Table 3: Virtual font with accented letters generated from `TFM` of the “raw” `ADOBE` font from Table 2.

| | ´0 | ´1 | ´2 | ´3 | ´4 | ´5 | ´6 | ´7 | |
|---|---|---|---|---|---|---|---|---|---|
| ´00x | | | | | | | | | ″0x |
| ´01x | | | | | ﬁ | ﬂ | | | |
| ´02x | ı | | ` | ´ | ˇ | ˘ | ¯ | ˚ | ″1x |
| ´03x | ¸ | ß | æ | œ | ø | Æ | Œ | Ø | |
| ´04x | | ! | ” | # | $ | % | & | ’ | ″2x |
| ´05x | ( | ) | * | + | , | - | . | / | |
| ´06x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ″3x |
| ´07x | 8 | 9 | : | ; | ¡ | = | ¿ | ? | |
| ´10x | @ | A | B | C | D | E | F | G | ″4x |
| ´11x | H | I | J | K | L | M | N | O | |
| ´12x | P | Q | R | S | T | U | V | W | ″5x |
| ´13x | X | Y | Z | [ | “ | ] | ˆ | ˙ | |
| ´14x | ‘ | a | b | c | d | e | f | g | ″6x |
| ´15x | h | i | j | k | l | m | n | o | |
| ´16x | p | q | r | s | t | u | v | w | ″7x |
| ´17x | x | y | z | – | — | ˝ | ˜ | ¨ | |
| ´20x | Ă | Ą | Ć | Č | Ď | Ě | Ę | Ğ | ″8x |
| ´21x | Ĺ | Ľ | Ł | Ń | Ň | | Ő | Ŕ | |
| ´22x | Ř | Ś | Š | Ş | Ť | Ţ | Ű | Ů | ″9x |
| ´23x | Ÿ | Ź | Ž | Ż | | İ | | | |
| ´24x | ă | ą | ć | č | ď | ě | ę | ğ | ″Ax |
| ´25x | ĺ | ľ | ł | ń | ň | | ő | ŕ | |
| ´26x | ř | ś | š | ş | ť | ţ | ű | ů | ″Bx |
| ´27x | ÿ | ź | ž | ż | | | | | |
| ´30x | À | Á | Â | Ã | Ä | Å | | Ç | ″Cx |
| ´31x | È | É | Ê | Ë | Ì | Í | Î | Ï | |
| ´32x | | Ñ | Ò | Ó | Ô | Õ | Ö | | ″Dx |
| ´33x | | Ù | Ú | Û | Ü | Ý | | | |
| ´34x | à | á | â | ã | ä | å | | ç | ″Ex |
| ´35x | è | é | ê | ë | ì | í | î | ï | |
| ´36x | | ñ | ò | ó | ô | õ | ö | | ″Fx |
| ´37x | | ù | ú | û | ü | ý | | | |
| | ″8 | ″9 | ″A | ″B | ″C | ″D | ″E | ″F | |

Table 2: “Raw” ADOBE stardard encoding font layout.

| | ´0 | ´1 | ´2 | ´3 | ´4 | ´5 | ´6 | ´7 | |
|---|---|---|---|---|---|---|---|---|---|
| ´04x | | ! | " | # | $ | % | & | ’ | ″2x |
| ´05x | ( | ) | * | + | , | - | . | / | |
| ´06x | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ″3x |
| ´07x | 8 | 9 | : | ; | < | = | > | ? | |
| ´10x | @ | A | B | C | D | E | F | G | ″4x |
| ´11x | H | I | J | K | L | M | N | O | |
| ´12x | P | Q | R | S | T | U | V | W | ″5x |
| ´13x | X | Y | Z | [ | \ | ] | ^ | _ | |
| ´14x | ‘ | a | b | c | d | e | f | g | ″6x |
| ´15x | h | i | j | k | l | m | n | o | |
| ´16x | p | q | r | s | t | u | v | w | ″7x |
| ´17x | x | y | z | { | \| | } | ~ | | |
| ´24x | | ¡ | ¢ | £ | ⁄ | ¥ | ƒ | § | ″Ax |
| ´25x | ¤ | ' | “ | « | ‹ | › | fi | fl | |
| ´26x | | – | † | ‡ | · | | ¶ | • | ″Bx |
| ´27x | ‚ | „ | ” | » | … | ‰ | | ¿ | |
| ´30x | | ` | ´ | ˆ | ˜ | ¯ | ˘ | ˙ | ″Cx |
| ´31x | ¨ | | ˚ | ¸ | | ˝ | ˛ | ˇ | |
| ´32x | — | | | | | | | | ″Dx |
| ´33x | | | | | | | | | |
| ´34x | | Æ | | ª | | | | | ″Ex |
| ´35x | Ł | Ø | Œ | º | | | | | |
| ´36x | | æ | | | | ı | | | ″Fx |
| ´37x | ł | ø | œ | ß | | | | | |
| | ″8 | ″9 | ″A | ″B | ″C | ″D | ″E | ″F | |

