On a new LaTeX package for automatic Hungarian definite article

Tibor Tómács

Eszterházy Károly Catholic University,
Institute of Mathematics and Informatics, Eger, Hungary
tomacs.tibor@uni-eszterhazy.hu

Abstract. In the LaTeX document preparation system (see [3]) it is possible to insert automatically the appropriate definite article for cross-references and other commands containing texts in Hungarian language documents. Thus, if these change, the definite articles will also change accordingly.

Such a tool is the `magyar.ldf`, which sets the Hungarian typography and also handles the automatic definite articles. Another one is the `nevelok` package, created specifically for this task. Both packages work with numerous errors and have shortcomings. This motivated the author of this paper to develop a new package, that corrects these errors and fills the gaps in. The new package is called *huaz*.

Keywords: Hungarian definite article, LaTeX, babel package, magyar.ldf, utf8, latin2, pdflatex, xelatex, lualatex

AMS Subject Classification: 68U01, 68U15, 68U99

1. Introduction

In Hungarian there are two definite articles, *"a"* and *"az"*, which are determined by the pronunciation of the subsequent word. The definite article is *"az"*, if the first phoneme of the pronounced word is a vowel, otherwise it is *"a"*. This seems simple, but consider the following cases:

- When you refer to a page number in a LaTeX document, you have to use the `\pageref` command. If you want to put a definite article in front of it, it depends on what the page number is. For example *"az 1. oldal"* (the page 1), *"a 2. oldal"*, *"az 5. oldal"*, *"a 10. oldal"*. So "a `\pageref{`⟨key⟩`}`" may not

give the correct result. The problem is similar for all cross-references (`\ref`, `\pageref`, `\eqref`, `\cite`).

- It is also important whether a word is a Roman numeral or not. For example *"az V. fejezet"* (the Chapter V), if *V* is a Roman numeral, but *"a V. fejezet"*, if *V* represents a letter or the 22 alphanumeric number (*V* is the 22nd letter in the English alphabet).

- Some Hungarian consonants have special properties from the aspect of the definite article. For example *"az M betű"*, but *"a Magyar Közlöny"*; *"az Ny betű"*, but *"a Nyugdíjfolyósító Igazgatóság"*; etc.

First, we mention two existing LATEX solutions for the automatic handling of Hungarian definite articles, highlighting their errors and shortcomings.

1.1. The magyar.ldf

A possible way is to use the `magyar.ldf` file, that sets the `magyar` (or `hungarian`) option of the `babel` package (see full documentation in [6, 10]).

The commands defined in `magyar.ldf` (`\az`, `\aref`, `\apageref`, `\acite`, etc.) work properly on a basic level, but major errors can occur. This motivated the writing of a new package:

- The `\eqref` command has no definite article version. (The `\eqref` is defined in the `amsmath` package to reference equations. See in [1].) Instead, `\aref({⟨key⟩})` can be used, but in italics font style environment, it does not give upright result like the `\eqref`. Another option solving this problem is the `\az{\eqref{⟨key⟩}}` command. However, neither solution handles Roman numerals, or `\tag` commands labeled equations.

- The `\az{\ref{⟨key⟩}}` and the `\az{\pageref{⟨key⟩}}` do not handle Roman numeral references.

- The previous error also exists when the `\az` command contains a text starting with a Roman numeral. For example the result of the `\az{V.~osztály}` is *"a V. osztály"* (the Class V). The correct form would be *"az V. osztály"*.

- If you need an automatic definite article for a none cross-reference, but for a text or a command that stores text, then accented letters are detected incorrectly in case of UTF-8 encoding. The basic cause of this error is that UTF-8 use more than one byte to encode the characters, which `magyar.ldf` does not take into account. Therefore, for example, the result of the `\az{ágy}` is *"a ágy"* (the bed), because it does not perceive the letter *"á"* as a letter, so it does not recognize that it is a vowel. Surprisingly, the result of the `\az{száz}` is *"az száz"* (the hundred), which is also incorrect. The reason is that since *"á"* is not a letter for the `magyar.ldf`, it considers the two-digit consonant *"sz"* separate, which requires to be preceded by *"az"*.

