The `ogonek` package*

Janusz Stanisław Bień

94/12/21

Abstract

This LaTeX 2ε package provides a command to typeset letters with the ogonek diacritic mark; they are used in Polish and Lithuanian. The command is named `\k` in accordance with the recommendation of the Technical Working Group on Multiple Language Coordination of the TeX Users Group. The principal purpose of the command is to provide the high quality ogonek with CM fonts, although for Polish the best results are obtained with the special Polish PL fonts; the command can be also used with DC fonts.

1 Introduction

The ogonek diacritic mark (`\k`) is absent in the original Computer Modern font ([9]), probably because it was not needed for Donald Knuth's Art of Computer Programming. The ogonek was included in the extended TeX layout agreed in 1990 at the TeX conference in Cork in Ireland and therefore often called simply the Cork layout; however, there was still no standard command to typeset it. This was remedied in 1992, when the TeX Users Group Technical Working Group on Multiple Language Coordination WG-92-03[1] recommended a set of TeX conventions concerning languages (cf. [5]). In particular, the command names were proposed for typesetting letters and accents introduced in the extended layout; the command `\k` was assigned to the ogonek and the name justified as the last letter of the word *ogonek*[2]

In [5] WG-92-3 proposed also a set of two-letter names for the language-switching macros. We use the two names from this list (but without the preceeding backslash) as the option names in our package: PL for Polish and LT for Lithuanian.

The lack of a standard way to typeset ogonek with Computer Modern fonts and its predecessors (including AM, i.e. Almost Computern Modern fonts) was from the very beginning a very serious obstacle for high quality typesetting of Polish texts. Several various techniques were developed independently to circumvent this problem; in the present package we use the method developed at the Faculty of

*Version v0.51 dated 95/07/17.

[1] The group was described in [4]

[2] Actually Jörg Knappen wrote in [8] that `\k` stands also for the first letter of the Scandinavian kvist. It can be viewed also as the first letter of the German word Krummhaken

Mathematics, Informatics and Mechanics of the Warsaw University and used in LaTeX styles plfonts and plhb.[3]

The primary problem was to find a CM character which bears sufficient ressemblance to ogonek. Several characters (including e.g. comma) were tried till 1988, when Jerzy Ryll suggested to use `\lhook` (left hook) symbol available in Plain TeX as a part of the `\hookrightarrow` (↪) symbol; this is the character `'54` in math italics fonts. Ryll's idea was described in the note [1] and Janusz S. Bień's `pl.sty` style using this technique was sent to the TeXline editor to be included in the Aston TeX archive; unfortunately, it seems that it never managed to get there. The idea was also presented in a paper written in Polish in 1988, which however appeared much later ([2]).

The remaining problem was to achieve proper positioning of the left hook character with the appropriate letters for every fonts size and shape; as ogonek accompanies such different letters as a, A, e and E, this was not an easy task. At first it was done simply by hand, as in Janusz S. Bień's `plfonts.tex` file loaded during the LaTeX format generation. The credit for solving this problem is due to Leszek Holenderski, who in 1989 created his `plfonts.sty`, which patched the standard LaTeX font switching mechanism with the code for adjusting the placing of ogonek. We use his code here without any substantial changes.[4]

In the extended TeX layout used at present practically only in Norbert Schwarz's DC fonts (cf. [6], [7]) but envisaged as the future TeX standard and therefore recommended for LaTeX 2_ε users the slots are assigned for both Polish letters with ogonek and the ogonek itself; typeseting all Polish letters and some Lithuanian ones causes therefore no problem and requires only referencing the appropriate characters; the remaining Lithuanian characters have to be typeset using by the composition of the appropriate chcaracters (the `\accent` primitive cannot be used for this purpose because it placed the accent over the letter).

The primary problem with the extended TeX font layout was (at to some measure still is) its incompatibility with the standard CM layouts, which for many users makes the migration to the new layout prohibitively difficult. For many applications a good solution was a mixed layout, with the lower part (character codes from 0 to 127) fully compatible with CM fonts and the higher part more or less compatible with the Cork layout. We will call this layout Cork-extended CM layout[5].

