

---

## The Program `a2ac` — Font Handling on the PostScript Level

Petr Olšák

### Abstract

In this article, the `a2ac` program written by the author is described. The program enables the use of PostScript fonts while typesetting texts in which accented letters are used. The font doesn't need to contain the complete alphabet of a given language; merely the presence of the accents themselves (and not all of the accented characters) is sufficient. The configuration files of the `a2ac` program are independent of the PostScript font encoding as well as of the typesetting system encoding. The program may be used to prepare a font for any typesetting system, but only  $\TeX$  was tested. The `a2ac` package is available on CTAN including the source in C and the documentation. It can serve as an alternative to the `fontinst` program by Alan Jeffrey. The program was tested on the operating systems SUN OS, Linux, and DOS.

— \* —

### 1 Introduction

The program `a2ac` (Afm to Afm and add Composites) works with the `afm` file (Adobe Font Metrics) as its input and creates a new modified `afm` file as its output. The program reads a so-called “description file” during its work. The changes we want to make are defined in this file. The program adds the

```
NC Ecaron 2 ; PCC E 0 0 ; PAT caron 0 Carontop ;
NC Eacute 2 ; PCC E 0 0 ; PAT acute Acuteshift Acutetop ;
NK (Ecaron,Eacute) : E
NC ecaron 2 ; PCC e 0 0 ; PAC caron 0 0 ;
NC eacute 2 ; PCC e 0 0 ; PAC acute acuteshift vshift ;
NK (ecaron,eacute) : e
RK (P,T,V) ecaron 0
RK (P,T) eacute 0
```

In this example we define new letters Ě, Ě́, ě and é. First, `Ecaron` and `Eacute` are defined as composite letters (NC lines), then new kerns are made for `Ecaron` and `Eacute`. The kerns are the same as the kerns for the letter E (NK line). Next, `ecaron` and `eacute` are defined as composite letters with the same kern information as the letter e (NC lines and NK line). Last, some exceptions for the kerns are specified. The pairs Pě, Tě and Vě will have no kern, in contrast to the pairs Pe, Te and Ve, where

full description of new composites (accented letters) including kern pairs, etc., to the output `afm` file.

The `afm` file describes information about the letters using the symbolic name for each character (`Aacute` is the A with an acute accent, for example). Each character is described through one of two possible ways. Either the character name is bound to a definite encoding position and to a PostScript procedure for rendering the image of the character, or the character is described as a so-called “composite character”. In this case the encoding position of the character is set to be `-1` and the description of how to make the character from elements is stored in `afm`. The elements are usually characters described as mentioned previously and only their symbolic names (not their encoding positions) are used.

All desired new composite characters are defined in the description file. Since only the symbolic names are used, the description file is totally independent of the encoding of the PostScript font and of the encoding used by the typesetting system.

The program already has nonzero intelligence when it reads the description file. You can declare and use so-called “variables”. You can write metric and composite information in the form of simple expressions. You can add new kern pairs using the information from already existing similar kern pairs (with exceptions being possible).

An example of the description file format is shown below.

negative kerns are usually defined. The same exceptions are specified for the pairs Pě and Tě.

For more information about the description file format see the `a2ac-eng.doc` file, which is included in the `a2ac` package.

Ligature information is not defined in the description file. We will show that it essentially doesn't matter. Ligatures are described in the input `afm` file at the end of lines with the `C` prefix as is shown in our following example:

```
C 102 ; WX 333 ; N f ; B 20 0 383 683 ; L i fi ; L l fl ;
```

The `a2ac` program just copies the ligature information to the output `afm` file. This is sufficient for generating the ligtables which are used by the typesetting systems. For example, the `afm2tfm` program reads this information and generates the proper data

```
% LIGKERN hyphen hyphen =: endash ; endash hyphen =: emdash ;
```

## 2 Font preparation for T<sub>E</sub>X

After `a2ac` is used, preparation of the font for T<sub>E</sub>X continues on in the standard way (as for the English language). You can use the `afm2tfm` program, which reads the converted `afm` file and the encoding information file `enc` to match the internal T<sub>E</sub>X encoding. The result is a T<sub>E</sub>X virtual font which includes two kinds of information: the information about re-encoding from the internal T<sub>E</sub>X encoding to the raw PostScript font encoding (as was specified in the `enc` file) and the information about building the accented letters from the elements (as was specified in the “rebuilt” `afm` file).