- Although it cannot be considered as a mistake, it is uncomfortable that, as an example, in the `\az{\textbf{N betű}}` the `\textbf` command interferes with the detection of the space after N, so the definite article will incorrectly become *"a"* ("a **N betű**" = the letter N).

1.2. The nevelok package

The `nevelok` LaTeX package was created in 2015, and it is also designed to handle automatic Hungarian definite articles (see in [2]). The author probably didn't know the opportunities of the `magyar.ldf`, that already existed at that time, that is why he has created this package.

Tested on TeX Live 2022, I have experienced that the `nevelok` package does not handle any form of accented letters, and in a lot of cases gives wrong definite article for cross-references. The reason of the latter problem is that it examines the `\ref` command, that is not expandable. It is not recommended to use this package in the current version (1.03).

2. Purpose and operation of the huaz package

The mistakes and shortcomings of the `magyar.ldf` and the `nevelok.sty` motivated me to design a completely new package called *huaz*.

The purpose of the `huaz` package is to help the Hungarian LaTeX users inserting automatically the correct definite articles for cross-references and commands containing text. Thus, if these change, the definite articles will also change accordingly.

The package is still under testing, but the Reader can try it on the Overleaf before publishing it on The Comprehensive TeX Archive Network.

The `huaz` package adds the definite article "az" to the given text in the following cases:

1. The first letter is a vowel (lowercase or uppercase, accented or not, Hungarian or not, specified with UTF-8, ISO-8859-2 characters or accent commands).

2. The first letter is a consonant, whose pronunciation begins with a vowel (for example F, L, M, etc.) while the second character (if any) is not a letter, but a number, punctuation mark or space. For example, "M-10". We also listed some non-Hungarian accented consonants here. For example "Ñ.1".

3. The block of the first two characters is a two-digit consonant, whose pronunciation begins with a vowel (for example Ny, Ly, Sz, etc.) while the third character (if any) is not a letter, but a number, punctuation mark or space. For example, "NY betű".

4. The first character is 5.

5. It starts with a 1, 4, 7 or 10 digit number and the first digit is 1 (egy = one, ezer = thousand, egymillió = one million, egymilliárd = one billion).

If the characters, at the beginning of the word, can also be interpreted as Roman numerals, you can choose to convert them to Arabic numerals and define the definite article for that or not. For example, in the case of "X/A".

3. Using the huaz package

The `huaz` package must be loaded in the usual way:

```
\usepackage{huaz}
```

There are no package options. It works for UTF-8 (`utf8`) and ISO-8859-2 (`latin2`) encoded source files, but it also handles accent commands correctly. It is also compatible with `pdflatex`, `xelatex` and `lualatex` compilers. As an example, with the `pdflatex` compiler, the following loading is suitable:

```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage{huaz}
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
\usepackage[magyar]{babel}
\begin{document}
 ...
\end{document}
```

It is also compatible with the `hyperref` package (see [5]).

The `defaults=hu-min` option of the `magyar.ldf` makes its own automatic definite article commands available. It is not required, but useful to turn it off when using the `huaz` package. To do this, replace the line

```
\PassOptionsToPackage{defaults=hu-min}{magyar.ldf}
```

with

```
\PassOptionsToPackage{defaults=hu-min,az=no,
                      shortrefcmds=no,hunnewlabel=no}{magyar.ldf}
```

The `huaz` package uses the services of the `xstring`, `refcount` and `iftex` packages (see [4, 7, 9]), so they are also loaded in. On the other hand, some changes in the LATEX kernel introduced on October 10, 2021 are used (see in [8]), so the package only works correctly on systems installed after that.