The PL fonts, developed by Bogusław Jackowski and Marek Ryćko with some advice of a professional typographer Roman Tomaszewski and included in the MeX distribution[6], are a special case of Cork-like extended Computer Modern fonts—in the higher part they contain Polish letters with ogonek placed in the same slots as in the Cork layout; however, they contain also the Polish double opening quote

[3]Thanks to the contribution of Piotr Filip Sawicki, the support of these styles is a standard feature of AUC TeX, a sophisticated (La)TeX environment for Emacs, since the release 9.0 of May 1994.

[4]Bień's notes say that he started to use Ryll's technique on 22nd June 1988 and created a version of Holenderski's style on 17th October 1999 (the version was called `plhb.sty`, where `hb` stands for Holenderski's style as modified by Bień and `pl` stands both for Polish and the earlier `pl` style; it used a different input convention than the original Holenderski's style)

[5]It seems to be little known that the layout should be coded in the TFM and PK files by means of the Metafont **font_coding_scheme** command; to the best of our knowledge, the only program which takes advantage of this fact is Eberhard Mattes' dvispell

[6]available e.g. from Comprehensive TeX Archive Network in the directory tex-archive/languages/polish

moved from its Cork position in the lower part to the higher part of the font. This layout can be called PL-extended CM fonts[7]

The PL fonts provide the best quality for Polish texts; however, for those Lithuanian letters with ogonek which do not coincide with Polish ones it is necessary to use the same technique as for CM fonts. In consequence, for Lithuanian texts the use of DC fonts is probably an optimal solution.

2 Usage

The package is loaded in the standard way with the `\usepackage{ogonek}` command.

As the fonts called by us the PL-extended CM fonts are not widely used, they do not have also a generally accepted symbol for their layout. Mariusz Olko in his preliminary version of polski package referes to them as OT1P, while Włodzimierz Bzyl in his LaMeXe uses the OT4 symbol. In consequence `ogonek` works with the following font encodings: OT1 (standard meaning) OT1P (PL fonts with Olko's package) OT4 (PL fonts with LaMeX2e and later versions of Olko's package) T1 (standard meaning)

The package accepts two language options:

PL only Polish letters with ogonek

LT Lithuanian letters — which subsume the Polish ones — with ogonek

Omitting the language option allows to use any letter with ogonek.

3 Hyphenation of words with ogonek accent

The full and correct hyphenation of words with ogonek (and other Polish letters) is possible with DC and PL fonts; details to be written later.

4 Implementation

Beware: comments in this section were written by Igor Moo.

4.1 Identification

We start the code with standard identification of the package

```
1 ⟨*style⟩
2 \NeedsTeXFormat{LaTeX2e}[95/06/01]
3
4 \ProvidesPackage{ogonek}[\filedate\space\fileversion\space
5      Provides macro `\string\k' for ogonek]
```

[7] At present (i.e. in all MeX releases including 1.5) a PL font have the **font_coding_scheme** identical with the CM font it is compatible with. For example, both plr10 and cmr10 have the coding scheme TeX text, pltt10 and cmtt10 TeX typewriter text etc. Dvispell users would appreciate very much if the PL fonts were distinguishable from CM fonts by the coding scheme field, which can be asigned such values as PL-extended TeX text, PL-extended TeX typewriter text etc.

4.2 Processing options

4.2.1 Encoding selection options

\ogonek@obsolete

In previous versions of ogonek options were present for selection of font encoding(s) used in a document. Now they are no longer needed since we try to guess what encodings are really used.

```
\newcommand\ogonek@obsolete[1]{%
  \PackageWarningNoLine{ogonek}{Option #1 is now obsolete \MessageBreak
     due to dynamic encodings testing}
  }
\DeclareOption{T1}{\ogonek@obsolete{T1}}
\DeclareOption{OT1}{\ogonek@obsolete{OT1}}
\DeclareOption{OT1P}{\ogonek@obsolete{OT1P}}
\DeclareOption{OT4}{\ogonek@obsolete{OT4}}
```

4.2.2 Language selection options

\@testogonekletter

Here we define macro that will be used below to test if a ogonked letter is 'legal'. Primarily we define it just to gobble it's argument.

