

The **huaz** package

v2.0 (2024/04/28)

Tibor Tómacs

tomacs.tibor@gmail.com

Abstract

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*”.

The **huaz** package helps the user to insert automatically the correct definite article for cross-references and other commands containing text. Thus, if these change, the definite articles will also change accordingly.

The documentation is in Hungarian, because only Hungarian-speaking users are interested.

1. Bevezetés

A magyar nyelvben a határozott névelő egy szó előtt aszerint „*a*” vagy „*az*”, hogy a szó kiejtve mássalhangzóval vagy magánhangzóval kezdődik. Például „*a szék*” vagy „*az asztal*”. Ez egyszerű szabály, de írásban összetett vizsgálatot igényel. Itt felsorolunk néhány nehézséget:

- Ha oldalszámra akarunk hivatkozni, akkor a `\pageref` parancsot szoktuk használni. Ha ez elé kell határozott névelőt rakni, akkor nem mindegy, hogy mi az oldalszám. Például „*az 1. oldal*”, „*a 2. oldal*”, „*az 5. oldal*”, „*a 10. oldal*”. Így egyáltalán nem biztos, hogy az „`a \pageref{<kulcs>}`” kód megfelelő eredményt ad. Hasonló a probléma minden kereszthivatkozás esetén (`\ref`, `\pageref`, `\eqref`, `\cite`).
- Az is fontos, hogy egy kifejezés római szám vagy sem. Például „*az V. fejezet*”, ha *V* római szám, azaz 5 a jelentése. Ugyanakkor „*a V. fejezet*”, ha *V* betűt jelöl vagy alfanumerikus szám esetén a 22 értéket (*V* az angol ábécében a 22. betű).
- Néhány mássalhangzó speciális tulajdonságú. Például „*az M betű*”, de „*a Magyar Közlöny*”; „*az Ny betű*”, de „*a Nyugdíjfolyósító Igazgatóság*”; stb.

Először megemlítünk két eddigi megoldást a határozott névelők automatikus kezelésére, kiemelve azok hibáit és hiányosságait.

1.1. A magyar.ldf

Egyik lehetőség a `babel` csomag `magyar` opcióját beállító `magyar.ldf` fájl használata. Ennek teljes dokumentációja elérhető innen: <https://math.bme.hu/latex/magyarldf-doc.pdf>.

Az ebben definiált `\az`, `\aref`, `\apageref`, `\acite`, stb. parancsok alapvetően jól használhatóak, de van néhány fontos hiba, melyek a **huaz** csomag megírását motiválták:

- Az `\eqref` parancsnak nincs névelős verziója. Helyette az `\aref{\langle kulcs \rangle}` használható, de ez dőlt betűs környezetben nem álló betűs eredményt ad, mint az `\eqref` esetében. Másik lehetőség, ami ezt a problémát megoldja, az `\az{\eqref{\langle kulcs \rangle}}` parancs. Ugyanakkor egyik megoldás sem kezeli a római számozású vagy a `\tag` parancssal címkézett egyenleteket.
- Az `\az{\ref{\langle kulcs \rangle}}` és `\az{\pageref{\langle kulcs \rangle}}` nem kezeli a római számozású hivatkozásokat.
- Az előző hiba akkor is fennáll, amikor az `\az` parancsban egy római számmal kezdődő szöveg van. Például `\az{V.~osztály}` eredménye „*a V. osztály*”.
- Amennyiben nem kereszthivatkozáshoz, hanem egy szöveghez, vagy valamilyen szöveget tároló parancshoz kell automatikus névelő, akkor az ékezetes betűket UTF-8 kódolás esetén rosszul detektálja. A hiba alapvető oka, hogy az UTF-8 kódolású karakterek nem egy, hanem több bajton vannak kódolva, amit a `magyar.ldf` nem vesz figyelembe. Ezért például `\az{ágy}` eredménye „*a ágy*”, mert nem betűként érzékeli az „*á*” betűt, így azt sem tudja, hogy ez magánhangzó. Meglepő módon `\az{száz}` eredménye „*az száz*” is rossz. Ennek az az oka, hogy mivel számára az „*á*” nem betű, ezért az „*sz*” betűt különállónak tekinti, ami elé valóban „*az*” kell.
- Például `\az{\textbf{N betű}}` esetén a `\textbf` parancs megzavarja a szó betűinek detektálását, így a névelő helytelenül „*a*” lesz.

