowel. This case and the case when vowel starts a word are handled by other rules [18].



— JAN KOCHANOWSKI, About love

Fig. 16. Tengwar font [22]

It appears that that all rules could be implemented with an appropriate ligature and kerning table.

Fig. 17. Redis font (see also [25])

What was considered unreadable yesterday is readable today. People are more visually sophisticated and typographically savvy than ever before, so my next font, except letters, contains ideograms for love and for some other emotions.

Special math fonts could be useful too. We can use them on title pages, on slides. Math characters are colored according to their math class as defined in plain format. So, binary operators are painted in green, large operators in magenta etc.

Fig. 19 shows a piece of text typeset with a computer replica of the font used in Polish ABC Primer by Falski. I learned to write letters with the help of Falski's primary, my wife and my daughter too. All children in Poland learn Falski's letters. The characteristic features of this font are listed below:

- size of characters: BIG,
- width: PROPORTIONAL,
- slanting: UPRIGHT,
- interletter spacing: BIG,
- uniformity of pressure: CONSTANT,
- strength of pressure: AVERAGE,
- interword spacing: BIG,

$$M = \begin{array}{c c} & \begin{array}{ccc} C & I & C' \end{array} \\ \begin{array}{c} C \\ I \\ C' \end{array} & \begin{pmatrix} 1 & 0 & 0 \\ \beta & 1-\beta & 0 \\ 0 & \xi & 1-\xi \end{pmatrix} \end{array}$$

$$\begin{aligned} \left(\int_{-\infty}^{\infty} e^{-x^2}\,dx\right)^2 &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-(x^2+y^2)}\,dx\,dy \\ &= \int_0^{2\pi}\int_0^{\infty} e^{-r^2}\,r\,dr\,d\theta \\ &= \int_0^{2\pi}\left(-\frac{e^{-r^2}}{2}\bigg|_{r=0}^{r=\infty}\right)d\theta \\ &= \pi \end{aligned} \tag{78}$$

Fig. 18. New Punk Math font (see also [19])

DIALOGS ARE REWRITTEN AGAIN TO SPARE THE AUTHOR AND THE OTHER CHARACTERS THE SHAME OF SOUNDING AS INARTICULATE AS THEY INVARIABLY DO OR WOULD IF THEIR SENTENCES ALMOST INVARIABLY BEGUN WITH THE WORD DUDE AS IN FOR EXAMPLE DUDE SHE DIED WERE MERELY TRANSCRIBED

Fig. 19. Ala font [24]

- overall appearance: OVAL,
- readability: CLEAR.

Handwriting can reveal a personality of the writer. The writer in this case is my computer so we can get 'his' psychological portrait easily: *A person which writes this way is well-wishing, Easy to cooperate with and friendly. Usage of uppercase letters and big interletter spacing indicates this. Constant pressure means emotional maturity. Oval appearance might mean uncertainty and submissiveness. Finally, constant and average pressure and wide characters indicate uneasy an over excitable person.*

The most important thing about this example is that how easy is to make this font to look differently, for example more 'texnical'. It suffices to change few numbers which define this font. But it would be difficult to simulate other important features of handwritten scripts, such as variable baseline, variable letters shape, pressure of script.

The last example shows the I-Ching font (see also [26]). The I Ching or „The Book of Changes" is an old Chinese oracle (Fig. 20).

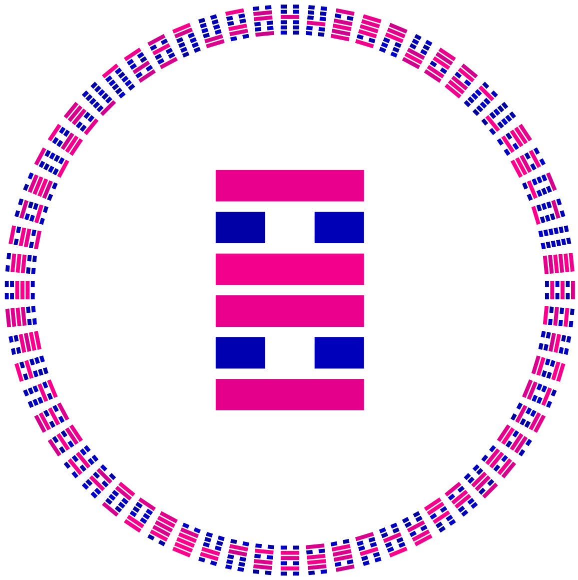
Type 1 font example

Type 1 font programming differs from Type 3 font programming. Type 3 glyphs can use any PostScript command, but Type 1 glyphs use a subset of PostScript. Moreover, we must construct an outline of glyph instead drawing it. The outline is filled when the glyph is printed.

Each METAPOST font should input METAPOST Type1 library. The library contains macros which help to compute outlines, output various font data to several files. These data are used by the mkfont1 script which assembles Type 1 font and mkproof1 script which typesets hardcopy proofs.

LI

To Shine Brightly, to Part



INTERPRETATION

To Part. It is useful to stand firm and behave well. This will bring success. Take care of the cows. There will be good fortune.

