1 The Macedonian language

The file `macedonian.dtx`[1] provides the language-specific macros for the Macedonian language.

1.1 Most important points

The following most important points of the Macedonian language are implemented in this class:

1. The alphabet is complete and is in a correct order, so the numbering (enumerate option) will be in a correct way in Macedonian.
2. Some italic letters in Macedonian (like: d, g, t, p) differ from Bulgarian (and Russian). These characters in italic are perhaps similar to Serbian language. Moreover, the character gje "g with accent" is unique and does not appear in both Bulgarian and Serbian alphabets. These issues have been addressed in the class.
3. The first paragraph of each section is indented.
4. The date format is according to the Macedonian practice. (e.g., [dd] [month in Macadonian] [yyyy] god.)
5. All the terms like chapter, bibliography, index, figure, table, theorem, months names are all adjusted according to the Macedonian practice in books and other materials.

Various "cyrillic" dashes and quotation marks traditionally used in Macedonian are borrowed from German language. French quotation marks may be seen as well in older books. To make them available, the implementation from `bulgarian.dtx` have been inherited for this class. Additional details for dashes and quotation marks can be found in `bulgarian.dtx`. For example, the character `"` is made active.

1.2 Updates to v1.1.

The month names are now in accordance with the Macedonian orthography i.e. a months starts with a non-capital letter[2]. A small typo is fixed.

[1] The file described in this section has version number v1.0 and was last revised on 2015/11/04. This file provides the source code for the Macedonian language definition file. A contribution was made by Stojan Trajanovski ([name].[surname]@gmail.com)

The file heavily relies on the `bulgarian.dtx` version of the babel class and `bulgarian.dtx` was used as a starting point. `bulgarian.dtx` was initially developed by Georgi N. Boshnakov with a final modification by Johannes L. Braams. In addition, some parts from `serbianc.dtx` (created by Filip Brčić) are also used. The specific differences of the Macedonian language from Bulgarian and Serbian have been addressed in this class.

[2] Dario Gjorgjevski contributed for this addition.

1.3 Implementation

The macro `\LdfInit` takes care of preventing that this file is loaded more than once, checking the category code of the `@` sign, etc.

```
⟨*code⟩
\LdfInit{macedonian}{captionsmacedonian}
```

When this file is read as an option, i.e., by the `\usepackage` command, `macedonian` will be an 'unknown' language, in which case we have to make it known. So we check for the existence of `\l@macedonian` to see whether we have to do something here.

```
\ifx\l@macedonian\@undefined
  \@nopatterns{Macedonian}
  \adddialect\l@macedonian0
\fi
```

`\latinencoding` We need to know the encoding for text that is supposed to be which is active at the end of the babel package. If the `fontenc` package is loaded later, then ... too bad!

```
\let\latinencoding\cf@encoding
```

The user may choose between different available Cyrillic encodings—e.g., `X2`, `LCY`, or `LWN`. If the user wants to use a font encoding other than the default (`T2A`), he has to load the corresponding file *before* `macedonian.sty`. This may be done in the following way:

```
\usepackage[LCY,OT1]{fontenc}     %overwrite the default encoding;
\usepackage[english,macedonian]{babel}
```

Note: most people would prefer the `T2A` to `X2`, because `X2` does not contain Latin letters, and users should be very careful to switch the language every time they want to typeset a Latin word inside a Macedonian phrase or vice versa. On the other hand, switching the language is a good practice anyway. With a decent text processing program it does not involve more work than switching between the Macedonian and English keyboard. Moreover that the far most common disruption occurs as a result of forgetting to switch back to cyrillic keyboard.