1.2. A nevelok csomag

A `nevelok` csomag szintén az automatikus határozott névelők kezelésére készült. Nem kezeli az ékezetes betűk semmilyen formáját és sok esetben ad kereszthivatkozásoknál rossz névelőt. Az utóbbinak az az oka, hogy közvetlenül a `\ref` és `\pageref` parancsokat vizsgálja, amelyek nem kifejthetőek. Jelenlegi állapotában nem ajánlott használni ezt a csomagot.

1.3. A huaz csomag célja és működése

A **huaz** csomag célja segíteni a felhasználót abban, hogy a kereszthivatkozásokhoz és szöveget tartalmazó parancsokhoz automatikusan beszúrja a megfelelő határozott névelőt. Így ha ezek változnak, akkor a határozott névelők is megfelelően változnak.

A másik cél ezt úgy megtenni, hogy a `magyar.ldf` és a `nevelok.sty` hibáit és hiányosságait korrigálja illetve pótolja.

A **huaz** csomag a következő esetekben illeszt az adott szöveg elé „az” névelőt:

1. Az első betű magánhangzó (legyen az kis- vagy nagybetű, ékezetes vagy sem, magyar vagy sem, UTF-8, ISO-8859-2 karakterrel vagy repülő ékezettel megadva).
2. Az első betű olyan kis vagy nagy mássalhangzó, amit önmagában kiejtve magánhangzóval kezdünk (például F, L, M, stb.) míg a második karakter (ha van) nem betű, hanem szám, írásjel vagy szóköz. Például „M-10”. Ide soroltunk néhány nem magyar ékezetes mássalhangzót is. Például „Ň.1”.
3. Az első két karakter egy olyan kis vagy nagy kétjegyű mássalhangzó, amit önmagában kiejtve magánhangzóval kezdünk (például Ny, Ly, Sz, stb.) míg a harmadik karakter (ha van) nem betű, hanem szám, írásjel vagy szóköz. Például „NY betű”.
4. Az első karakter 5.
5. 1, 4, 7 vagy 10 jegyű számmal kezdődik és az első számjegy 1 (egy, ezer, egymillió, egymilliárd).

Ha a szó elején található karakterek római számként is értelmezhetőek, akkor megválasztható, hogy azt konvertálja arab számra és a névelőt ahhoz határozza meg vagy sem. Például „XII/A” esetén.

2. A huaz csomag használata

A **huaz** csomagot a szokott módon kell betölteni:

```
\usepackage{huaz}
```

Csomagopciók nincsenek. UTF-8 (`utf8`) és ISO-8859-2 (`latin2`) kódolású forrásfájl esetén működik, de a repülő ékezeteket is jól kezeli. Kompatibilis a `pdflatex`, `xelatex` és `lualatex` fordítókkal is. Például `pdflatex` fordítóval a következő betöltés megfelelő:

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

A `hyperref` csomaggal is kompatibilis, így az is betölthető a **huaz** mellé.

A `magyar.ldf` `defaults=hu-min` opciója bekapcsol néhány olyan opciót is, amely elérhetővé teszi a saját automatikus névelőparancsait. Ezeket nem kötelező, de hasznos a **huaz** csomag használata esetén kikapcsolni. Ehhez a

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

sor helyett használja a következőt:

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

A `xelatex` és `lualatex` fordítók esetén `babel` helyett a

```
\usepackage{polyglossia}  
\setdefaultlanguage{hungarian}
```

is használható.

A **huaz** csomag felhasználja az `xstring`, `refcount` és `iftex` csomagok szolgáltatásait, így ezek is betöltődnek. Másrészt felhasználásra kerül néhány 2021. október 10-én bevezetett változás a \LaTeX kerneljében, így csak az ezután telepített rendszereken működik megfelelően a csomag.