Interpretation of the 4 th change line

It comes unexpectedly. It is like a fire which dies down and is discarded.

Fig. 20. I-Ching font

The Type 1 font programmed below contains only plus symbol. Let's start the with reading basic macros.

```
input type1;
```

Next follows the usual font administration stuff. Each font should define several variables [see Tables 5.1–4, 9].

```
pf_info_familyname "Plus";
pf_info_fontname "Plus-Regular";
pf_info_weight "Normal";
pf_info_version "1.0";
pf_info_fixedpitch true;
pf_info_author "Anonymous 2002";
pf_info_creationdate;
```

The mpost program does all its drawing on its internal 'graph paper' with 1000 1000 coordinate space on it. The data below provides information about how to typeset with this font.

```
pf_info_quad 760;
pf_info_capheight 760;
pf_info_xheight 760;
pf_info_space 333;
```

The adl suffix is the acronym for *Ascender*, *Descender*, and *Lineskip*.

```
pf_info_adl 750, 250, 0;
```

The PostScript ll operator is used to paint the entire region closed by the current path. For each path, *non-zero winding number rule* [p. 161, 9] determines whether a given point is inside a path. This behaviour is simulated by Fill and unFill macros. The fill_outline macro, for each closed path stored in the array s[1..s.num], fills or unfills it based on its *turning number* [p. 111, 4].

```
def fill_outline suffix s =
  for i:=1 upto s.num:
    if turningnumber s[i] > 0: Fill
    else: unFill fi s[i];
  endfor
enddef;
```

The plus sign has squared-off ends. Macro butt_end simplifies the task of cutting of ends of paths.

```
def butt_end(text nodes) =
  cut(rel 90)(nodes)
enddef;
```

Horizontal line of the same width as vertical line seems to be thicker. To cancel this optical illusion we use an elliptical pen.

```
numeric px; px:=100;
numeric py; py:=90;
default_nib:=fix_nib(px,py,0);
```

These names are intended to make code more readable.

```
path vertical_stem, horizontal_stem;
path glyph;
```

Each glyph should be defined within beginfont and endfont block.

```
beginfont
```

Programmed symbols must be given names and positions in the font.

```
encode("plus",43);
```

Each glyph starts with beginglyph and ends with endglyph macro. The following macros initialize several variables, used for the glyph data bookkeeping.

```
standard_introduce("plus");
beginglyph("plus");
```

For convenience, width, height and depth of the character are assigned to variables w, h, and d.

```
w:=760; h:=760; d:=0;
```

The horizontal and the vertical bar of the plus glyph are centered with respect its bounding box.

```
z0=(w/2,d); z1=(w/2,h);
z2=(0,(h-d)/2); z3=(w,(h-d)/2);
```

To draw paths z0--z1 and z2--z3 the pen with default_nib shaped nib is used. The macro pen_stroke finds outline of each path. Outlines are assigned to vertical_stem and horizontal_stem paths. The macro butt_end cuts off ends these paths at times 0 (begining) and 1 (end).

```
pen_stroke(butt_end(0,1))(z0--z1)
  (vertical_stem);
pen_stroke(butt_end(0,1))(z2--z3)
  (horizontal_stem);
```

Programming a Type 1 glyph means constructing its outline (which could be made up several cyclic paths). The macro below finds the outline of the paths constructed above and stores it in the array named in the second argument.

```
find_outlines
  (vertical_stem,horizontal_stem)(glyph);
```

Now, we are ready to draw the plus symbol.

```
fill_outline glyph;
```

Finally, we fix the width of the glyph to w and its left and right sidebearings to 0.

```
fix_hsbw(w,0,0);
```

Each symbol should include so called *hints* [p. 56–57, 8] that make it render better on a wide variety of devices.

```
fix_hstem(py)(horizontal_stem);
fix_vstem(px)(vertical_stem);
```

To make hardcopy proof more readable we put on it conctruction points (see the figure on the next page).

```
dotlabels(0,1,2,3);
```

The last two macros end the subprogram for plus symbol and the whole font program.

```
endglyph;
endfont;
```

Now, we can create a TFM file, Type 1 font, and dvips fontmap file with the command:

```
mkfont1 plus.mp
```

The plus character just constructed is used to print the divider line below.

+ ++++ + +

The hardcopy proof below was typeset with the command:

```
mkproof1 -u plus.map plus.mp
```