Let us show an example of the PostScript font preparation for T<sub>E</sub>X for the Czech and Slovak languages. First, we run the `a2ac` program. The command line looks like this:

```
a2ac input.afm desc.tab output.afm
```

The first parameter is the name of the input `afm` file, the second one is the name of the description file and the third one is the name of the output `afm` file. The extensions (`.afm` and `.tab`) are obligatory — the program has no algorithm for adding the extensions automatically.

The command to run the `a2ac` program is usually part of a script or a batch file. For example, the UNIX script to convert the font from the `afm` format to the `tfm` format may look like this:

```
a2ac $1.afm cscorr.tab c$2.afm
afm2tfm c$2.afm -t xl2.enc -v c$2 r$2
vptovf c$2.vpl c$2.vf c$2.tfm
```

The name of this script is `mkfnt`, say. Then the usage is simple:

```
mkfnt Times-Roman ptmr
```

The file `cscorr.tab` is included in the `a2ac` package. The file includes definitions of composites for Czech and Slovak accented letters (i.e., the output font metric includes Czech and Slovak letters and the proper kerns for these letters). The input font has to contain the standard set of 26 Latin letters (lower- and uppercase) and all the accents necessary for the composite letters, which means the acute, caron, ring, quoteright, dieresis, and circumflex accents. If the Adobe font in `StandardEncoding`

in the `ligtable` of the `tfm` format. New ligatures for T<sub>E</sub>X can be defined through the `enc` files. For instance, the LIGKERN information below is present in the file `xl2.enc` and defines new ligatures for the dash characters “--” and “---”.

is used, then this requirement is satisfied. The output `afm` file then specifies all characters (plain or composite) of the Czech and Slovak alphabets and the new kern information.

The description file `cscorr.tab` was used without changes for all standard PostScript fonts (32 fonts installed in all RIPS) with very good results. Of course, if you want, you can rearrange the `cscorr.tab` file for your new font to obtain the best result.

If the above UNIX script is used, the T<sub>E</sub>X font suitable for Czech and Slovak texts is prepared in the `CS`-font encoding, which is an extension of the Computer Modern font encoding (see Section “The T<sub>E</sub>X Encoding”).

The output of the script is `c*.tfm` (for T<sub>E</sub>X input), `r*.tfm` and `c*.vf` (for `dvips` input). The only work we must do is place these files into the appropriate directories (or rearrange the script to make this work automatically) and add one line to the configuration file `psfonts.map` of the `dvips` driver. The line looks like this:

```
rfont PostScript-Font
```

If the PostScript font is not installed in the PostScript RIP of the output device, we have to store the `pfb` (`pfa`) format of the PostScript font in our computer. Then the line in `psfont.map` looks like this:

```
rfont PostScript-Font </path/to/file.pfb
```

or (for DOS):

```
rfont PostScript-Font <d:\path\to\file.pfb
```

The new font can be loaded into T<sub>E</sub>X using the `\font` command, as shown below. From that point onward, the font in question is available and text using the same encoding as the font can be conveniently typeset by T<sub>E</sub>X (and printed with `dvips`).

```
\font\newfont=cfont
{\newfont the eight-bit text in the same
 encoding as defined in xl2.enc
 can be used and the new font
 will be shown.}
```

### 3 The $\TeX$ encoding

The input text encoding for  $\TeX$  doesn't need to be the same as the internal  $\TeX$  encoding. The re-encoding can be performed by the `tcp` table of `emTeX`, by changes to the `xord/xchr` vectors in the `tex.ch` file ( $\TeX$ 's WEB source hacking) or by setting active characters. The `enc` file describes exactly the same encoding as the internal  $\TeX$  encoding into which the hyphenation patterns have been read.