2.1. Parancsok

`\az{<szöveg>}`

A `<szöveg>` elé a megfelelő határozott névelő kerül kisbetűs alakban. Ha a `<szöveg>` római számként is értelmezhető (pl. „XII”), esetleg azt írásjel vagy szóköz követi (pl. „XII.A”), akkor a névelő az arab megfelelőjéhez igazodik. Például

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

eredménye „Idén az V.B osztály rendezi a farsangot.”. De

```
\az{VB} \az{V6}
```

eredménye „a VB a V6” lesz, mert a „V” után betű illetve szám következik, ezért nem 5-nek értelmezi, hanem V betűnek.

A `<szöveg>` lehet szöveget tároló parancs is. Például

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

eredménye „Idén az V.B osztály rendezi a farsangot.”.

A `<szöveg>` tartalmazhat szövegformázó parancsokat is (bővebben lásd a 2.3. szakaszban). Például

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

eredménye „Idén az **V.B** osztály rendezi a farsangot.”.

A `<szöveg>` lehet standard keresztshivatozás is (`\ref`, `\ref*`, `\pageref`, `\pageref*`, `\eqref`, `\refeq`, `\cite`). Például

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

eredménye „az 1. szakaszban, a **2.** szakaszban”. Ha a `section` számláló kiírása római számozásra van állítva a `\renewcommand{\thesection}{\Roman{section}}` paranccsal, akkor az előző kód eredménye „az **I.** szakaszban, a **II.** szakaszban”.

Korlátozások

1. A `<szöveg>` elején csak a `\ref`, `\ref*`, `\pageref`, `\pageref*`, `\eqref`, `\refeq`, `\cite` keresztshivatozásokkal működik helyesen. (Az `\eqref` az `amsmath`, a `\refeq` pedig a `mathtools` csomaggal van definiálva.)
2. A `<szöveg>` elején a `\cite` parancs jól működik a `natbib` csomaggal és `bibtex` esetében is. A `biblatex` csomag használata esetén akkor működik jól, ha a `style` vagy `citestyle` opciók értéke `numeric`, `numeric-verb`, `alphabetic`, `alphabetic-verb` vagy `authoryear`. Akkor is jól működik, ha nem adjuk meg a `style` illetve `citestyle` opciók egyikét sem.
3. A pdf vázlatfájába nem lehet `\az` paranccsal szöveget beilleszteni. Tehát például a következő kód nem ad helyes vázlatfát, ha `hyperref` vagy `bookmark` csomagot használ (bár a cím a szövegben, fejlécben és a tartalomjegyzékben jól fog megjelenni).

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

A problémát a később ismertetett `\azsaved` illetve `\aznotshow` parancsokkal lehet megoldani.

`\az*{<szöveg>}`

Ugyanaz mint a `*` nélküli esetben, de ekkor csak a névelő kerül kiírásra. Például, ha az `amsmath` és `hyperref` csomagok használata mellett szeretnénk egy link nélküli egyenlethivatkozást, akkor ezt (mivel nincs `\eqref*` definiálva) például így tehetjük meg:

```
\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \textup{(\ref*{eq})} egyenletet!}
```

Szebb lenne a mondat határozott névelővel a képlet száma előtt, de most nem ad feltétlenül helyes eredményt az

```
\az{\textup{(\ref*{eq})}}
```

kód, mert az első karakter a (jel lesz, és nem a képlet száma. Így minden esetben „a” lesz a névelő. Ezt a következő módon javíthatjuk:

```
\az*{\ref{eq}} \textup{(\ref*{eq})}
```

\azv{<szöveg>}

Ugyanaz mint `\az{<szöveg>}`, de ha a `<szöveg>` elején római számként értelmezhető karakterek vannak, akkor a névelő nem az arab megfelelőjéhez igazodik, hanem mint egyszerű karakterekhez. Például

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

eredménye „az V. szakaszban, a V. szakaszban”, mert az első esetben a V betűt római számként értelmezte, a második esetben pedig nem. Mivel most alfanumerikus számra van beállítva a `section` számláló, ezért a második eset a jó.

\azv*{<szöveg>}

Ugyanaz mint a `*` nélküli esetben, de ekkor csak a névelő kerül kiírásra.