3.1. The commands

`\az{`⟨text⟩`}` • The ⟨text⟩ is preceded by the appropriate definite article in a low-ercase form. If at the beginning of the ⟨text⟩, there are characters that can be interpreted as Roman numerals, then the definite article is adjusted to the Arabic equivalent. For example

```
Idén \az{V.B}~osztály rendezi a farsangot.
```

„Idén az V.B osztály rendezi a farsangot.".

The ⟨text⟩ can also be a command that stores text. For example

```
\newcommand{\osztaly}{V.B}
Idén \az{\osztaly}~osztály rendezi a farsangot.
```

„Idén az V.B osztály rendezi a farsangot.".

The ⟨text⟩ can also contain text formatting commands (see the Subsection 3.3 for more information). For example

```
\newcommand{\osztaly}{V.B}
Idén \az{\textbf{\osztaly}}~osztály rendezi a farsangot.
```

„Idén az **V.B** osztály rendezi a farsangot.".

The ⟨text⟩ can also be a standard cross-reference (`\ref`, `\pageref`, `\eqref`, `\cite`). For example

```
\section{Cím}\label{seca}
\section{Cím}\label{secb}
\az{\ref{seca}}.~szakaszban, \az{\textbf{\ref{secb}}}.~szakaszban
```

„az 1. szakaszban, a **2**. szakaszban".

```
\renewcommand{\thesection}{\Roman{section}}
\section{Cím}\label{seca}
\section{Cím}\label{secb}
\az{\ref{seca}}.~szakaszban, \az{\textbf{\ref{secb}}}.~szakaszban
```

„az I. szakaszban, a **II**. szakaszban".

The ⟨text⟩ and the `\az` command have the following limitations:

1. At the beginning of the ⟨text⟩, only `\ref`, `\pageref`, `\eqref`, `\cite` cross-reference commands work correctly. For example the `\ref*` and `\pageref*` commands of the `hyperref` package do not work directly as ⟨text⟩, but it can be solved with the `\az*` command (see later).

2. At the beginning of the ⟨text⟩, the `\cite` command works fine in default case, with `natbib`, and with `bibtex`. When using the `biblatex` package, it works well if the `style` or `citestyle` options are `numeric`, `numeric-verb`, `alphabetic`, `alphabetic-verb` or `authoryear`. It also works well if we do not specify any of the `style` or `citestyle` options.

3. You cannot insert text into the pdf bookmark with the `\az` command. So, for example, the following code will not give correct bookmark, if you use `hyperref` or `bookmark` package:

```
\section{...\az{\ref{sec}}...}
```

However, the title will appear fine in the text, headers and the table of contents. The problem can be solved with the `\azsaved` command (see later).

`\az*{⟨text⟩}` • The same case without `*`, but only the definite article is written out, the ⟨text⟩ is not. For example, using the `hyperref` package

```
\section{Cím}\label{sec}
\az*{\ref{sec}}~\ref*{sec}.~szakaszban
```

`\azv{⟨text⟩}` • Same as `\az{⟨text⟩}`, but if at the beginning of the ⟨text⟩ there are characters that can be interpreted as Roman numerals, then the definite article is not aligned with the Arabic equivalent, but as simple characters. For example

```
\renewcommand{\thesection}{\Alph{section}}
\setcounter{section}{21}
\section{Cím}\label{sec}
\az{\ref{sec}}.~szakaszban, \azv{\ref{sec}}.~szakaszban
```

The result is "az V. szakaszban, a V. szakaszban", as in the first case the letter V was interpreted as a Roman numeral, but not in the second case. Since the `section` counter is now set to an alphanumeric number, the second case is the right one.

`\azv*{⟨text⟩}` • The same case without `*`, but only the definite article is written out, the ⟨text⟩ is not.

`\Az{⟨text⟩}` `\Az*{⟨text⟩}` `\Azv{⟨text⟩}` `\Azv*{⟨text⟩}` • In the names of the commands, the letter "a" can be replaced by the letter "A". Then the definite article will begin with capital letter, which is necessary at the beginning of sentences.