In the Czech Republic and the Slovak Republic, the so-called  $\mathcal{CS}$ -encoding (the encoding of the  $\mathcal{CS}$ -fonts) is used very often as the internal  $\TeX$  encoding. The  $\mathcal{CS}$ -encoding is an extension of the Computer Modern text font encoding and follows the ISO-8859-2 layout for the Czech and Slovak letters. The  $\mathcal{CS}$ -fonts are included in the widely available  $\TeX$  installation package with the name  $\mathcal{CS}\TeX$ .

The advantages of the  $\mathcal{CS}$ -encoding (in comparison to the T1 encoding) from our point of view are the following:

- The fonts include the letters from the Czech and Slovak alphabets only. Font generation from the METAFONT sources is faster and the `pk` files are smaller. Exceptions (non-Czech/Slovak letters in names, for example) are accessible by plain  $\TeX$  macros or by the `\accent` primitive.
- The  $\mathcal{CS}$ -font encoding is the conservative extension of the Computer Modern text fonts. Therefore,  $\mathcal{CS}$ -fonts can replace CM text fonts in the installation packages without problems and with backward compatibility. This feature saves space on the disk. No complicated macro redefinitions are necessary. For example, the `csplain` format, which is included in the  $\mathcal{CS}\TeX$  package, gives the same results as the original plain  $\TeX$ . There are only three differences: (1) The new letters with codes  $\geq 128$  are introduced by `\catcode`, `\lccode`, `\uccode`; (2)  $\mathcal{CS}$ -fonts are loaded instead of CM text fonts; and (3) the Czech and Slovak hyphenation patterns are read in addition to the US English ones.
- The layout of the accented letters in the  $\mathcal{CS}$ -font is the subset of the ISO-8859-2 standard. Therefore, it is possible to use an eight-bit input Czech/Slovak text without re-encoding in the UNIX environment. The power is in simplicity! The `tcp` table (`emTeX`) is used in the DOS installation because there are many different layouts for the Czech/Slovak alphabet and there is no fixed standard in DOS (in contrast to the UNIX-like systems).
- Kerning and other aspects of the fonts are language-dependent. The multi-language font (T1 encoded, for example) is nonsense from the typographer's point of view. In addition, we can't rely on the results of the font hacking from abroad.

The Computer Modern text font encoding (and its conservative extensions) is font-type dependent. For example, two different characters are encoded to the same position depending on `\rm` versus `\tt` font (the 'fi' ligature versus downarrow, for example) or on `\rm` versus `\it` font (the dollar versus the sterling). This feature implies some problems.

Two files, `x12.enc` and `xt2.enc`, are included in the `a2ac` package. These files define extensions of the  $\mathcal{CS}$ -font encoding. The extensions include some characters which are not contained in the Czech and Slovak alphabets (i.e., in the  $\mathcal{CS}$ -font), but which are contained in the Adobe StandardEncoding material.

It is recommended to use `x12.enc` for `\rm` and `\it`-like fonts and `xt2.enc` for `\tt`-like fonts. The dollar is in position 36 under any circumstances and position number 132 was given to the sterling. If you load the PostScript `\it`-like font then you need to know that the sterling is not accessible by the plain  $\TeX$  `{\it\}` but you need to use a macro with `\char132` code.

Position number 32 is not defined in the `x12.enc` file because the cross accent for the Polish L and l is not included in the Adobe StandardEncoding. The L and l themselves are in positions 163 and 179, respectively. If we need to use these characters in plain  $\TeX$  with the PostScript font, the redefinition of the macros `\l` and `\L` is necessary.