\Az{<szöveg>}

\Az*{<szöveg>}

\Azv{<szöveg>}

\Azv*{<szöveg>}

Ugyanaz, mint a kis „a” betűvel kezdődő parancsok, de ekkor a határozott névelő nagybetűvel kezdődik, ami mondatok elején szükséges. Például

```
\section{Cím}\label{sec}
\Az{\ref{sec}}.~szakaszban áttekintjük a téma rövid történetét.
```

eredménye „Az 1. szakaszban áttekintjük a téma rövid történetét.”.

\azsaved

Az előző parancsok bármelyikét használva, generálódik egy kifejthető `\azsaved` parancs. Ennek eredménye az a határozott névelő amelyik a szó elé kell.

Amikor `hyperref` vagy `bookmark` csomagot használ, akkor (ahogyan azt korábban említettük) a következő kód esetén a cím, fejléc, tartalomjegyzék rendben lesz, de a pdf vázlatfája nem:

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

Ez az `\azsaved` paranccsal a következő módon oldható meg:

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

Ekkor a vázlatfába az

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

kód kerül, ami már helyes eredményt ad.

`\aznotshow`

Az előző probléma a `\texorpdfstring` helyett ezzel a paranccsal egyszerűbben is megoldható. Ugyanis ezt helyezve az `\az` (vagy bármelyik verziója) elé, az eredmény nem jelenik meg, csak az `\azsaved` generálódik le a megfelelő határozott névelővel. Így

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

szintén helyes eredményt ad a vázlatfában is.

`\azseparator{<kód>}`

Ezzel lehet megadni, hogy a határozott névelő és a `<szöveg>` mivel legyen elválasztva. Alapértelmezetten ez a törhetetlen normál méretű szóköz, ami

```
\azseparator{~}
```

módon lett beállítva. Ha át akarjuk állítani például törhető szóközre akkor ez

```
\azseparator{ }
```

módon tehető meg.

2.2. Rövidítések

A `\ref`, `\ref*`, `\eqref`, `\refeq`, `\pageref`, `\pageref*`, `\cite` keresztshivatkozások elé nem csak az előbbi parancsokkal tehetünk határozott névelőt, ugyanis mind-egyiknek van egy rövidített egyparancsos verziója is:

```
\aref{<kulcs>}    ≡ \az{\ref{<kulcs>}}  
\aref*{<kulcs>}  ≡ \az{\ref*{<kulcs>}}  
\avref{<kulcs>}  ≡ \azv{\ref{<kulcs>}}  
\avref*{<kulcs>} ≡ \azv{\ref*{<kulcs>}}  
  
\aeqref{<kulcs>} ≡ \az{\eqref{<kulcs>}}  
\aveqref{<kulcs>} ≡ \azv{\eqref{<kulcs>}}  
  
\arefeq{<kulcs>} ≡ \az{\refeq{<kulcs>}}  
\avrefeq{<kulcs>} ≡ \azv{\refeq{<kulcs>}}
```

```

\apageref{<kulcs>} ≡ \az{\pageref{<kulcs>}}
\apageref*{<kulcs>} ≡ \az{\pageref*{<kulcs>}}
\avpageref{<kulcs>} ≡ \azv{\pageref{<kulcs>}}
\avpageref*{<kulcs>} ≡ \azv{\pageref*{<kulcs>}}

\acite[<szöveg>]{<kulcs1>,<kulcs2>,...} ≡ \az{\cite[<szöveg>]{<kulcs1>,<kulcs2>,...}}
\avcite[<szöveg>]{<kulcs1>,<kulcs2>,...} ≡ \azv{\cite[<szöveg>]{<kulcs1>,<kulcs2>,...}}

```

A parancsok nevében az első „a” betű kicserélhető „A” betűre. Ekkor a határozott névelő nagybetűvel kezdődik, ami mondatok elején szükséges:

```

\Aref \Aref* \Avref \Avref*
\Aeqref \Aveqref
\Arefeq \Avrefeq
\Apageref \Apageref* \Avpageref \Avpageref*
\Acite \Avcite

```

Például

```

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

```

eredménye „Az 1. és a 2. szakaszokban”.