Table 1: Default output font layout of `ACCENTS`.

| | *´0* | *´1* | *´2* | *´3* | *´4* | *´5* | *´6* | *´7* | |
|---|---|---|---|---|---|---|---|---|---|
| *´00x* | Γ | Δ | Θ | Λ | Ξ | Π | Σ | ϒ | ″0x |
| *´01x* | Φ | Ψ | Ω | ﬀ | ﬁ | ﬂ | ﬃ | ﬄ | |
| *´02x* | ı | ȷ | ` | ´ | ˇ | ˘ | ¯ | ˚ | ″1x |
| *´03x* | ¸ | ß | æ | œ | ø | Æ | Œ | Ø | |
| *´04x* | - | ! | ” | # | $ | % | & | ’ | ″2x |
| *´05x* | ( | ) | * | + | , | - | . | / | |
| *´06x* | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ″3x |
| *´07x* | 8 | 9 | : | ; | ¡ | = | ¿ | ? | |
| *´10x* | @ | A | B | C | D | E | F | G | ″4x |
| *´11x* | H | I | J | K | L | M | N | O | |
| *´12x* | P | Q | R | S | T | U | V | W | ″5x |
| *´13x* | X | Y | Z | [ | “ | ] | ˆ | ˙ | |
| *´14x* | ‘ | a | b | c | d | e | f | g | ″6x |
| *´15x* | h | i | j | k | l | m | n | o | |
| *´16x* | p | q | r | s | t | u | v | w | ″7x |
| *´17x* | x | y | z | – | — | ˝ | ˜ | ¨ | |
| *´20x* | Ă | Ą | Ć | Č | Ď | Ě | Ę | Ğ | ″8x |
| *´21x* | Ĺ | L’ | Ł | Ń | Ň | | Ő | Ŕ | |
| *´22x* | Ř | Ś | Š | Ş | Ť | Ţ | Ű | Ů | ″9x |
| *´23x* | Ÿ | Ź | Ž | Ż | | İ | | | |
| *´24x* | ă | ą | ć | č | d’ | ě | ę | ğ | ″Ax |
| *´25x* | ĺ | l’ | ł | ń | ň | | ő | ŕ | |
| *´26x* | ř | ś | š | ş | t’ | ţ | ű | ů | ″Bx |
| *´27x* | ÿ | ź | ž | ż | | | | | |
| *´30x* | À | Á | Â | Ã | Ä | Å | | Ç | ″Cx |
| *´31x* | È | É | Ê | Ë | Ì | Í | Î | Ï | |
| *´32x* | | Ñ | Ò | Ó | Ô | Õ | Ö | | ″Dx |
| *´33x* | | Ù | Ú | Û | Ü | Ý | | | |
| *´34x* | à | á | â | ã | ä | å | | ç | ″Ex |
| *´35x* | è | é | ê | ë | ì | í | î | ï | |
| *´36x* | | ñ | ò | ó | ô | õ | ö | | ″Fx |
| *´37x* | | ù | ú | û | ü | ý | | | |
| | ″8 | ″9 | ″A | ″B | ″C | ″D | ″E | ″F | |

Some characters from the input table may disappear in the course of the font layout translation. As noted earlier, all of them remain in the source font, and hence they can be accessed there should the user needed them.

Accented font from Table 3 has been generated from the metric information describing the "raw" `ADOBE` font from Table 2.

## 4 Conclusion

The `ACCENTS` processor has been presented, designed for automatic generation of accented virtual fonts for European languages from English input fonts in the TEX text font layout. We have discussed the reasons why accented virtual fonts are worth being considered as a viable alternative to genuine METAFONT-defined accented fonts.

The `ACCENTS` processor has been programmed in `WEB` with substantial parts taken from `VFTOVP` and `VPTOVF`. It can be distributed freely and its source text modified under the GNU General Public Licence condition of the Free Software Foundation. So far, change files for ports under MSDOS and SCO Unix are available (by the author and David Toman).