`\azsaved` • When you use any of the previous commands, an `\azsaved` expandable command is generated, too. The expansion of it is the definite article that must precede the ⟨text⟩.

Using `hyperref` or `bookmark` package, as mentioned before, the title, header, table of contents will be fine with the following code, but the bookmark of the pdf will not:

```
\section{\Az{\ref{sec}}...}
```

The **\azsaved** command can solve this problem:

```
\section{\texorpdfstring{\Az{\ref{sec}}...}{\azsaved~\ref{sec}...}}
```

Then the code

```
\azsaved~\ref{sec}...
```

is added to the bookmark, that finally gives correct result.

\aznotshow • The previous problem can be solved with the command **\aznotshow** instead of **\texorpdfstring**. Because by placing **\aznotshow** before **\az** (or any version of it), the result is not shown, only **\azsaved** is generated with the appropriate definite article. Therefore

```
\aznotshow\Az{\ref{sec}}
\section{\azsaved~\ref{sec}...}
```

also gives correct result in the pdf bookmark.

3.2. Abbreviations

```
\aref{⟨key⟩}    ≡ \az{\ref{⟨key⟩}}
\aref*{⟨key⟩}   ≡ \az*{\ref{⟨key⟩}}
\avref{⟨key⟩}   ≡ \azv{\ref{⟨key⟩}}
\avref*{⟨key⟩}  ≡ \azv*{\ref{⟨key⟩}}

\aeqref{⟨key⟩}    ≡ \az{\eqref{⟨key⟩}}
\aeqref*{⟨key⟩}   ≡ \az*{\eqref{⟨key⟩}}
\aveqref{⟨key⟩}   ≡ \azv{\eqref{⟨key⟩}}
\aveqref*{⟨key⟩}  ≡ \azv*{\eqref{⟨key⟩}}

\apageref{⟨key⟩}    ≡ \az{\pageref{⟨key⟩}}
\apageref*{⟨key⟩}   ≡ \az*{\pageref{⟨key⟩}}
\avpageref{⟨key⟩}   ≡ \azv{\pageref{⟨key⟩}}
\avpageref*{⟨key⟩}  ≡ \azv*{\pageref{⟨key⟩}}

\acite[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}    ≡ \az{\cite[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}}
\acite*[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}   ≡ \az*{\cite[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}}
\avcite[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}   ≡ \azv{\cite[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}}
\avcite*[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}  ≡ \azv*{\cite[⟨text⟩]{⟨key1⟩,⟨key2⟩,...}}
```

In the names of the commands, the first letter "a" can be replaced by the letter "A". Then the definite article will begin with a capital letter, which is necessary at the beginning of sentences:

```
\Aref \Avref \Aeqref \Aveqref \Apageref \Avpageref \Acite \Avcite
```

For example

```
\section{Cím}\label{seca}
\section{Cím}\label{secb}
\Aref{seca}.~és \aref{secb}.~szakaszban
```

„Az 1. és a 2. szakaszban".

3.3. The huaz hook

During the process of determining the definite article, the `huaz` package replaces the cross-reference commands with their expandable versions, and the formatting commands (`\emph`, `\textbf`, `\small`, etc.) are ignored. Because of this, it is possible that the following example codes work:

```
\newcommand{\osztaly}{V.B}
Idén \az{\textbf{\osztaly}}~osztály rendezi a farsangot.
```

```
\section{Cím}\label{sec}
\az{\textbf{\ref{sec}}}
```

`\AddToHook{huaz}{⟨code⟩}` • The `huaz` package ignores all the text formatting commands, that is listed in the `huaz` hook. If a formatting command is not in this hook, it can be added by the user. For example

```
\newcommand{\myfont}[1]{{\usefont{T1}{yv1d}{m}{n}#1}}
\newcommand{\mytext}{X.A~osztály}
\az{\myfont{\mytext}}
```