We parse the `\cdp@list` containing the encodings known to LaTeX in the order they were loaded. We set the `\cyrillicencoding` to the *last* loaded encoding in the list of supported Cyrillic encodings: `OT2`, `LWN`, `LCY`, `X2`, `T2C`, `T2B`, `T2A`, if any.

```
\def\reserved@a#1#2{%
   \edef\reserved@b{#1}%
   \edef\reserved@c{#2}%
   \ifx\reserved@b\reserved@c
     \let\cyrillicencoding\reserved@c
   \fi}
\def\cdp@elt#1#2#3#4{%
   \reserved@a{#1}{OT2}%
   \reserved@a{#1}{LWN}%
```

```
    \reserved@a{#1}{LCY}%
    \reserved@a{#1}{X2}%
    \reserved@a{#1}{T2C}%
    \reserved@a{#1}{T2B}%
    \reserved@a{#1}{T2A}}
\cdp@list
```

Now, if `\cyrillicencoding` is undefined, then the user did not load any of supported encodings. So, we have to set `\cyrillicencoding` to some default value. We test the presence of the encoding definition files in the order from less preferable to more preferable encodings. We use the lowercase names (i.e., `lcyenc.def` instead of `LCYenc.def`).

```
\ifx\cyrillicencoding\undefined
  \IfFileExists{ot2enc.def}{\def\cyrillicencoding{OT2}}\relax
  \IfFileExists{lwnenc.def}{\def\cyrillicencoding{LWN}}\relax
  \IfFileExists{lcyenc.def}{\def\cyrillicencoding{LCY}}\relax
  \IfFileExists{x2enc.def}{\def\cyrillicencoding{X2}}\relax
  \IfFileExists{t2cenc.def}{\def\cyrillicencoding{T2C}}\relax
  \IfFileExists{t2benc.def}{\def\cyrillicencoding{T2B}}\relax
  \IfFileExists{t2aenc.def}{\def\cyrillicencoding{T2A}}\relax
```

If `\cyrillicencoding` is still undefined, then the user seems not to have a properly installed distribution. A fatal error.

```
\ifx\cyrillicencoding\undefined
    \PackageError{babel}%
    {No Cyrillic encoding definition files were found}%
    {Your installation is incomplete. \MessageBreak
    You need at least one of the following files: \MessageBreak
    \space\space
    x2enc.def, t2aenc.def, t2benc.def, t2cenc.def, \MessageBreak
    \space\space
    lcyenc.def, lwnenc.def, ot2enc.def.}%
  \else
```

We avoid `\usepackage[\cyrillicencoding]{fontenc}` because we don't want to force the switch of `\encodingdefault`.

```
    \lowercase
      \expandafter{\expandafter\input\cyrillicencoding enc.def\relax}%
  \fi
\fi

    \PackageInfo{babel}
      {Using `\cyrillicencoding' as a default Cyrillic encoding}%


\DeclareRobustCommand{\Macedonian}{%
  \fontencoding\cyrillicencoding\selectfont
  \let\encodingdefault\cyrillicencoding
  \expandafter\set@hyphenmins\macedonianhyphenmins
  \language\l@macedonian}
```

```
\DeclareRobustCommand{\English}{%
  \fontencoding\latinencoding\selectfont
  \let\encodingdefault\latinencoding
  \expandafter\set@hyphenmins\englishhyphenmins
  \language\l@english}
\let\Mkd\Macedonian
\let\Mk\Macedonian
\let\cyrillictext\Macedonian
\let\cyr\Macedonian
\let\Eng\English
\def\selectenglanguage{\selectlanguage{english}}
\def\selectmklanguage{\selectlanguage{macedonian}}
```

Since the `X2` encoding does not contain Latin letters,we should make some redefinitions of LaTeX macros which implicitly produce Latin letters.

```
\expandafter\ifx\csname T@X2\endcsname\relax\else
```

We put `\latinencoding` in braces to avoid problems with `\@alph` inside minipages (e.g., footnotes inside minipages) where `\@alph` is expanded and we get for example '`\fontencoding OT1`' (`\fontencoding` is robust).

```
  \def\@Alph@eng#1{{\fontencoding{\latinencoding}\selectfont
      \ifcase#1\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
      K\or L\or M\or N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or
      X\or Y\or Z\else \@ctrerr\fi}}%
  \def\@alph@eng#1{{\fontencoding{\latinencoding}\selectfont
      \ifcase#1\or a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or
      k\or l\or m\or n\or o\or p\or q\or r\or s\or t\or u\or v\or w\or
      x\or y\or z\else \@ctrerr\fi}}%
  \let\@Alph\@Alph@eng
  \let\@alph\@alph@eng
```