Note that the three-letter ligatures are not included in the Adobe StandardEncoding. This doesn't cause any problems because the three-letter ligatures are not used in Czech and Slovak. If the font is available in the ExpertEncoding you can do some hacking at the `vp1` level to obtain these ligatures.

If you want to typeset math using the PostScript font, you need to edit the `vp1` file [1] to include the uppercase Greek characters taken usually from the Symbol font. Sorry, but it is not obvious how to typeset math with the PostScript font; some more hacking must be done—on the plain  $\TeX$  macro level as well as on the `\fontdimen` font information. The pretty "look" of math is a result of a font designer's work and a lot of  $\TeX$ -macro programming. "Its emphasis is on art and technology, as in the underlying Greek word" (see [2], Chapter 1.). Therefore, the pretty math is not accessible from the text PostScript font via any automatic process.

For more information about the  $\mathcal{C}\mathcal{S}$ -font encoding and its history see [3] and appendix F in [4].

#### 4 About the names of the accented letters

The names `dquoteright` and `tquoteright` should be used for the characters `d'` and `t'` in the description file `cscorr.tab`. However, we use the names `dcaron` and `tcaron` instead. The reason is that the uppercase alternatives of the characters `d'` and `t'` are  $\check{D}$  and  $\check{T}$ , i.e., `Dcaron` and `Tcaron`, respectively. The parameter `-V` (for `afm2tfm` to make the small caps variant of a font) does not work with the names `dquoteright` and `tquoteright`. We use `Lcaron` and `lcaron` instead of `Lquoteright` and `lquoteright` because the semantic for these accents is same as for `Dcaron`, `dcaron`, `Tcaron` and `tcaron`.

#### References

- [1] Donald Knuth: Virtual fonts, more fun for grand wizards. *TUGboat* 11(1):13–23, April 1990.
- [2] Donald Knuth: *The T<sub>E</sub>Xbook*, vol. A of *Computers & Typesetting*. Addison-Wesley, Reading, MA, 1986.
- [3] Petr Olšák: Úvaha o fontech v  $\mathcal{C}\mathcal{S}\text{T}_{E}\text{X}$  [A reflection about fonts in  $\mathcal{C}\mathcal{S}\text{T}_{E}\text{X}$ ], *T<sub>E</sub>Xbulletin* (of  $\mathcal{C}\mathcal{S}\text{TUG}$ ) 3/93 (121–131).
- [4] Petr Olšák: *Typografický systém T<sub>E</sub>X* [Typesetting System T<sub>E</sub>X],  $\mathcal{C}\mathcal{S}\text{TUG}$  1995, 270 pages.

◇ Petr Olšák  
 Department of Mathematics  
 Czech Technical University in  
 Prague  
 Czech Republic  
 Email: [olsak@math.feld.cvut.cz](mailto:olsak@math.feld.cvut.cz)

— \* — \* — \* — \* — \* — \* — \* — \* — \* — \* — \* — \* —

#### Note added by production editor

In order to better understand how the above procedure works, I decided to download the program from `ftp.tex.ac.uk` in the directory `fonts/utilities/a2ac` below the CTAN root. I also got the `.afm` files for the PostScript fonts from the directory `fonts/psfonts/adobe`.

I now was all set to compile the `a2ac.c` program, and start to make the  $\mathcal{C}\mathcal{S}$ -encoded files using the script `mkfnt`. It essentially runs the program `afm2tfm` that comes with the `dvips` distribution and creates `.tfm` and `.vf` files for T<sub>E</sub>X and `dvips` according to the desired encoding (see the Section “Font Preparation for T<sub>E</sub>X”). This was done for all “Standard” PostScript fonts. Figure 1 shows the font table for the font `NewCenturySchlbk-Roman`.