This code returns with an error, since the `\myfont` is not listed in the `huaz` hook. This can be supplemented with the following code:

```
\AddToHook{huaz}{\def\myfont{}}
```

This results that while determining the definite article, the `\myfont` command means nothing. So the following code works perfectly:

```
\AddToHook{huaz}{\def\myfont{}}
\newcommand{\myfont}[1]{{\usefont{T1}{yv1d}{m}{n}#1}}
\newcommand{\mytext}{X.A~osztály}
\az{\myfont{\mytext}}
```

The previous case can be solved without the `huaz` hook in the following way:

```
\newcommand{\myfont}[1]{{\usefont{T1}{yv1d}{m}{n}#1}}
\newcommand{\mytext}{X.A~osztály}
\az*{\mytext}~\myfont{\mytext}
```

If the `\myfont` is included in the `\mytext` definition, the usage of the `huaz` hook is definitely necessary.

```
\AddToHook{huaz}{\def\myfont{}}
\newcommand{\myfont}[1]{{\usefont{T1}{yv1d}{m}{n}#1}}
\newcommand{\mytext}{\myfont{X.A~osztály}}
\az{\mytext}
```

In the following case, it is also necessary to use the `huaz` hook.

```
\AddToHook{huaz}{\def\myfont{}}
\DeclareRobustCommand{\myfont}[1]{{\usefont{T1}{yv1d}{m}{n}#1}}
\renewcommand{\thesection}{\myfont{\arabic{section}}}
\section{Cím}\label{sec}
\aref{sec}
```

Here the `\myfont` is defined as a robust command, because it goes into a moving argument. The case of the previous example rarely occurs, because it is not customary to use a text formatting command when specifying a counter type (Arabic, Roman, etc.).

4. Implementation

We do not publish the complete code here, which is approx. 500 input lines, we just show two essential details.

4.1. Detecting accented characters

All the accented vowels (Hungarian and non-Hungarian) are collected into the `\huaz@list@A` list command. On the other hand, `\huaz@list@X` contains the list of non-Hungarian accented consonants whose pronunciation begin with a vowel (e.g. ń, ñ, ň, ŕ, ř, ś, š, etc.).

The detection of accented characters must be solved in different ways in the following three cases:

Case 1. If the compiler is `pdflatex` and the source file is UTF-8 encoded, then for example, the letter "ö" can be detected with the ASCII characters `^^c3^^b6`, where `c3b6` is the UTF-8 hexadecimal code of "ö". So, with these conditions the definition of the `\huaz@list@A` and `\huaz@list@X` are

```
\def\huaz@list@A{^^c3^^b6,^^c3^^bc,^^c3^^b3,^^c5^^91,^^c3^^ba,^^c3^^a9,...}
\def\huaz@list@X{^^c4^^b9,^^c5^^81,^^c4^^bc,^^c4^^ba,^^c5^^82,^^c5^^85,...}
```

Case 2. If the compiler is `pdflatex` and the source file is ISO-8859-2 encoded, then for example, the letter "ö" can be detected with the ASCII characters `^^f6`, where `f6` is the ISO-8859-2 hexadecimal code of "ö". (Although in this case the accent commands are also suitable for detection.) So then the definition of the `\huaz@list@A` and `\huaz@list@X` are

```
\def\huaz@list@A{^^a1,^^b1,^^c1,^^c2,^^c3,^^c4,^^c9,^^ca,^^cb,^^cc,^^cd,^^ce,...}
\def\huaz@list@X{^^a3,^^a5,^^a6,^^a9,^^aa,^^b3,^^b6,^^b9,^^ba,^^c0,^^c5,^^d1,...}
```

Case 3. If the compiler is `xelatex` or `lualatex` then the source file can only be UTF-8 encoded. In this case for example, the letter "ö" can be detected with the ASCII characters `^^^^00f6`, where `U+00f6` is the Unicode code point of "ö". So, on these terms the definition of the `\huaz@list@A` and `\huaz@list@X` are

```
\def\huaz@list@A{^^^^00f6,^^^^00fc,^^^^00f3,^^^^0151,^^^^00fa,^^^^00e9,...}
\def\huaz@list@X{^^^^0139,^^^^0141,^^^^013c,^^^^013a,^^^^0142,^^^^0145,...}
```

The `\ifpdftex` and the `\inputencodingname` commands are used to distinguish these cases.