Unfortunately, the commands `\AA` and `\aa` are not encoding dependent in LaTeX (unlike e.g., `\oe` or `\DH`). They are defined as `\r{A}` and `\r{a}`. This leads to unpredictable results when the font encoding does not contain the Latin letters 'A' and 'a' (like `X2`).

```
  \DeclareTextSymbolDefault{\AA}{OT1}
  \DeclareTextSymbolDefault{\aa}{OT1}
  \DeclareTextCommand{\AA}{OT1}{\r A}
  \DeclareTextCommand{\aa}{OT1}{\r a}
\fi
```

The following block redefines the character class of uppercase Greek letters and some accents, if it is equal to 7 (variable family), to avoid incorrect results if the font encoding in some math family does not contain these characters in places of OT1 encoding. The code was taken from `amsmath.dtx`. See comments and further explanation there.

```
\begingroup\catcode`\"=12
% uppercase greek letters:
\def\@tempa#1{\expandafter\@tempb\meaning#1\relax\relax\relax\relax
```

```
 "0000\@nil#1}
\def\@tempb#1"#2#3#4#5#6\@nil#7{%
\ifnum"#2=7 \count@"1#3#4#5\relax
\ifnum\count@<"1000 \else \global\mathchardef#7="0#3#4#5\relax \fi
 \fi}
\@tempa\Gamma\@tempa\Delta\@tempa\Theta\@tempa\Lambda\@tempa\Xi
\@tempa\Pi\@tempa\Sigma\@tempa\Upsilon\@tempa\Phi\@tempa\Psi
\@tempa\Omega
% some accents:
\def\@tempa#1#2\@nil{\def\@tempc{#1}}\def\@tempb{\mathaccent}
\expandafter\@tempa\hat\relax\relax\@nil
\ifx\@tempb\@tempc
\def\@tempa#1\@nil{#1}%
\def\@tempb#1{\afterassignment\@tempa\mathchardef\@tempc=}%
\def\do#1"#2{}
\def\@tempd#1{\expandafter\@tempb#1\@nil
 \ifnum\@tempc>"FFF
 \xdef#1{\mathaccent"\expandafter\do\meaning\@tempc\space}%
 \fi}
\@tempd\hat\@tempd\check\@tempd\tilde\@tempd\acute\@tempd\grave
\@tempd\dot\@tempd\ddot\@tempd\breve\@tempd\bar
\fi
\endgroup
```

The user should use the `inputenc` package when any 8-bit Cyrillic font encoding is used, selecting one of the Cyrillic input encodings. We do not assume any default input encoding, so the user should explicitly call the `inputenc` package by `\usepackage{inputenc}`. We also removed `\AtBeginDocument`, so `inputenc` should be used before babel.

```
\@ifpackageloaded{inputenc}{}{%
\def\reserved@a{LWN}%
\ifx\reserved@a\cyrillicencoding\else
\def\reserved@a{OT2}%
\ifx\reserved@a\cyrillicencoding\else
\PackageWarning{babel}%
{No input encoding specified for Macedonian language}\fi\fi}
```

Now we define two commands that offer the possibility to switch between Cyrillic and Roman encodings.

`\cyrillictext` `\latintext`

The command `\cyrillictext` will switch from Latin font encoding to the Cyrillic font encoding, the command `\latintext` switches back. This assumes that the 'normal' font encoding is a Latin one. These commands are *declarations*, for shorter peaces of text the commands `\textlatin` and `\textcyrillic` can be used.

We comment out `\latintext` since it is defined in the core of babel (babel.def). We add the shorthand `\lat` for `\latintext`. Note that `\cyrillictext` has been defined above.

```
% \DeclareRobustCommand{\latintext}{%
```

```
% \fontencoding{\latinencoding}\selectfont
%   \def\encodingdefault{\latinencoding}}
\let\lat\latintext
```

`\textcyrillic` `\textlatin` These commands take an argument which is then typeset using the requested font encoding. `\textlatin` is commented out since it is defined in the core of babel. (It is defined there with `\DeclareRobustCommand` instead.)

```
\DeclareTextFontCommand{\textcyrillic}{\cyrillictext}
% \DeclareTextFontCommand{\textlatin}{\latintext}
```

The next step consists of defining commands to switch to (and from) the Macedonian language.