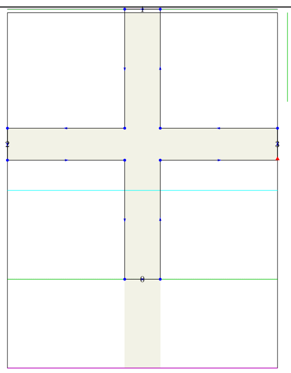
```
def fill_outline suffix s =
   for i := 1 upto s_num:
      if turningnumber s[i] > 0: Fill else: unFill fi s[i] ;
   endfor
enddef ;

def butt_end(text nodes) = cut(rel 90)(nodes) enddef ;

numeric px ; px := 100 ;
numeric py ; py := 90 ;
default_nib := fix_nib(px, py, 0) ;

path vertical_stem, horizontal_stem, glyph ;

beginfont
```



```
encode("plus", 43) ;
standard_introduce("plus") ;

beginglyph("plus") ;
   w := 760 ; h := 760 ; d := 0 ;
   z0 = (w/2, d) ; z1 = (w/2, h) ;
   z2 = (0, (h - d)/2) ; z3 = (w, (h - d)/2) ;

   pen_stroke(butt_end(0, 1))(z0 -- z1)(vertical_stem) ;
   pen_stroke(butt_end(0, 1))(z2 -- z3)(horizontal_stem) ;
   find_outlines(vertical_stem, horizontal_stem)(glyph) ;

   fill_outline glyph ;

   fix_hsbw(w, 0, 0) ;

   fix_hstem(py)(horizontal_stem) ;
   fix_vstem(px)(vertical_stem) ;

   dotlabels(0, 1, 2, 3) ;
endglyph ;

endfont ;
```

21:05 17 X 2002 PLUS 3

References

[1] John D. Hobby. 1992. *A user's Manual for MetaPost*. Technical Report 162. AT&T Bell Laboratories, Murray Hill / New Jersey. Available online as a part of METAPOST distribution.

[2] Donald E. Knuth. 1982. "The Concept of a Meta-Font." *Visible Language* 16, 3–27.

[3] Donald E. Knuth. 1985. "Lessons Learned from METAFONT." *Visible Language* 19, 35–53.

[4] Donald E. Knuth. 1986. *The METAFONTbook*. American Mathematical Society and Addison-Wesley.

[5] Donald E. Knuth. 1992. *Computer Modern Typefaces*. Addison-Wesley.

[6] Adobe Systems Incorporated. 1985. *Tutorial and Cookbook*. Addison-Wesley.

[7] Adobe Systems Incorporated. 1992. *The PostScript Font Handbook*. Addison-Wesley.

[8] Adobe Systems Incorporated. 1993 (3rd printing), Version 1.1. *Adobe Type 1 Font Format*. Addison-Wesley.

[9] Adobe Systems Incorporated. 1999 (3rd printing). *PostScript Language Reference Manual*. Addison-Wesley.

[10] Bogusaw Jackowski at all. 1999. "Antykwa Pótawskiego: a parametrized outline font." EuroTEX 99 Proceedings. Ruprecht-Karls-Univerität Heidelberg, 117–141.

[11] Bogusaw Jackowski, Janusz M. Nowacki, and Piotr Strzelczyk. 2001. "METATYPE1: A MetaPost-based engine for generating Type 1 fonts." EuroTEX 2001 Proceedings. Kerkrade, the Nedtherlands, 111–119.

[12] Wodzimierz Bzyl. 2001. "Re-introducing Type 3 fonts to the world of TEX." EuroTEX 2001 Proceedings. Kerkrade, the Nedtherlands, 219–243.

[13] Apostolos Syropoulos. 2000. "The MF2PT3 tool." Available online from www.obelix.ee.duth.gr/~apostolo.

[14] Rob Carter. 1997. *Experimental Typography*. A RotoVision Book. Watson Guptill Publications.

[15] František Muzika. 1965. *Die Schöne Schrift*. Verlag Werner Dausien, Hanau/Main. Vol I & II.

[16] Halina Tchórzewska Ed. 2000. *More Precious Than Gold. Treasures of the Polish National Library*. Biblioteka Narodowa. Warszawa.

[17] Charlotte & Peter Fiell. 1999. *William Morris (1834–1896)*. Benedikt Taschen Verlag GmbH.

[18] J. R. R. Tolkien. 1981. *The Fellowship of the Ring*. Spódzielnia Wydawnicza "Czytelnik". Warszawa.

[19] Donald E. Knuth. 1988. "A Punk Meta-Font". *TUGboat* 9, 152–168.

[20] Mark Shoulson, 1994. *Okuda Font*. METAFONT source available online from CTAN/fonts/okuda.

[21] Karol Jarmakiewicz. 2002. *Czcionka Klingoska*. Instytut Matematyki, Uniwersytet Gdaski.

[22] Mieszko Zieliski. 2002. *Kto i dlaczego wymyli Tengwar*. Instytut Matematyki, Uniwersytet Gdaski.

[23] Wojciech Górski. 2002. *Font Fenicki*. Instytut Matematyki, Uniwersytet Gdaski.

[24] Sawomir Lis. 2002. *Pismo Rczne*. Instytut Matematyki, Uniwersytet Gdaski.

[25] Jacek Neuman. 2002. *Just Smiley!*. Instytut Matematyki, Uniwersytet Gdaski.

[26] Alan M. Stanier. 1994. *iching*. METAFONT source available online from CTAN/fonts/iching.

[27] Jan Jelinek. 1977. *Wielki Atlas Prahistorii Czowieka*. Pastwowe Wydawnictwo Rolnicze i Lene. Warszawa.

Włodzimierz Bzyl
matwb@univ.gda.pl