The righthand side of the table shows the characters specific to the Czech and Slovak languages. To obtain PostScript output from a `.dvi` file one more step has to be taken; namely, one has to tell the `dvips` program which file to read when a font-name is encountered. The easiest way of achieving this is to generate a new “config” file, since several can be specified with one `dvips` execution. In this case, I generated a one-line `config.cs` file containing

```
p +pscs.map
```

This states that the contents of the file `pscs.map` has to be concatenated with the contents of previously declared map files, in particular the default

`psfonts.map`. The file `pscs.map` itself contains one line for each of the fonts considered: and starts like

```
rpagd AvantGarde-Demi
rpagdo AvantGarde-DemiOblique
rpagk AvantGarde-Book
rpagko AvantGarde-BookOblique
...
```

as explained in the text of the article.

As a further exercise, I decided it would be interesting to make this encoding available with L<sub>A</sub>T<sub>E</sub>X, so I started by defining an encoding L2 in the file `L2enc.def` (See Chapter 7 of *The L<sub>A</sub>T<sub>E</sub>X Companion*, by Goossens et al., Addison-Wesley, 1994 and the file `fntguide.tex` “L<sub>A</sub>T<sub>E</sub>X 2<sub>ε</sub> font selection”, which is distributed electronically and comes with the standard L<sub>A</sub>T<sub>E</sub>X 2<sub>ε</sub> distribution.). This file contains merely:

```
\ProvidesFile{L2enc.def}%
      [1995/09/22 CS Encoding]
\DeclareFontEncoding{L2}{-}{-}
\DeclareFontSubstitution{L2}{ptm}{m}{n}
```

Next, one has to make *font definition* `.fd` files for each of the fonts one wants to use, e.g., for the Times-Roman family one would define a file `L2ptm.fd`, where `ptm` is a “short” name chosen in the framework of the font-selection scheme of Karl Berry (Version 2), which can represent all possible font names within the ISO-9660 (and DOS)-compatible eight characters limit for file names. Below the font definition file (`L2pag.fd`) for the L2 encoding of the AvantGarde font family is shown.

	00	10	20	30	40	50	60	70	80	90	AO	BO	CO	DO	EO	FO
0		ı		0	@	P	'	p	...	~	ı	°	Ř	Ā	ř	
1			!	1	A	Q	a	q	†	¢	Ê	â	Á	Ā	á	
2		`	"	2	B	R	b	r	‡	/	˘	˙	Â	Ň	â	ň
3		˘	#	3	C	S	c	s	•	¥	Ł	ł	Ů	Ó	ª	ó
4		˘	\$	4	D	T	d	t	£	'	◊	˘	Ä	Ö	ä	ö
5		˘	%	5	E	U	e	u	¶	·	Ł	ł	Ĺ	Ń	Í	
6		˘	&	6	F	V	f	v	"	,	f	ÿ	İ	Ö	Ö	ö
7		˘	'	7	G	W	g	w	<	°	§	˘	ı	Ÿ	õ	
8		˙	(	8	H	X	h	x	>	ˆ	ˆ	ˆ	Č	Ř	č	ř
9		β	)	9	I	Y	i	y	\	ê	Š	š	É	Ů	é	ů
A		æ	*	:	J	Z	j	z	^	ò	è	Ò	È	Ú		ú
B		œ	+	:	K	l	k	-	_	û	Ť	t	Ě	ù	ě	
C	fi	ø	,	ı	L	“	l	—	{	-	◊	İ	Ě	Ů	ě	ů
D	fl	Æ	-	=	M	l	m	"	%o	˙	˙	˙	Í	Ý	í	ý
E		Œ	.	¿	N	^	n	˘		«	Ž	ž	Ī	ň	ı	„
F		Ø	/	?	O	˘	o	˘	}	»	ā	Ů	Ď	β	d'	“