`\captionsmacedonian` The macro `\captionsmacedonian` defines all strings used in the four standard document classes provided with LaTeX. The two commands `\cyr` and `\lat` activate Cyrillic resp. Latin encoding.

```
\addto\captionsmacedonian{%
  \def\prefacename{%
    {\cyr\CYRP\cyrr\cyre\cyrd\cyrg\cyro\cyrv\cyro\cyrr}}%
  \def\refname{%
    {\cyr\CYRL\cyri\cyrt\cyre\cyrr\cyra\cyrt\cyru\cyrr\cyra}}%
  \def\abstractname{%
    {\cyr\CYRA\cyrb\cyrs\cyrt\cyrr\cyra\cyrk\cyrt}}%
  \def\bibname{%
    {\cyr\CYRB\cyri\cyrb\cyrl\cyri\cyro\cyrg\cyrr\cyra\cyrf\cyri\cyrje\cyra}}%
  \def\chaptername{%
    {\cyr\CYRG\cyrl\cyra\cyrv\cyra}}%
  \def\appendixname{%
    {\cyr\CYRP\cyrr\cyri\cyrl\cyro\cyrg}}%
  \def\contentsname{%
    {\cyr\CYRS\cyro\cyrd\cyrr\cyrzh\cyri\cyrn\cyra}}%
  \def\listfigurename{%
    {\cyr\CYRL\cyri\cyrs\cyrt\cyra\ \cyrn\cyra\ \cyrs\cyrl\cyri\cyrk\cyri}}%
  \def\listtablename{%
    {\cyr\CYRL\cyri\cyrs\cyrt\cyra\ \cyrn\cyra\ \cyrt\cyra\cyrb\cyre\cyrl\cyri}}%
  \def\indexname{%
    {\cyr\CYRI\cyrn\cyrd\cyre\cyrk\cyrs\ \cyrn\cyra\ \cyrt\cyre\cyrr\cyrm\cyri\cyrn\cyri}}%
  \def\authorname{%
    {\cyr\CYRA\cyrv\cyrt\cyro\cyrr}}%
  \def\figurename{%
    {\cyr\CYRS\cyrl\cyri\cyrk\cyra}}%
  \def\tablename{%
    {\cyr\CYRT\cyra\cyrb\cyre\cyrl\cyra}}%
  \def\partname{%
    {\cyr\CYRD\cyre\cyrl}}%
  \def\enclname{%
    {\cyr\CYRP\cyrr\cyri\cyrl\cyro\cyrz\cyri}}%
  \def\ccname{%
    {\cyr\cyrk\cyro\cyrp\cyri\cyrje\cyra}}%
```

```
  \def\headtoname{%
    {\cyr\CYRZ\cyra}}%
  \def\pagename{%
    {\cyr\cyrs\cyrt\cyrr.}}%
  \def\seename{%
    {\cyr\cyrv\cyri\cyrd\cyri}}%
  \def\alsoname{%
    {\cyr\cyrv\cyri\cyrd\cyri\ \cyri\cyrs\cyrt\cyro\cyrt\cyra\cyrk\cyra}}%
    \def\proofname{%
    {\normalfont \textbf{\cyr\cyrd\cyro\cyrk\cyra\cyrz}}}%
     \def\theoremname{%
    {\normalfont \textbf{\cyr\CYRT\cyre\cyro\cyrr\cyre\cyrm\cyra}}}%
    \def\corollaryname{%
    {\normalfont \textbf{\cyr\CYRP\cyro\cyrs\cyrl\cyre\cyrd\cyri\cyrc\cyra}}}%
    \def\lemmaname{%
    {\normalfont \textbf{\cyr\CYRL\cyre\cyrm\cyra}}}%
    \def\glossaryname{%
    {\cyr\CYRR\cyre\cyrc\cyrn\cyri\cyrk}}%
}
```