If option PL is specified the macro is redefined to accept only Polish ogonked letters. Option LT redefines it to allow only Lithuanian letters.

If both options were specified all aAeEiIuU letters will be allowed since in LaTeX 2_ε options are processed in order of declaration and LT overwrites PL.

```
\let\@testogonekletter\@gobble
\DeclareOption{PL}{
  \def\@testogonekletter#1{%
    \ifx a#1\else    \ifx A#1\else
    \ifx e#1\else    \ifx E#1\else
      \PackageWarning{ogonek}%
{Unusual Polish letter #1 with ogonek}\fi
\fi \fi \fi
  }
}
\DeclareOption{LT}{
  \def\@testogonekletter#1{%
    \ifx a#1\else    \ifx A#1\else
    \ifx e#1\else    \ifx E#1\else
    \ifx i#1\else    \ifx I#1\else
    \ifx u#1\else    \ifx U#1\else
\PackageWarning{ogonek}%
       {Unusual Lithuanian letter #1 with ogonek}\fi
\fi \fi \fi \fi \fi \fi \fi
    }
}
```

Now we're ready to process the options

```
\ProcessOptions
```

4.3 Positioning of ogonek in old fonts

This comes from L. Holenderski's plfonts.sty. Positionig of ogonek for specific letters is tuned for 300dpi Computer Modern fonts, but works reasonably well

with other resolutions.

\sob Macro \sob positioning ogonek under a letter.

```
\dimendef\pl@left=0 \dimendef\pl@down=1
\dimendef\pl@right=2 \dimendef\pl@temp=3

% typeset 'ogonek' box
\def\sob#1#2#3#4#5{% parameters: letter and fractions hl,ho,vl,vo
\setbox0\hbox{#1}\setbox1\hbox{$_\mathchar'454$}\setbox2\hbox{p}%
\pl@right=#2\wd0 \advance\pl@right by-#3\wd1
\pl@down=#5\ht1 \advance\pl@down by-#4\ht0
\pl@left=\pl@right \advance\pl@left by\wd1
\pl@temp=-\pl@down \advance\pl@temp by\dp2 \dp1=\pl@temp
\leavevmode\kern\pl@right\lower\pl@down\box1\kern-\pl@left #1}
```

\aob Special positioning of ogonek for some letters
\Aob
\eob
\Eob
\iob
\Iob
\uob
\Uob

```
\def\aob{\sob a{.66}{.20}{0}{.90}}
\def\Aob{\sob A{.80}{.50}{0}{.90}}
\def\eob{\sob e{.50}{.35}{0}{.93}}
\def\Eob{\sob E{.60}{.35}{0}{.90}}
\def\iob{\sob i{.66}{.20}{0}{.90}}
\def\Iob{\sob I{.80}{.50}{0}{.90}}
\def\uob{\sob u{.66}{.20}{0}{.90}}
\def\Uob{\sob U{.60}{.35}{0}{.90}}
```

\@iIuUogonek
\@oldfontsogonek

Below we define macros producing ogonek in encodings OT4 (OT1P) (this needs special positioning of ogonek only for iIuU since for aAeE we have composities) and OT1. This could be done in a more LaTeXy way if we had something like \DeclareTextComposite allowing replacement to be macro not a single character. But we haven't.

```
\def\@iIuUogonek#1{%
\ifx i#1\iob\else
\ifx I#1\Iob\else
\ifx u#1\uob\else
\ifx U#1\Uob\else\sob {#1}{.50}{.35}{0}{.90}\fi
\fi \fi \fi
}
\def\@oldfontsogonek#1{%
\ifx a#1\aob\else
\ifx A#1\Aob\else
\ifx e#1\eob\else
\ifx E#1\Eob\else
\@iIuUogonek{#1}
\fi \fi \fi \fi
   }
```

4.4 Testing of encodings used in a document

This testing must be carried off when the document begins, since only then all used encodings are already known. We use \AtBeginDocument hook for that purpose. This will work unless some package loaded after ogonek has an idea to declare encodings 'at begin document' (I cannot think of any reason for that).