2.3. Ignorált parancsok

Amikor a **huaz** csomag megkeresi a megfelelő határozott névelőt, akkor a szövegben a következő parancsokat figyelmen kívül hagyja:

<code>\bfseries</code>	<code>\itshape</code>	<code>\selectfont</code>	<code>\textsl</code>
<code>\color</code>	<code>\LARGE</code>	<code>\sffamily</code>	<code>\textssc</code>
<code>\emph</code>	<code>\Large</code>	<code>\slshape</code>	<code>\textsw</code>
<code>\em</code>	<code>\large</code>	<code>\small</code>	<code>\texttt</code>
<code>\enquote</code>	<code>\lowercase</code>	<code>\sscshape</code>	<code>\textulc</code>
<code>\fbox</code>	<code>\makebox</code>	<code>\swshape</code>	<code>\textup</code>
<code>\fontencoding</code>	<code>\MakeLowercase</code>	<code>\textbf</code>	<code>\tiny</code>
<code>\fontfamily</code>	<code>\MakeUppercase</code>	<code>\textcolor</code>	<code>\told</code>
<code>\fontseries</code>	<code>\mbox</code>	<code>\textit</code>	<code>\ttfamily</code>
<code>\fontshape</code>	<code>\mdseries</code>	<code>\textmd</code>	<code>\ulcshape</code>
<code>\fontsize</code>	<code>\normalfont</code>	<code>\textnormal</code>	<code>\underline</code>
<code>\footnotesize</code>	<code>\normalsize</code>	<code>\textqq</code>	<code>\uppercase</code>
<code>\framebox</code>	<code>\rmfamily</code>	<code>\textrm</code>	<code>\upshape</code>
<code>\Huge</code>	<code>\scriptsize</code>	<code>\textsc</code>	<code>\usefont</code>
<code>\huge</code>	<code>\scshape</code>	<code>\textsf</code>	

Emiatt lehetséges, hogy a következő kódok jól működnek:

```

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

```

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

De ez is helyes eredményt ad:

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

Azért kapunk helyes eredményt, mert a `\myfont` kifejtésében található `\usefont` szerepel az előző listában. Ha a `\myfont` erős parancsként lett volna definiálva, akkor már nem működne megfelelően az előző kód (lásd később). Ebben az esetben a `\myfont` parancsot is fel kell venni a listára.

`\huazignorecmd[⟨opció⟩]{⟨parancs⟩}`

Az ignoráló listát ezzel lehet bővíteni. A `⟨parancs⟩` helyére az ignorálandó parancsot kell írni az argumentumai és `*` nélkül. Az `⟨opció⟩` 0 és 9 közötti egész szám lehet. Amennyiben ezektől eltérő a beírt `⟨opció⟩`, akkor erre figyelmeztet a fordító.

A használata három esetre bontható.

1. eset ♦ Opció nélkül használva, a `⟨parancs⟩` a névelő megállapításakor törlődik, de az utána írt `*`, opció vagy argumentum nem. Deklarációs parancsok esetén jól használható. Például a `\tiny` parancs így lett besorolva a listába:

```
\huazignorecmd{\tiny}
```

Ugyanez alkalmazható abban az esetben is, ha a `⟨parancs⟩`-nak nincs csillagos verziója sem opciója, továbbá az első `{ }` jelek közé írt argumentuma a figyelembe vett szöveg. Például a `\textrm` ilyen, ezért így lett besorolva:

```
\huazignorecmd{\textrm}
```

A `shadowtext` csomag `\shadowtext` parancsa nincs a listában, ezért a következő kód eredménye hibás lesz:

```
\newcommand{\osztaly}{X.A~osztály}
\az{\shadowtext{\osztaly}}
```

A listára

```
\huazignorecmd{\shadowtext}
```

módon vehető fel, mert a `\shadowtext` parancsoknak sem opciója sem csillagos verziója nincs, továbbá egyetlen argumentuma a szöveg. Tehát a következő kód már megfelelően működik:

```
\huazignorecmd{\shadowtext}
\newcommand{\osztaly}{X.A~osztály}
\az{\shadowtext{\osztaly}}
```