`\datemacedonian` The macro `\datemacedonian` redefines the command `\today` to produce Macedonian dates. It also provides the command `\todayRoman` which produces the date with the month in capital roman numerals, a popular format for dates in Macedonian.

```
\def\datemacedonian{%
  \def\month@macedonian{\ifcase\month\or
    \cyrje\cyra\cyrn\cyru\cyra\cyrr\cyri\or
    \cyrf\cyre\cyrv\cyrr\cyru\cyra\cyrr\cyri\or
    \cyrm\cyra\cyrr\cyrt\or
    \cyra\cyrp\cyrr\cyri\cyrl\or
    \cyrm\cyra\cyrje\or
    \cyrje\cyru\cyrn\cyri\or
    \cyrje\cyru\cyrl\cyri\or
    \cyra\cyrv\cyrg\cyru\cyrs\cyrt\or
    \cyrs\cyre\cyrp\cyrt\cyre\cyrm\cyrv\cyrr\cyri\or
    \cyro\cyrk\cyrt\cyro\cyrm\cyrv\cyrr\cyri\or
    \cyrn\cyro\cyre\cyrm\cyrv\cyrr\cyri\or
    \cyrd\cyre\cyrk\cyre\cyrm\cyrv\cyrr\cyri\fi}%
  \def\month@Roman{\expandafter\@Roman\month}%
  \def\today{\number\day~\month@macedonian\ \number\year~\cyrg\cyro\cyrd.}%
  \def\todayRoman{\number\day.\,\month@Roman.\,\number\year~\cyrg\cyro\cyrd.}%
}
```

`\todayRoman` The month is often written with roman numbers in Macedonian dates. Here we define date in this format:

```
\def\Romannumeral#1{\uppercase\expandafter{\romannumeral #1}}
\def\todayRoman{\number\day.\Romannumeral{\month}.\number\year~\cyrg\cyro\cyrd.}
```

`\extrasmacedonian` The macro `\extrasmacedonian` will perform all the extra definitions needed for

the Macedonian language. The macro `\noextrasmacedonian` is used to cancel the actions of `\extrasmacedonian`.

The first action we define is to switch on the selected Cyrillic encoding whenever we enter 'macedonian'.

```
\addto\extrasmacedonian{\cyrillictext}
```

When the encoding definition file was processed by LaTeX the current font encoding is stored in `\latinencoding`, assuming that LaTeX uses `T1` or `OT1` as default. Therefore we switch back to `\latinencoding` whenever the Macedonian language is no longer 'active'.

```
\addto\noextrasmacedonian{\latintext}
```

For Macedonian the `"` character also is made active.

```
\initiate@active@char{"}
```

The code above is necessary because we need extra active characters. The character `"` is used as indicated in table **??**. We specify that the Macedonian group of shorthands should be used.

```
\addto\extrasmacedonian{\languageshorthands{macedonian}}
```

These characters are 'turned on' once, later their definition may vary.

```
\addto\extrasmacedonian{%
  \bbl@activate{"}}
\addto\noextrasmacedonian{%
  \bbl@deactivate{"}}
```

The `X2` and `T2*` encodings do not contain `spanish_shriek` and `spanish_query` symbols; as a consequence, the ligatures '`?`' and '`!`' do not work with them (these characters are useless for Cyrillic texts anyway). But we define the shorthands to emulate these ligatures (optionally).

We do not use `\latinencoding` here (but instead explicitly use `OT1`) because the user may choose `T2A` to be the primary encoding, but it does not contain these characters.

```
⟨*spanishligs⟩
\declare@shorthand{macedonian}{?`}{\UseTextSymbol{OT1}\textquestiondown}
\declare@shorthand{macedonian}{!`}{\UseTextSymbol{OT1}\textexclamdown}
⟨/spanishligs⟩
```

To be able to define the function of '`"`', we first define a couple of 'support' macros.

`\dq` We save the original double quote character in `\dq` to keep it available, the math accent `\"`can now be typed as '`"`'.

```
\begingroup \catcode`\"12
\def\reserved@a{\endgroup
  \def\@SS{\mathchar"7019}
  \def\dq{"}}
\reserved@a
```

Now we can define the doublequote macros: german and french quotes. We use definitions of these quotes made in babel.sty. The french quotes are contained in the `T2*` encodings.

```
\declare@shorthand{macedonian}{"`}{\glqq}
\declare@shorthand{macedonian}{"'}{\grqq}
\declare@shorthand{macedonian}{"<}{\flqq}
\declare@shorthand{macedonian}{">}{\frqq}
```