## References

[Ferguson90] Michael J. FERGUSON: "Fontes latines européennes et TEX 3.0", *Cahiers GUTenberg*, no. 7, November 1990, pp. 29–31; adaptation for French from *TUGboat*, vol. 11 no. 4.

[Novák89] Petr NOVÁK, *private communication*, 1989.

In our case, one of the starting design requirements was the ability to process `dvi` files with virtual accented letters also by older drivers. `ACCENTS` therefore takes care to ensure that

- the generated `TFM` width is exactly the source `TFM` width of the core character; and
- the core character is always printed last, so that its right end aligns with the right end of the whole virtual character.

As a result of the precautions described above, the `dvi` file containing references to virtual accented fonts are guaranteed to work even with older `dvi` drivers, after performing virtual font characters expansion using `DVICOPY`.

### 3.4 Additional metric information

As far as the information from the ligature and kerning array of the input `TFM` is concerned, it is expanded and copied into the output `TFM`. The expansion concerns accented characters: they inherit the same kerning information as their core ones.

Ligatures are copied verbatim: any ligature defined in the input font will be accessible from the virtual font as well. (Provided all the participating symbols make it through into the `VF` font.)

### 3.5 Processing `ADOBE` encoded input

Virtual font generation can also be combined with character encoding rearrangement, as used by, e.g., the `AFM2TFM` by from `dvips` distribution.

The `ACCENTS` processor can recognize `ADOBE` standard encoding `TFM`'s on the input, and to perform two tasks in one run:

- It performs the changing of the font layout to (a subset of) TeX text font encoding with all those characters present, which can be found in `ADOBE` fonts.
- For certain accented characters, operations "behind the scene" are performed: Accents may be taken from the source font even though they would not appear as individual characters in the resulting font (such as Polish "ogonek"), or accented characters may be taken from the input ("raw") font (such as Polish suppressed ł).

Moreover, the accent over capital `C` should also go a bit higher, as we mentioned earlier. Better resulting appearance would be:

Č č

This was achieved by appending the adjustments input shown above by:

```
(CHARACTER O 203 (RIGHT D 0.75) (UP R 0.4))
(CHARACTER O 243 (RIGHT D 0.25))
```

After this correction, "háček" can become "háček".

Practical experience with a METAFONT design of native fonts for Czech and Slovak (derived from Computer Modern font family by Petr Novák [Novák89]) has shown that the accent placement and corrections to the shape of the accents required quite a huge amount of assistance from a professional typographer. That made the time required for that task extend to about half a year including the METAFONT code debugging. It can be expected that for professionally looking accented characters defined for various typefaces by `VF`-defined compositions, it would in general be a good idea to invite a professional typographer to assist with the accent placement fine-tuning.

### 3.3 Generated character definitions

When defining `VF` definitions of characters, it is possible to design intriguing combinations of source font characters and operations with them. Dimensions of the `VF` character are taken from the `TFM` as quantities independent of the actual definition of the virtual character. When processing `dvi` file with virtual characters, the `dvi` processor encloses the virtual definition by `push` and `pop`, and after printing the character, it moves the reference point to the right by the amount specified as the character width in `TFM`.

If an output `dvi` driver cannot handle `VF` fonts, it is necessary to expand virtual characters by some post-processor to the TEX, such as, e.g., `DVICOPY`. Here is where potential trouble starts: If the last character expanded before the final enclosing `pop` does not right-align with the desired next reference point position, `DVICOPY` generates an "invisible" rule performing horizontal move compensating for the difference. Older drivers do not understand it properly, and instead of the move they really generate a rather huge solid black vertical rule.

factor set by design units).

Accented font from Table 1 shows the effect of default accent placement. This usually leads to a satisfactorily looking result, but for more professionally looking accented font some corrections are usually needed. I will comment on a few modifications involving Czech letters so that the character of the modifications became more apparent. First, there are several cases when the “háček” accent (ˇ) transforms itself into an apostrophe being put after the letter – this is the case of the letters `L`, `d`, `l`, and `t`. The amount of adjustment in those cases strongly depends on the particular font design, in this case we could try the following:

```
(DESIGNUNITS R 15)
(CHARACTER O 211 (LEFT R 3))
(CHARACTER O 264 (LEFT R 2.5) (UP R 2))
(CHARACTER O 244 (LEFT R 1))
(CHARACTER O 251 (LEFT R 1))
```

The result of these changes can be seen in the following table:

| | *default position* | *adjusted position* |
|---|---|---|
| `'211` | L’ | Ľ |
| `'244` | d’ | ď |
| `'251` | l’ | ľ |
| `'264` | t’ | ť |

Another example of a character where accent placement should be corrected is `C`, both upper and lower case. Because of the shape of the core letter, centered accent placement makes the “háček” over them to slide to the left:

Č č

For better appearance, both of the accents should be shifted to the right.

any emulation of genuine fonts with the Cork encoding scheme, but rather as just a conservative extension of the underlying font with TEX standard encoding: that scheme is also a standard, after all. This allows to consider any TEX standard font on input, not just Computer Modern, which is also why `ACCENTS` was conceived for.