4.2. The central inside command of the huaz package

The most important inside command of the **huaz** package is the `\huaz@z{⟨input⟩}`. The effect of this command: The value of `\ifhuaz@must@z@` will be true, if and only if the ⟨input⟩ must be preceded by the definite article "az", otherwise its value will be false. If the `\ifhuaz@must@z@` is true, then the output of the `\huaz@z{⟨input⟩}` is "z", otherwise nothing. The definition of this macro consists of the following major blocks:

Block 1. The first syntax unit of ⟨input⟩ is included in the following list: `\AA,\aa, \AE,\ae,e,u,i,o,a,E,U,I,O,A,5`.

```
\StrChar{#1}{1}[\huaz@temp]%
\@for\huaz@list:={\AA,\aa,\AE,\ae,e,u,i,o,a,E,U,I,O,A,5}%
\do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@must@z@true}{}}%
```

Block 2. The first syntax unit of ⟨input⟩ is an accent command and the second one is one of the following `e,u,i,o,a,E,U,I,O,A`.

```
\huaz@temp@if@false%
\StrChar{#1}{1}[\huaz@temp]%
\@for\huaz@list:={\",\',\H,\`,\~,\^,\v,\u,\=,\k}%
\do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@temp@if@true}{}}%
\ifhuaz@temp@if@%
  \StrChar{#1}{2}[\huaz@temp]%
  \@for\huaz@list:={e,u,i,a,E,U,I,O,A}%
  \do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@must@z@true}{}}%
\fi%
```

Block 3. The first syntax unit of ⟨input⟩ is included in the `\huaz@list@A`.

```
\ifhuaz@must@oneunit@%
  \StrLeft{#1}{1}[\huaz@tempa]%
\else%
  \StrLeft{#1}{2}[\huaz@tempa]%
\fi%
\@for\huaz@list:=\huaz@list@A%
\do{\StrLeft{\huaz@list}{2}[\huaz@tempb]%
    \IfStrEq{\huaz@tempa}{\huaz@tempb}{\huaz@must@z@true}{}}%
```

Block 4. The second syntax unit of ⟨input⟩ (if any) is not a letter, but a number, punctuation mark or space, while the first syntax unit is included in the following list: `f,l,m,n,r,x,y,F,L,M,N,S,R,X,Y` or (in case `latin2`, `xelatex`, `lualatex`) in `\huaz@list@X`.

```
\huaz@temp@if@false%
\StrChar{#1}{2}[\huaz@temp]%
\@for\huaz@list:={;,`,',",+,!,/,=,(,),<,>,@,.,?,:,-,*,0,1,2,3,4,5,6,7,8,9,{,},
                 { },{},\&,\#,\_,\unskip,\kern}%
\do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@temp@if@true}{}}%
\ifhuaz@temp@if@%
  \StrChar{#1}{1}[\huaz@temp]%
  \@for\huaz@list:={f,l,m,n,r,x,y,F,L,M,N,S,R,X,Y}%
  \do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@must@z@true}{}}%
  \ifhuaz@must@oneunit@%
    \ifhuaz@must@z@\else%
      \@for\huaz@list:=\huaz@list@X%
      \do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@must@z@true}{}}%
    \fi%
  \fi%
\fi%
```