Some additional commands:

```
\declare@shorthand{macedonian}{""}{\hskip\z@skip}
\declare@shorthand{macedonian}{"~}{\textormath{\leavevmode\hbox{-}}{-}}
\declare@shorthand{macedonian}{"=}{\nobreak-\hskip\z@skip}
\declare@shorthand{macedonian}{"|}{%
\textormath{\nobreak\discretionary{-}{}{\kern.03em}%
\allowhyphens}{}}
```

The next two macros for `"-` and `"---` are somewhat different. We must check whether the second token is a hyphen character:

```
\declare@shorthand{macedonian}{"-}{%
```

If the next token is '`-`', we typeset an emdash, otherwise a hyphen sign:

```
  \def\macedonian@sh@tmp{%
    \if\macedonian@sh@next-\expandafter\macedonian@sh@emdash
    \else\expandafter\macedonian@sh@hyphen\fi
  }%
```

TEX looks for the next token after the first '`-`': the meaning of this token is written to `\macedonian@sh@next` and `\macedonian@sh@tmp` is called.

```
  \futurelet\macedonian@sh@next\macedonian@sh@tmp}
```

Here are the definitions of hyphen and emdash. First the hyphen:

```
\def\macedonian@sh@hyphen{\nobreak\-\bbl@allowhyphens}
```

For the emdash definition, there are the two parameters: we must 'eat' two last hyphen signs of our emdash ...:

```
\def\macedonian@sh@emdash#1#2{\cdash-#1#2}
```

`\cdash` ... these two parameters are useful for another macro: `\cdash`:

```
\ifx\cdash\undefined % should be defined earlier
\def\cdash#1#2#3{\def\tempx@{#3}%
\def\tempa@{-}\def\tempb@{~}\def\tempc@{*}%
 \ifx\tempx@\tempa@\@Acdash\else
  \ifx\tempx@\tempb@\@Bcdash\else
   \ifx\tempx@\tempc@\@Ccdash\else
    \errmessage{Wrong usage of cdash}\fi\fi\fi}
```

second parameter (or third for `\cdash`) shows what kind of emdash to create in next step

`"---` ordinary (plain) Cyrillic emdash inside text: an unbreakable thinspace will be inserted before only in case of a *space* before the dash (it is necessary for dashes after display maths formulae: there could be lists, enumerations etc. started with "—where a is ..." i.e., the dash starts a line). (Firstly there were planned rather soft rules for user:he may put a space before the dash or not. But it is difficult to place this thinspace automatically, i.e., by checking modes because after display formulae TEX uses horizontal mode. Maybe there is a misunderstanding? Maybe there is another way?) After a dash a breakable thinspace is always placed;

```
% What is more grammatically: .2em or .2\fontdimen6\font?
\def\@Acdash{\ifdim\lastskip>\z@\unskip\nobreak\hskip.2em\fi
\cyrdash\hskip.2em\ignorespaces}%
```

`"--~` emdash in compound names or surnames (like Mendeleev–Klapeiron); this dash has no space characters around; after the dash some space is added `\exhyphenalty`

```
\def\@Bcdash{\leavevmode\ifdim\lastskip>\z@\unskip\fi
 \nobreak\cyrdash\penalty\exhyphenpenalty\hskip\z@skip\ignorespaces}%
```

`"--*` for denoting direct speech (a space like `\enskip` must follow the emdash);

```
\def\@Ccdash{\leavevmode
 \nobreak\cyrdash\nobreak\hskip.35em\ignorespaces}%
%\fi
```

`\cyrdash` Finally the macro for "body" of the Cyrillic emdash. The `\cyrdash` macro will be defined in case this macro hasn't been defined in a fontenc file. For T2*fonts, cyrdash will be placed in the code of the English emdash thus it uses ligature `---`.

```
% Is there an IF necessary?
\ifx\cyrdash\undefined
\def\cyrdash{\hbox to.8em{--\hss--}}
\fi
```

Here a really new macro—to place thinspace between initials. This macro used instead of `\,` allows hyphenation in the following surname.

```
\declare@shorthand{macedonian}{",}{\nobreak\hskip.2em\ignorespaces}
```