Figure 1: The layout of the  $\mathcal{CS}$  encoded font

```

\ProvidesFile{L2pag.fd}[1995/09/22 L2 (CS) encoding for pag, MG]
\DeclareFontFamily{L2}{pag}{}
\DeclareFontShape{L2}{pag}{db}{n}{<->cpagd}{}
\DeclareFontShape{L2}{pag}{db}{sl}{<->cpagdo}{}
\DeclareFontShape{L2}{pag}{m}{n}{<->cpagk}{}
\DeclareFontShape{L2}{pag}{m}{sl}{<->cpagko}{}

\DeclareFontShape{L2}{pag}{b}{n}{<->ssub * pag/db/n}{}
\DeclareFontShape{L2}{pag}{bx}{n}{<->ssub * pag/b/n}{}
\DeclareFontShape{L2}{pag}{db}{sc}{<->ssub * pag/db/n}{}
....

```

The first line (`\ProvidesFile`) declares the name and version of the file to the  $\text{\LaTeX}$  system (as it will appear in the `.log` and on the console). Then the family name `pag` for the L2 encoding is declared. This is followed by four declarations for the series and shapes for which actual external files are available, namely:

```

pagd  AvantGarde-Demi
pagdo  AvantGarde-DemiOblique
pagk  AvantGarde-Book
pagko  AvantGarde-BookOblique

```

The remaining declarations are substitution rules for combinations of font parameters which are not available as separate instances.

Finally, with such specifications for all used fonts, one can take a text encoded according to the given  $\mathcal{CS}$ -encoding, and run it through  $\text{\LaTeX}$ . I have

taken the test file distributed by Petr, and transformed it into  $\text{\LaTeX}$  to test the above setup. Figure 2 shows the  $\text{\LaTeX}$  source file. The command

```
dvips -E -Pcs L2test -o
```

yields the output shown in Figure 3, where the same Czech text has been typeset in Times-Roman, Times-Bold, Times-Roman-SmallCaps, AvantGarde-BookOblique, and finally AvantGarde-DemiOblique, showing clearly the advantages of this system.

◊ Michel Goossens  
 CERN, CN Division  
 CH-1211 Geneva 23  
 Switzerland  
 Email: goossens@cern.ch

```

\documentclass[a4paper]{article}
\usepackage[L2]{fontenc}
\renewcommand{\rmdefault}{ptm} % Text font family is Times
\renewcommand{\sfdefault}{pag} % San-serif font family is AvantGarde
\renewcommand{\ttdefault}{pcr} % Teletype font family is Courier
\pagestyle{empty}
\begin{document}
... % text will be in Times-Roman
\bfseries ... % text will be in Times-Bold
\mdseries\scshape ... % text will be in Times-Roman-SmallCaps
\sffamily\bfseries\itshape ... % text will be in AvantGarde-DemiOblique
\mdseries\itshape ... % text will be in AvantGarde-BookOblique
\end{document}

```

Figure 2: L<sup>A</sup>T<sub>E</sub>X source of  $\mathcal{CS}$  document

Tento text je napsán PostScriptovým fontem „*Times-Roman*“ za použití T<sub>E</sub>Xovské metriky `cptmr`. Metrika fontu v české/slovenské variantě byla připravena konvertorem `a2ac` s následným použitím programu `afm2tfm`. Je potřeba si všimnout méně obvyklých slov, jako je slovo šťastný (písmeno *t* uprostřed slova), nebo třeba slovenské kolko. Věřme, že výsledný text půjde nejen přečíst, ale bude i typograficky pokud možno v pořádku. Také je třeba „sledovat chování“ našich „uvozovek“.

Á á Ä ä Č č Ď ě É é Ě ě Í í Ĺ ĺ Ľ ľ Ń ň Ó ó Ô ô Ö ö Ř ř Ř ř Š š Ť ť Ú ú Ů ů Ū ů Ý ý Ž ž.