Megjegyezzük, hogy az előző eset így is megoldható:

```
\newcommand{\osztaly}{X.A~osztály}
\az*{\osztaly}~\shadowtext{\osztaly}
```

Ha a `\shadowtext` az `\osztaly` definíciójában szerepel, akkor nem tudjuk megke-
rültni a `\huazignorecmd` használatát az előző módon:

```
\huazignorecmd{\shadowtext}
\newcommand{\osztaly}{\shadowtext{X.A~osztály}}
\az{\osztaly}
```

2. eset ♦ Ha az `<opció>` 0, akkor a `<parancs>` és az utána található első `{` jelig
található kódok mindegyike ignorálódik a névelő megállapításakor. Tehát például

```
\huazignorecmd[0]{\foo}
```

esetén a

```
\foo{...} \foo*{...} \foo[...]{...} \foo*[...]{...}
```

kódokban a

```
\foo \foo* \foo[...] \foo*[...]
```

mindegyike figyelmen kívül lesz hagyva a névelő megállapításakor. A `\makebox`
például

```
\huazignorecmd[0]{\makebox}
```

módon lett besorolva, ezért például

```
\makebox[2cm][s]{szöveg}
```

esetén a `\makebox[2cm][s]` rész törlődik és csak a „szöveg” lesz figyelembe véve.

Ha a formázó parancs erős, akkor a `\huazignorecmd` használata nem kerülhető
meg még akkor sem, ha a listán szereplő parancsokkal épül fel. Például

```
\huazignorecmd[0]{\myfont}
\DeclareRobustCommand{\myfont}[2][cmr]{\usefont{T1}{#1}{m}{n}#2}
\renewcommand{\thesection}{\myfont[yv1d]{\arabic{section}}}
\section{Cím}\label{sec}
\aref{sec}
```

Ebben a példában nem hagyható el a [0], mert a `\myfont`-nak van opciója.

3. eset ♦ Ha az `<opció>` 1 és 9 közötti egész szám, akkor a `\huazignorecmd` parancs a névelő megállapítása során a `<parancs>`-ot és az utána következő első `{` jelig tartó kódokat ignorálja, továbbá ezután az első `<opció>` darab `{ }` jelek közé írt részeket szintén figyelmen kívül hagyja. Tehát például

```
\huazignorecmd[1]{\foo}
```

esetén a

```
\foo{...} \foo*{...} \foo[...]{...} \foo*[...]{...}
```

kódok mindegyike figyelmen kívül lesz hagyva a névelő megállapításakor. Például a `\textcolor` parancs

```
\huazignorecmd[1]{\textcolor}
```

módon lett a listába sorolva, így a

```
\textcolor[RGB]{20,30,40}{szöveg}
```

parancsban a

```
\textcolor[RGB]{20,30,40}
```

kódrész ignorálva lesz, és csak a „szöveg” kerül vizsgálatra a névelő megállapításakor. Ha a

```
\huazignorecmd[2]{\foo}
```

parancsot használjuk, akkor a

```
\foo{...}{...} \foo*{...}{...}  
\foo[...]{...}{...} \foo*[...]{...}{...}
```

kódok mindegyike figyelmen kívül lesz hagyva a névelő megállapításakor. Például a `\fontsize` parancs

```
\huazignorecmd[2]{\fontsize}
```

módon lett a listába sorolva, így a

```
\fontsize{14pt}{14pt}
```

parancs figyelmen kívül lesz hagyva a névelő megállapításakor.

2.4. A keresztivatkozások kifejthető verziói

Ha a szöveg elején keresztivatkozás van, akkor a határozott névelő megállapítása előtt a keresztivatkozás parancsát le kell cserélni annak kifejthető verziójára. A `refcount` csomag két ilyen parancsot definiál: A `\getrefnumber` a `\ref`, `\ref*`, `\eqref`, `\refeq` esetén használható, a `\getpagerefnumber` pedig a `\pageref` és `\pageref*` esetén.

```
\huazrefcmd{<parancs>}
```

Ezzel a paranccsal elérhető, hogy a `<parancs>` helyére beírt parancsot és annak csillagos verzióját a határozott névelő megállapítása során a `\getrefnumber` paranccsal helyettesítse.