First my favourite hack for operations on names constructed with \csname:

```
\newcommand\n@me[2]{\expandafter#1\csname#2\endcsname}
```

You can not only say `\n@me\ifx{T@T1}\sth` but even `\n@me\show{a name}` or `\n@me\newcommand{and another}{...}` (sic!).

The testing really starts here. If an encoding `XXX` is known a macro with name `\T@XXX` is defined. In that way we check what encodings are in use.

For every encoding found we define `\k` to test if accentee is legal and put appropriate kind of ogonek.

```
\AtBeginDocument{%
```

We don't make any changes for T1, since all we need is defined by default.

```
  \n@me\ifx{T@T1}\relax \else \PackageInfo{ogonek}{T1 is known} \fi
```

For OT1 encoding we define ogonek as `\@oldfontsogonek`

```
  \n@me\ifx{T@OT1}\relax
  \else \PackageInfo{ogonek}{Defining ogonek for OT1}
    \DeclareTextCommand\k{OT1}[1]{%
       \@testogonekletter{#1}\@oldfontsogonek{#1}}
    \fi
```

For OT4 or OT1P `\k` won't know how to put ogonek under aAeE, but we have composities for that cases.

We are lucky that ogonek is always allowed under aAeE. Otherwise we would have to invent a way to incorporate test into composities.

```
  \n@me\ifx{T@OT4}\relax
  \else \PackageInfo{ogonek}{Defining ogonek for OT4}
    \DeclareTextCommand\k{OT4}[1]{%
       \@testogonekletter{#1}\@iIuUogonek{#1}}
    \DeclareTextComposite\k{OT4}{a}{"A1}
    \DeclareTextComposite\k{OT4}{A}{"81}
    \DeclareTextComposite\k{OT4}{e}{"A6}
    \DeclareTextComposite\k{OT4}{E}{"86}
    \fi
  \n@me\ifx{T@OT1P}\relax
  \else \PackageInfo{ogonek}{Defining ogonek for OT1P}
    \DeclareTextCommand\k{OT1P}[1]{%
       \@testogonekletter{#1}\@iIuUogonek{#1}}
    \DeclareTextComposite\k{OT1P}{a}{"A1}
    \DeclareTextComposite\k{OT1P}{A}{"81}
    \DeclareTextComposite\k{OT1P}{e}{"A6}
    \DeclareTextComposite\k{OT1P}{E}{"86}
    \fi
  }
```

And that's all.

```
\endinput
```

⟨/style⟩

References

[1] Janusz S. Bień. Polish Language and TEX. TEXline 8, January 1989, p. 2.

[2] Janusz S. Bień. Co to jest TEX. Available by anonymous FTP from ftp.mimuw.edu.pl in pub/users/jsbien/tex as cttex90.tar.Z or from LISTSERV@PLEARN.edu.pl as CTTEX90 PACKAGE.

[3] Michael J. Ferguson. Report on Multilingual Activities. TUGboat Vol. 11, No 4, November 1990, pp 514–516

[4] Michael J. Ferguson. The Technical Council. *TEX and TUG News* Vol. 1, No. 3, November 1992, pp 5–8.

[5] Yannis Haralambous. TEX Convention Concerning Languages. *TEX and TUG News* Vol. 1, No. 4, December 1992, pp 3–10.

[6] Yannis Haralambous. DC fonts—questions and answers (I). *TEX and TUG News* Vol. 1, No. 4, December 1992, pp 15–17.

[7] Yannis Haralambous. DC fonts—questions and answers (II). *TEX and TUG News* Vol. 2, No. 1, February 1993, pp 10–12.

[8] Jörg Knappen. Summary on Polish (La)TEX. Info-TeX@SHSU.edu and TeX-Euro@vm.urz.uni-heidelberg.de, 20th Novemebr 1992.

[9] Donald E. Knuth. Computer Modern Typefaces. Computers and Typesetting Vol. E. Addison-Wesley, Reading, Massachusetts 1986.