Block 5. The third syntax unit of ⟨input⟩ (if any) is not a letter, but a number, punctuation mark or space, the second syntax unit is included in the following list: `f,l,m,n,r,x,y,F,L,M,N,S,R,X,Y` and the first one is an accent command.

```
\huaz@temp@if@false%
\StrChar{#1}{3}[\huaz@temp]%
\@for\huaz@list:={;,`,',",+,!,/,=,(,),<,>,@,.,?,:,-,*,0,1,2,3,4,5,6,7,8,9,{,},
                 { },{},\&,\#,\_,\unskip,\kern}%
\do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@temp@if@true}{}}%
\ifhuaz@temp@if@%
  \huaz@temp@if@false%
  \StrChar{#1}{2}[\huaz@temp]%
  \@for\huaz@list:={f,l,m,n,r,x,y,F,L,M,N,S,R,X,Y}%
  \do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@temp@if@true}{}}%
  \ifhuaz@temp@if@%
    \StrChar{#1}{1}[\huaz@temp]%
    \@for\huaz@list:={\",\',\H,\`,\~,\^,\v,\u,\=,\k}%
    \do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@must@z@true}{}}%
```

```
    \fi%
\fi%
```

Block 6. The third syntax unit of ⟨input⟩ (if any) is not a letter, but a number, punctuation mark or space, and the first two syntax unit is included in the following list: `ly,Ly,LY,ny,Ny,NY,sz,Sz,SZ` or (in case `pdflatex` + `utf8`) in `\huaz@list@X`.

```
\huaz@temp@if@false%
\StrChar{#1}{3}[\huaz@temp]%
\@for\huaz@list:={;,`,',",+,!,/,=,(,),<,>,@,.,?,:,-,*,0,1,2,3,4,5,6,7,8,9,{,},
                { },{},\&,\#,\_,\unskip,\kern}%
\do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@temp@if@true}{}}%
\ifhuaz@temp@if@%
  \StrLeft{#1}{2}[\huaz@temp]%
  \@for\huaz@list:={ly,Ly,LY,ny,Ny,NY,sz,Sz,SZ}%
  \do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@must@z@true}{}}%
  \ifhuaz@must@oneunit@\else%
    \ifhuaz@must@z@\else%
      \@for\huaz@list:=\huaz@list@X%
      \do{\IfStrEq{\huaz@temp}{\huaz@list}{\huaz@must@z@true}{}}%
    \fi%
  \fi%
\fi%
```

Block 7. The first syntax unit of ⟨input⟩ is 1 ("egy") and the second one is not number.

```
\StrChar{#1}{1}[\huaz@temp]%
\IfStrEq{\huaz@temp}{1}{%
  \StrChar{#1}{2}[\huaz@temp]%
  \IfInteger{\huaz@temp}{}{\huaz@must@z@true}%
}{}%
```

Block 8. The first four characters of ⟨input⟩ is a number from 1000 to 1999 ("ezer...") and the fifth one is not number.

```
\StrLen{#1}[\huaz@temp]%
\setcounter{huaz@temp@count}{\huaz@temp}%
\ifnum\value{huaz@temp@count}>3%
  \StrChar{#1}{1}[\huaz@temp]%
  \IfStrEq{\huaz@temp}{1}{%
    \StrMid{#1}{2}{4}[\huaz@temp]%
    \IfInteger{\huaz@temp}{%
      \StrChar{#1}{5}[\huaz@temp]%
      \IfInteger{\huaz@temp}{}{\huaz@must@z@true}%
    }{}%
  }{}%
\fi%
```

Block 9. The first seven characters of ⟨input⟩ is a number from 1000000 to 1999999 ("egymillió...") and the eighth one is not number.

```
\StrLen{#1}[\huaz@temp]%
\setcounter{huaz@temp@count}{\huaz@temp}%
\ifnum\value{huaz@temp@count}>6%
  \StrChar{#1}{1}[\huaz@temp]%
  \IfStrEq{\huaz@temp}{1}{%
    \StrMid{#1}{2}{7}[\huaz@temp]%
    \IfInteger{\huaz@temp}{%
      \StrChar{#1}{8}[\huaz@temp]%
      \IfInteger{\huaz@temp}{}{\huaz@must@z@true}%
    }{}%
  }{}%
\fi%
```