A `\ref`, `\ref*`, `\eqref`, `\refeq` parancsok így lettek besorolva a **huaz** csomagban:

```
\huazrefcmd{\ref}
\huazrefcmd{\eqref}
\huazrefcmd{\refeq}
```

Ezzel már az `\eqref*` és `\refeq*` is működne, de ezek a csillagos verziók nincsenek definiálva a csillag nélküli verziókat definiáló `amsmath` illetve `mathtools` csomagokban. Ezt mi is pótolhatjuk, amire a 2.6. alszakaszban láthatunk megoldást.

```
\huazpagerefcmd{<parancs>}
```

Ezzel a paranccsal elérhető, hogy a `<parancs>` helyére beírt parancsot és annak csillagos verzióját a határozott névelő megállapítása során a `\getpagerefnumber` paranccsal helyettesítse.

A `\pageref` és `\pageref*` parancsok így lettek besorolva a **huaz** csomagban:

```
\huazrefcmd{\pageref}
```

Ezekre láthatunk egy alkalmazást a 2.5. alszakaszban.

2.5. Az `\autoref` és `\autopageref` parancsok

Ezeket a parancsokat a `hyperref` csomag definiálja. Ezekkel hivatkozva egy cím-kére, nem csak a sorszám, hanem a sorszámozott egység neve (szakasz, alszakasz, oldal, stb.) is megjelenik link formájában. Ezeknek léteznek csillagos verziók is, melyeknek ugyanaz a hatásuk, de nem generálnak linket.

Magyar nyelvű dokumentumokban nem lehet közvetlenül alkalmazni, mert az angol szabály szerint „1. szakasz” helyett „szakasz 1” lesz az eredménye a következőnek:

```
\section{Cím}\label{sec}
\autoref*{sec}
```

Ezért a használatához először „magyarosítani kell” a következő kóddal a `hyperref` betöltése után a preambulumban:

```

\usepackage{regexpatch}
\makeatletter
\patchcmd{\HyRef@autosetref}{\HyRef@currentHtag}{~}{~}{}
\patchcmd{\HyRef@autosetref}{\null}
    {\null.~\HyRef@currentHtag}{~}{~}{}
\patchcmd*{\HyRef@testreftype}{~}{~}{}{}
\makeatother
\RenewDocumentCommand{\autopageref}{sm}{%
  \IfBooleanTF{#1}
    {\pageref*{#2}.~oldal}
    {\hyperref [{#2}]{\pageref*{#2}.~oldal}}}

```

Ezután már „1. szakasz” lesz az eredmény.

Ha az így módosított `\autoref`, `\autoref*`, `\autopageref`, `\autopageref*` parancsokhoz automatikus határozott névelőt szeretnénk, akkor írjuk be a következőket a **huaz** betöltése után:

```

\huazrefcmd{\autoref}
\huazpagerefcmd{\autopageref}

```

Ezután a következő kód eredménye „Az 1. szakasz az 1. oldalon kezdődik.” lesz:

```

\section{Cím}\label{sec}
\Az{\autoref*{sec}} \az{\autopageref*{sec}on} kezdődik.

```

2.6. Az `\eqref*` és `\refeq*` parancsok

A `\ref` és `\pageref` parancsoknak definiált a csillagos verziójuk is, amely alapesetben nem különbözik a normál verziótól, de a `hyperref` csomaggal használva a csillag nélküli verzió linket generál, míg a csillagos nem.

Az `amsmath` csomag által definiált `\eqref` a `hyperref` csomaggal együtt használva szintén generál linket, de ennek nincs definiálva csillagos verziója, így ekkor a link nélküli verzió nem érhető el közvetlenül. További gond az ilyen link nélküli egyenlethivatkozás előtti automatikus névelő megadása.

Négy lehetséges megoldást mutatunk.