The basic encoding scheme used by `ACCENTS` on the output, is therefore the TEX text font encoding for characters 0–127, and the extended TEX font encoding (or rather a subset of it) for characters 128–255; see Table 1.

In fact, at the time of writing this paper it is planned to add an additional *configuration* input to `ACCENTS` in order to allow for completely general re-arrangement of the output font layout, and for the specification which characters are made from which combinations particular core characters and one of the 13 accents built in `ACCENTS`. One such a re-arrangement is already performed by the program: `ACCENTS` can recognize a `TFM` with `ADOBE` standard encoding ("raw" version, using the terminology of `dvips` – see Table 2), and to re-arrange it into a subset of TEX text font encoding, see Table 3. Some of the additional `ADOBE` layout characters disappear as a result of this mapping, but from TEX they would still be accessible using the original "raw" `TFM`.

The possibility of selecting other font encoding scheme seems to be important also for generating specialized fonts for a particular language. The extended TEX font encoding may be suitable for exchanging texts internationally and for archiving purposes. For a practical TEX installation for a particular national language there may be an important obstacle to using a single encoding scheme encompassing all conceivable characters: The disk space needed for such fonts may be just too much on a PC-sized installation, and a specialized language-tailored font may prove itself to be a necessity for regular day-to-day use.

### 3.2 Accent position adjustment input

The `ACCENTS` processor allows for a limited control over details of accent placement in order to allow for precise fine-tuning of their respective position. The user-supplied auxiliary adjustment input consists of property lists for each character for which the automatic accent placement should be modified:

(`CHARACTER` *character_code adjustment_lists*)

Here *adjustment_lists* specify direction (`UP`, `DOWN`, `LEFT`, `RIGHT`), and distance in terms of design size of the font (with the possibility of modifying it by a

resulting virtual font.

Particular features of the `ACCENTS` processor are discussed in subsequent paragraphs.

### 3.1 Virtual font layout

In late 1990, TUG Multilingual Activities group designed the *Extended TEX Font Encoding* scheme, cf. [Ferguson90]. This encoding not only places 128 additional accented characters into the upper part (above 127) of the character table, but it also makes a few changes in the lower part of the table with respect to the TEX text font encoding as introduced by Donald Knuth: First 32 characters are devoted to accents, ligatures, and quotation marks, positions 32 – 127 follow exactly the ASCII characters and ASCII layout without any exception.

While the introduction of the new extended TEX font encoding may be suitable for a new font design, there are at least two reasons why changes in the lower part of the encoding scheme are not suitable as the basis for the virtual font layout:

- Some of the characters from the extended scheme are not present at all in the basic font layout that has to be used as the starting point. Some of those present must be re-designed: the English *opening* quotes |“| cannot be directly taken as German (and Czech, for that matter) *closing* quotes |“|, because in Computer Modern they are asymetric with respect to the box they are enclosed in.
- From the user's point of view, there seems to be virtue in font extensions being *conservative*, and not modifying their common parts. In particular, if families of fonts preserve characters accessible directly from the input text, changes from one font type to another can be made more easily – by just changing the name of the font in its `\font` definition, without the need for changing macros which provide the context for this font.

When the author started to think about the design of the `ACCENTS` processor, he sort of automatically thought of the only candidate for the underlying encoding for it as being the Cork scheme from [Ferguson90] – *some* standard is better than none. After thinking it over, hovever, he decided not to consider accented virtual fonts generated by `ACCENTS` as

use. (For Czech and Slovak, for example, the upper edge of accents over characters without ascenders is aligned to the same height. This puts umlaut rather high for German, where umlaut is the only accent in regular use, and hence there's no need to compare its placement with anything else. Another example may be accents over capital letters: In both Czech and Slovak, they can never be omitted even in texts in just capital letters, and there is no transliteration for them. The fact that they present an unseparable part of the letter accounts for their actual placement higher than in French or German.) Should minor modifications in accent placement be needed, virtual definitions are more suitable for maintaining such a "family" of typeface modifications, all within the same basic encoding scheme layout.