Block 10. The first ten characters of ⟨input⟩ is a number from 1000000000 to 1999999999 ("egymilliárd...") and the eleventh one is not number.

```
\StrLen{#1}[\huaz@temp]%
\setcounter{huaz@temp@count}{\huaz@temp}%
\ifnum\value{huaz@temp@count}>9%
  \StrChar{#1}{1}[\huaz@temp]%
  \IfStrEq{\huaz@temp}{1}{%
    \StrMid{#1}{2}{10}[\huaz@temp]%
    \IfInteger{\huaz@temp}{%
      \StrChar{#1}{11}[\huaz@temp]%
      \IfInteger{\huaz@temp}{}{\huaz@must@z@true}%
    }{}%
  }{}%
\fi%
```

There are still a lot of interesting and important parts of the code. For example, the conversion of the Roman numerals to Arabic, the expandable version of the `\cite` command, replacing the cross-reference commands with their expandable versions, ignoring the formatting commands.

5. Future work

The package has some limitations. It only recognizes the standard cross-reference commands, furthermore it handles the `\cite` command in default case, with `natbib` and `bibtex`, but only in certain cases with `biblatex`. Narrowing these limitations is an important development direction. Of course, at the end of testing, the goal is to publish the package on the The Comprehensive TeX Archive Network, which is the central place for all kinds of material around TeX.

6. Summary

Currently, there are two tools (`magyar.ldf`, `nevelok.sty`) in the LATEX system, with which the user can automatically insert Hungarian definite articles in front of cross-references and macros storing text. Both have numerous errors and omissions. The most serious problem is that the UTF-8 encoded source files are not handled. The `huaz` package provides an alternative to these two packages, that is of course UTF-8 compatible, less buggy, much more flexible and modern.

References

[1] AMERICAN MATHEMATICAL SOCIETY: *User's Guide for the amsmath Package (Version 2.1)*, The Comprehensive TeX Archive Network (2020), URL: https://www.ctan.org/pkg/amsmath.

[2] Z. BODÓ: *nevelok – LATEX package for automatic definite articles for Hungarian*, The Comprehensive TeX Archive Network (2015), URL: https://ctan.org/pkg/nevelok.

[3] L. LAMPORT: *LATEX: A document preparation system, User's guide and reference manual*, 2nd edition, Addison Wesley, 1994, ISBN: ISBN-10: 0201529831, ISBN-13: 978-0201529838.

[4] H. OBERDIEK: *The refcount package*, The Comprehensive TeX Archive Network (2019), URL: https://ctan.org/pkg/refcount.

[5] S. RAHTZ, H. OBERDIEK: *Hypertext marks in LATEX: a manual for hyperref*, The Comprehensive TeX Archive Network (2022), URL: https://ctan.org/pkg/hyperref.

[6] P. SZABÓ: *Typing Hungarian text with MagyarLATEX*, Hungarian (2009), URL: https://math.bme.hu/latex/magyarldf-doc.pdf.

[7] C. TELLECHEA: *xstring User's manual*, The Comprehensive TeX Archive Network (2021), URL: https://ctan.org/pkg/xstring.

[8] THE LATEX3 PROJECT TEAM: *LATEX News, Issues 1–35*, The Comprehensive TeX Archive Network (2022), URL: https://www.latex-project.org/news/latex2e-news/ltnews.pdf.

[9] THE LATEX PROJECT TEAM: *The iftex package*, The Comprehensive TeX Archive Network (2022), URL: https://ctan.org/pkg/iftex.

[10] F. WETTL, GY. MAYER, P. SZABÓ: *LATEX handbook*, Hungarian, Budapest, Hungary: Panem, 2004, ISBN: ISBN 963 545 398 1.