1. A `NoHyper` környezet használatával:

```

\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \begin{NoHyper}\aeqref{eq}\end{NoHyper}
    egyenletet!}

```

2. Az `\az*` parancs használatával:

```
\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \az*\ref{eq} \textup{(\ref*{eq})} egyenletet!}
```

3. Az `\eqrefstyle` parancs definiálásával és a `\huazignorecmd` paranccsal, amely az előzőnek egy kényelmesebben használható verziója:

```
\makeatletter
\newcommand{\eqrefstyle}[1]{\textup{\tagform@{#1}}}
\makeatother
\huazignorecmd{\eqrefstyle}

\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \az{\eqrefstyle{\ref*{eq}}} egyenletet!}
```

4. Ebben az esetben definiáljuk az `\eqref` csillagos verzióját (az `amsmath` betöltése után), amely nem generál linket a `hyperref` csomaggal együtt használva:

```
\makeatletter
\DeclareDocumentCommand{\eqref}{ s m }{%
  \textup{\tagform@{
    \IfBooleanTF{#1}{\ref*{#2}}{\ref{#2}}}%
  }}%
}
\makeatother
```

Ezután a

```
\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \az{\eqref*{eq}} egyenletet!}
```

kód helyes eredményt ad, mert a **huaz** csomagban nem csak az `\eqref`, hanem az `\eqref*` is a kifejthető `\getrefnumber` parancsra lesz kicserélve a határozott névelő meghatározásánál.

Ugyanez a probléma a `mathtools` csomag által definiált `\refeq` és a `hyperref` csomag együttes használata esetében.

Az előző négy megoldás mindegyikét lehet alkalmazni megfelelő módosításokkal:

1. A NoHyper környezet használatával:

```
\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \begin{NoHyper}\arefeq{eq}\end{NoHyper}
      egyenletet!}
```

2. Az \az* parancs használatával:

```
\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \az*\ref{eq} \textup{\ref*{eq}} egyenletet!}
```

3. A \refeqstyle parancs definiálásával:

```
\newcommand{\refeqstyle}[1]{\textup{#1}}

\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \az{\refeqstyle{\ref*{eq}}} egyenletet!}
```

Itt nincs szükség a \huazignorecmd{\refeqstyle} használatára.

4. A \refeq* parancs definiálásával:

```
\DeclareDocumentCommand{\refeq}{ s m }{%
  \textup{%
    \IfBooleanTF{#1}{\ref*{#2}}{\ref{#2}}%
  }%
}
```

Ezután a

```
\begin{equation}\label{eq}
a=b
\end{equation}
\emph{Lásd \az{\refeq*{eq}} egyenletet!}
```

kód helyes eredményt ad, mert a **huaz** csomagban nem csak a \refeq, hanem a \refeq* is a kifejthető \getrefnumber parancsra lesz kicserélve a határozott névelő meghatározásánál.

2.7. Parancsok átdefiniálása a határozott névelő megállapításakor

A `\huazignorecmd`, `\huazrefcmd` és `\huazpagerefcmd` parancsok is a címben szereplő funkciót látják el, de speciális esetekben. Ezek általánosított verziója a következő parancs:

```
\huazdefcmd{<átdefiniáló kód>}
```

Az `<átdefiniáló kód>` helyére egy parancsnak azt a definícióját kell írni, amelynek a határozott névelő megállapítása során kell érvényesülnie. Például a következő kódokban az egymást követő sorok ekvivalensek:

```
\huazignorecmd{\tiny}
\huazdefcmd{\def\tiny{}}
```

```
\huazignorecmd[0]{\makebox}
\huazdefcmd{\def\makebox#1#{}}
```

```
\huazignorecmd[1]{\textcolor}
\huazdefcmd{\def\textcolor#1#{\@gobble}}
```

```
\huazrefcmd{\ref}
\huazdefcmd{\def\ref#1#{\getrefnumber}}
```

Alkalmazásként tekintsük a következő kódot, amely hibával fog fordulni:

```
\newcommand{\tananyag}{\LaTeX\ alapjai}
A mai óra anyaga \az{\tananyag}.
```

A megfelelő eredményhez a `\LaTeX` parancsot át kell definiálni a névelő megállapításakor:

```
\huazdefcmd{\def\LaTeX{latex}}
\newcommand{\tananyag}{\LaTeX\ alapjai}

A mai óra anyaga \az{\tananyag}.
```