Tento text je napsán PostScriptovým fontem „*Times-Bold*“ za použití T<sub>E</sub>Xovské metriky `cptmb`. Metrika fontu v české/slovenské variantě byla připravena konvertorem `a2ac` s následným použitím programu `afm2tfm`. Je potřeba si všimnout méně obvyklých slov, jako je slovo šťastný (písmeno *t* uprostřed slova), nebo třeba slovenské kolko. Věřme, že výsledný text půjde nejen přečíst, ale bude i typograficky pokud možno v pořádku. Také je třeba „sledovat chování“ našich „uvozovek“.

Á á Ä ä Č č Ď ě É é Ě ě Í í Ĺ ĺ Ľ ľ Ń ň Ó ó Ô ô Ö ö Ř ř Ř ř Š š Ť ť Ú ú Ů ů Ū ů Ý ý Ž ž.

TENTO TEXT JE NAPSÁN POSTSCRIPTOVÝM FONTEM „*Times-Roman-SmallCaps*“ ZA POUŽITÍ T<sub>E</sub>XOVSKÉ METRIKY `cptmrc`. METRIKA FONTU V ČESKÉ/SLOVENSKÉ VARIANTĚ BYLA PŘIPRAVENA KONVERTOREM `a2ac` S NÁSLEDNÝM POUŽITÍM PROGRAMU `afm2tfm`. JE POTŘEBA SI VŠÍMAT MÉNĚ OBVYKLÝCH SLOV, JAKO JE SLOVO ŠTASTNÝ (PÍSMENO *t* UPROSTŘED SLOVA), NEBO TŘEBA SLOVENSKÉ KOLKO. VĚRME, ŽE VÝSLEDNÝ TEXT PŮJDE NEJEN PŘEČÍST, ALE BUDE I TYPOGRAFICKY POKUD MOŽNO V POŘÁDKU. TAKÉ JE TŘEBA „SLEDOVAT CHOVÁNÍ“ NAŠICH „UVOZOVEK“.

Á á Ä ä Č č Ď ě É é Ě ě Í í Ĺ ĺ Ľ ľ Ń ň Ó ó Ô ô Ö ö Ř ř Ř ř Š š Ť ť Ú ú Ů ů Ū ů Ý ý Ž ž.

Tento text je napsán PostScriptovým fontem „*AvantGarde-BookOblique*“ za použití T<sub>E</sub>Xovské metriky `cpagdo`. Metrika fontu v české/slovenské variantě byla připravena konvertorem `a2ac` s následným použitím programu `afm2tfm`. Je potřeba si všimnout méně obvyklých slov, jako je slovo šťastný (písmeno *t* uprostřed slova), nebo třeba slovenské kolko. Věřme, že výsledný text půjde nejen přečíst, ale bude i typograficky pokud možno v pořádku. Také je třeba „sledovat chování“ našich „uvozovek“.

Á á Ä ä Č č Ď ě É é Ě ě Í í Ĺ ĺ Ľ ľ Ń ň Ó ó Ô ô Ö ö Ř ř Ř ř Š š Ť ť Ú ú Ů ů Ū ů Ý ý Ž ž.

Tento text je napsán PostScriptovým fontem „*AvantGarde-DemiOblique*“ za použití T<sub>E</sub>Xovské metriky `cpagko`. Metrika fontu v české/slovenské variantě byla připravena konvertorem `a2ac` s následným použitím programu `afm2tfm`. Je potřeba si všimnout méně obvyklých slov, jako je slovo šťastný (písmeno *t* uprostřed slova), nebo třeba slovenské kolko. Věřme, že výsledný text půjde nejen přečíst, ale bude i typograficky pokud možno v pořádku. Také je třeba „sledovat chování“ našich „uvozovek“.

Á á Ä ä Č č Ď ě É é Ě ě Í í Ĺ ĺ Ľ ľ Ń ň Ó ó Ô ô Ö ö Ř ř Ř ř Š š Ť ť Ú ú Ů ů Ū ů Ý ý Ž ž.

Figure 3: Result of L<sup>A</sup>T<sub>E</sub>X run on document shown in Fig. 2