- Some older `dvi` device drivers may not be able to handle new eight-bit fonts. While it is easy to switch from pre-3.0 TEX to 3.0 or higher, changing the device driver (or screen previewer, for that matter) may be if not a real pain, than at least considerably more complicated, as far as modifications of drivers for every single device must potentially be made. For virtual fonts, there are post-processors available which perform virtual character expansion in a `dvi` file (e.g., `DVICOPY`, or `dvi2dvi`), so that the resulting `dvi` file can be processed by drivers not capable of handling either virtual fonts or eight-bit fonts.

## 3 Virtual font generator for accented letters

Under the assumption of a fixed character encoding of the basic (English) font, there is clearly a uniform way of composing core characters with the accents. This can be made mechanical by designing a virtual font generator. This is the purpose of the `ACCENTS` processor developed by the present author.

The `ACCENTS` virtual font generator needs to be able to perform character compositions to the same extent as TEX the program. Hence, the same *input* information about the source font is needed, that is the corresponding `TFM` font metric file. In order to compensate for the necessary fine-tuning of the relative position of the core character and the accent, additional *auxiliary input* may be present, containing differences in accent placement relative to the default positioning using algorithms based on plain TEX macros.

The *output* consists of a `TFM` and an associated `VF` files, defining the

also the accented characters as characters of their own. The other possibility comes from the fact that in overwhelming majority of European languages, the additional characters are made up from a core character taken from the English font, plus one of the accent characters already present in the standard TEX text encoding scheme. This means that a simpler device than full METAFONT definitions is actually needed: just definitions of accented characters as compositions of basic characters. Virtual fonts offer this possibility, with each character in them being defined as simple `dvi` program generating the appearance of the character needed.

## 2 New definition versus virtual font

New METAFONT definition of extended fonts may be seen as the ultimate solution allowing for proper typographical design of every "single core character – accent" combination, together with inclusion of also those characters which cannot be obtained as simple character superpositions. It also allows for taking into account the particular shape of the characters, which may influence the actual design of the result. Example of this may be placement of accents to optical center of the core character, not just the geometrical one taken from the font metric properties.

Virtual font definition presents a "poor man's" solution in this context. There are, however, several reasons why it is still worth considering:

- Virtual font definitions can also be used for fonts, for which no METAFONT definition is available, such as the PostScript fonts. Even for fonts defined in METAFONT, it may be considerably simpler to present definitions of accented characters as merely superpositions of existing ones than to redefine METAFONT descriptions.
- Virtual definitions of 256-character fonts are more concise than the full bitmaps generated from METAFONT descriptions. This may play considerable rôle on a PC installation of TEX.
- Although there is already a definition of an extended encoding containing characters from nearly all European languages, it is not obvious whether there is really a single one design of a unique set of accented characters which could be used for all the languages. In particular, the actual placement of accents in various languages seem to depend on typografic rules which depend on the actual language in

Cahiers GUTenberg n ° 10 — Sepembre 91

# Automatic generation of virtual fonts with accented letters for TEX

Jiří ZLATUŠKA

*Institute of Computer Science*
*Masaryk University*
*Burešova 20, 611 77 Brno*
*Czechoslovakia*
zlatuska@cspuni12.bitnet

**Abstract**

This paper presents an approach towards deriving fonts with accented letters for European languages using virtual fonts as an alternative to the development of genuine new fonts in the METAFONT. The ACCENTS processor is presented as a tool for mechanization of the process by enabling automatic generation of accented font layout and the virtual font definition from the TFM file of the source font in the TEX text encoding, and from an auxiliary input containing corrections of accent placement for specific characters.

**Key words:** fonts for European languages, virtual fonts, TEX text layout, ADOBE standard layout.

**Acknowledgment:** This work has been made possible in part thanks to a hardware support from the Czechoslovak Charter'77 Foundation.

## 1 Introduction

With TEX 3.0, Donald Knuth introduced support for 8 bit fonts and for virtual fonts (VF). This makes it possible to think of developing fonts for European languages involving accented letters in a better way than by merely using the built-in \accent primitive, or the plain TEX macros derived from it – the use of 8 bit character encoding is quite essential for also using language-specific hyphenation patterns.

Aside from the problem of choosing a particular 256 character table suitable for the alphabets with accented letters, there are still two basic options how to prepare fonts for European languages. First, it is possible to extend the METAFONT definitions of existing fonts in order to incorporate