The Macedonian hyphenation patterns can be used with `\lefthyphenmin` and `\righthyphenmin` set to 2.

```
\providehyphenmins{\CurrentOption}{\tw@\tw@}
\fi
```

Now the action `\extrasmacedonian` has to execute is to make sure that the command `\frenchspacing` is in effect. If this is not the case the execution of `\noextrasmacedonian` will switch it off again.

```
\addto\extrasmacedonian{\bbl@frenchspacing}
\addto\noextrasmacedonian{\bbl@nonfrenchspacing}
```

Make the double quotes produce the traditional quotes used in Macedonian texts (these are the German quotes).

```
% \initiate@active@char{‘}
%  \initiate@active@char{’}
% \addto\extrasmacedonian{%
%   \bbl@activate{‘}}
% \addto\extrasmacedonian{%
%   \bbl@activate{’}}
% \addto\noextrasmacedonian{%
%   \bbl@deactivate{‘}}
% \addto\noextrasmacedonian{%
%   \bbl@deactivate{’}}
% \def\mlron{\bbl@activate{‘}\bbl@activate{’}}
% \def\mlroff{\bbl@deactivate{‘}\bbl@deactivate{’}}
% \declare@shorthand{macedonian}{‘‘}{\glqq}
% \declare@shorthand{macedonian}{’’}{\grqq}
```

In Macedonian the first paragraph of each section should be indented. The implementation from `serbianc.dtx` (created by Filip Brčić [brcha@gna.org]) is used for such a purpose.

```
\ifx\fmtname plain \else
  \let\@aifORI\@afterindentfalse
  \def\bbl@frenchindent{\let\@afterindentfalse\@afterindenttrue
                        \@afterindenttrue}
  \def\bbl@nonfrenchindent{\let\@afterindentfalse\@aifORI
                           \@afterindentfalse}
  \addto\extrasmacedonian{\bbl@frenchindent}
  \addto\noextrasmacedonian{\bbl@nonfrenchindent}
\fi
```

Next we add a new enumeration style for Macedonian manuscripts with Cyrillic letters,and later on we define some math operator names in accordance with Macedonian typesetting traditions.

`\@Alph@mkd` We begin by defining `\@Alph@mkd` which works like `\@Alph`, but produces (uppercase) Cyrillic letters intead of Latin ones.

```

\def\enumBul{\let\@Alph\@Alph@mkd \let\@alph\@alph@mkd}
\def\enumEng{\let\@Alph\@Alph@eng \let\@alph\@alph@eng}
\def\enumLat{\let\@Alph\@Alph@eng \let\@alph\@alph@eng}
\addto\extrasmacedonian{\enumBul}
\addto\noextrasmacedonian{\enumLat}
\def\@Alph@mkd#1{%
  \ifcase#1\or
  \CYRA\or \CYRB\or \CYRV\or \CYRG\or \CYRD\or \’{\CYRG}\or \CYRE\or \CYRZH\or
  \CYRZ\or \CYRDZE\or \CYRI\or \CYRJE\or \CYRK\or \CYRL\or \CYRLJE\or
  \CYRM\or \CYRN\or \CYRNJE\or \CYRO\or \CYRP\or \CYRR\or \CYRS\or
  \CYRT\or \’{\CYRK}\or \CYRU\or \CYRF\or \CYRH\or \CYRC\or \CYRCH\or \CYRDZHE\or \CYRSH\else
  \@ctrerr\fi
```

```
}
\def\@Alph@eng#1{%
  \ifcase#1\or
  A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or K\or L\or M\or
  N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or X\or Y\or Z\else
  \@ctrerr\fi
  }
```

`\@alph@mkd` The macro `\@alph@mkd` is similar to `\@Alph@mkd`; it produces lowercase Macedonian letters.

```
\def\@alph@mkd#1{%
  \ifcase#1\or
  \cyra\or \cyrb\or \cyrv\or \cyrg\or \cyrd\or \'{\cyrg}\or \cyre\or \cyrzh\or
  \cyrz\or \cyrdze\or \cyri\or \cyrje\or \cyrk\or \cyrl\or \cyrlje\or
  \cyrm\or \cyrn\or \cyrnje\or \cyro\or \cyrp\or \cyrr\or \cyrs\or
  \cyrt\or \'{\cyrk}\or \cyru\or \cyrf\or \cyrh\or \cyrc\or \cyrch\or \cyrdzhe\or \cyrsh\else
  \@ctrerr\fi
  }
\def\@alph@eng#1{%
  \ifcase#1\or
  a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or k\or l\or m\or
  n\or o\or p\or q\or r\or s\or t\or u\or v\or w\or x\or y\or z\else
  \@ctrerr\fi
  }
```

`Characters in italic` When writing in italic, Macedonian differs from Bulgarian, Russian and other Slavic languages. (Perhaps similar to Serbian), the small italic letters: b, d, g, t and p are different. Moreover, the character gje "g with accent" is kind of unique. The following code ensures these characters to have a correct appearance in Macedonian Italic.

```
\newcommand{\overbar}[1]{\mkern 1.5mu\overline{\mkern-1.5mu#1\mkern-0.2mu}\mkern 1.0mu}
\newcommand*{\textoverline}[1]{$\overbar{\hbox{#1}}\m@th$}
\newcommand{\overbarshort}[1]{\mkern 1.7mu\overline{\mkern-0.0mu#1\mkern+1.2mu}\mkern 0.0mu}
\newcommand*{\textoverlineshort}[1]{$\overbarshort{\hbox{#1}}\m@th$}
\newcommand*{\my@test@it}{it}
\newcommand*{\IfItalic}[2]{%
  \ifx\f@shape\my@test@it
    #1%\expandafter#1
  \else
    #2%\expandafter#2
  \fi
}
\newcommand{\tbar}[1]{\mbox{%
  \sbox0{#1}\sbox2{\'{}}%
  \ooalign{\hidewidth\raise\dimexpr\ht0-\ht2+0.45ex\box2 \hidewidth\cr#1\cr}}}
\let\oldcyrb\cyrb
\renewcommand{\cyrb}{\IfItalic{\textit{$\delta$}}{\oldcyrb}}
\let\oldcyrd\cyrd
\renewcommand{\cyrd}{\IfItalic{\textit{g}}{\oldcyrd}}
```

```
\let\oldcyrg\cyrg
\renewcommand{\cyrg}{\IfItalic{\textoverlineshort{\i}}{\oldcyrg}}
\DeclareTextCompositeCommand{\'}{T2A}{\cyrg}{\IfItalic{\tbar{\textoverlineshort{\i}}}
{\'{\oldcyrg}}}
\let\oldcyrt\cyrt
\renewcommand{\cyrt}{\IfItalic{\textoverline{\textit{\cyrsh}}}{\oldcyrt}}
\let\oldcyrp\cyrp
\renewcommand{\cyrp}{\IfItalic{\textoverline{\textit{\cyri}}}{\oldcyrp}}
```

Some math functions in Macedonian math books have other names: e.g., `sinh` in Macedonian is written as `sh` etc. So we define a number of new math operators.

`\sinh`:

```
\def\sh{\mathop{\operator@font sh}\nolimits}
```

`\cosh`:

```
\def\ch{\mathop{\operator@font ch}\nolimits}
```

`\tan`:

```
\def\tg{\mathop{\operator@font tg}\nolimits}
```

`\arctan`:

```
\def\arctg{\mathop{\operator@font arctg}\nolimits}
```

`\arccot`:

```
\def\arcctg{\mathop{\operator@font arcctg}\nolimits}
```

The following macro conflicts with `\th` defined in Latin 1 encoding: `\tanh`:

```
\addto\extrasrussian{%
  \babel@save{\th}%
  \let\ltx@th\th
  \def\th{\textormath{\ltx@th}%
                     {\mathop{\operator@font th}\nolimits}}%
  }
```

`\cot`:

```
\def\ctg{\mathop{\operator@font ctg}\nolimits}
```

`\coth`:

```
\def\cth{\mathop{\operator@font cth}\nolimits}
```

`\csc`:

```
\def\cosec{\mathop{\operator@font cosec}\nolimits}
```

The macro `\ldf@finish` takes care of looking for a configuration file, setting the main language to be switched on at `\begin{document}` and resetting the category code of `@` to its original value.

```
\ldf@finish{macedonian}
```

⟨/code⟩