



The **huaz** package

v3.1 (2026/10/08)

Tibor Tómacs

tomacs.tibor@gmail.com

Abstract

There are two definite articles in Hungarian, “a” and “az”, depending on the first sound of the word that follows. If the word begins with a vowel sound, the definite article is “az”, otherwise it is “a”. The **huaz** package helps the user to automatically insert the correct definite article for cross-references and other commands containing text. This way, if the number or text changes, the definite articles will change accordingly.

Another service offered by the package is the automatic suffixing of numbers and cross-references, also based on their pronunciation.

The documentation of the **huaz** package is in Hungarian, as it is assumed that 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 „a Magyar Közlöny” de „az M betű”; „a Nyugdíjfolyósító Igazgatóság” de „az Ny betű”; stb.

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

A magyar.ldf. Egyik lehetőség a magyar.ldf fájl használata, amit a babel csomag magyar opciójával lehet betölteni. 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 használható az \aref({\langle kulcs \rangle}), 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, mivel azok nem egy, hanem több bájtton 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 oka, hogy számára az „á” nem betű, ezért az „sz” betűt különállónak tekint, 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.

A nevelok csomag. A nevelok csomag szintén a határozott névelők automatikus beillesztéséhez készült. Nem kezeli az ékezetes betűk semmilyen formáját és sok esetben ad kereszthivatkozásoknál rossz névelőt, ugyanis közvetlenül a \ref és \pageref parancsokat vizsgálja, amelyek nem kifejthetők. Ezért ennek a csomagnak a jelenlegi verzióját nem szabad használni!

A **huaz** csomag segíti 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.

Fontos cél volt 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 kódolással vagy ékezet paranccsal (`\'`, `\"`, `\H`, stb.) 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 „SZ 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 „V. emelet” esetén.

A **huaz** csomag másik szolgáltatása a számok illetve kereszthivatkozások automatikus toldalékolása, amely szintén a kiejtéstől függ. Megjegyezzük, hogy erre a `magyar.ldf` fájl `\told` parancsának használatával is van lehetőség.

2. A **huaz** csomag használata

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

```
\usepackage{huaz}
```

A csomag kompatibilis a `latex`, `pdflatex`, `xelatex` és `lualatex` fordítókkal is, továbbá felhasználja az `xstring`, `refcount` és `iftex` csomagok szolgáltatásait, így ezek is betöltődnek.

A `babel` csomag előtt és után is betölthető, a `magyar.ldf` által definiált névelőparancsokat minden esetben átdefiniálja.

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{V4}
```

eredménye „a VB, a V4” lesz, mert V után betű illetve szám következik, ezért nem 5-nek, hanem V betűnek értelmezi mindkét esetben.

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 2.2. alszakaszban). 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 kereszthivatkozás is. 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”.

A `<szöveg>` elején található kereszthivatkozásokra néhány korlátozást figyelembe kell venni:

1. A `<szöveg>` elején a `\ref`, `\ref*`, `\pageref`, `\pageref*`, `\eqref`, `\refeq`, `\refstruc` kereszthivatkozás-parancsokkal működik helyesen. Az `\eqref` az `amsmath`, a `\refeq` a `mathtools`, a `\refstruc` pedig a `magyar.ldf` csomaggal van definiálva.
2. A `<szöveg>` elején a `\cite` jól működik a `natbib` csomaggal és `bibtex` esetében is.

A `biblatex` csomaggal a `\cite`, `\citeauthor`, `\textcite`, `\citeyear` akkor működik jól, ha a `style` vagy `citestyle` opciók értéke:

```

numeric
numeric-verb
alphabetic
alphabetic-verb
authoryear

```

Akkor is jól működik, ha nem adjuk meg a `style` illetve `citestyle` opciók egyikét sem.

`\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 `\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 \dots
```

eredménye „Az 1. szakaszban ...”.

A `\ref`, `\ref*`, `\eqref`, `\refeq`, `\pageref`, `\pageref*`, `\cite` kereszthivatkozások elé nem csak az előbbi parancsokkal tehetünk határozott névelőt, ugyanis mindegyiknek 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 \Aeqref
\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”.

```
\azsaved
```

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

```
\aznotshow
```

Ezt helyezve az `\az` (vagy bármelyik verziója) elé, az eredmény nem jelenik meg, csak az `\azsaved` generálódik le a megfelelő határozott névelővel.

`\huazlike{<kiírt szöveg>}{<figyelembe vett szöveg>}`

Néha előfordul, hogy az eddigi szabályokkal nem kapunk helyes eredményt. Például

```
\az{SMARAGD}
```

esetben az eredmény „a SMARAGD” helyes, de

```
\az{SMS}
```

eredménye „a SMS” helytelen, hiszen „SMS” kiejtése „es em es”. Ilyenkor használható az

```
\az{\huazlike{SMS}{S.M.S.}}
```

parancs, ami az „S.M.S.”-hez határozza meg a névelőt, de utána az „SMS” szót írja ki. Tehát ekkor az eredmény a helyes „az SMS” lesz.

`\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.

`\husuffix{<szám>}{<toldalék>}`

Ezzel számok illetve kereszthivatkozások automatikus todalékolása oldható meg. A `<szám>` lehet sztring vagy kifejthető parancs, de lehetnek `\ref`, `\ref*`, `\pageref`, `\pageref*`, `\eqref`, `\refeq` kereszthivatkozás-parancsok is, feltéve hogy ezek kifejtésében az utolsó karakter arab számjegy.

A `<toldalék>` helyére a következő lehetséges todalékok írhatók be:

as es os (hányas)
at et ot (hányat)
val vel (hányval)
tol (hánytól)
rol (hányról)
bol (hányból)
ba be (hányba)
ban ben (hányban)
ra re (hányra)
nak nek (háynak)

nal nel (hánynál)
hoz hez (hányhoz)

Zárójelben a toldalékhoz tartozó kérdőszó található. Az egy sorban található alakok bármelyike használható és akár nagybetűvel is írható. Például, ha a `\ref{x}` eredménye 5, akkor

```
\husuffix{\ref{x}}{hez},  
\husuffix{\ref{x}}{at},  
\husuffix{\eqref{x}}{BEN}
```

eredménye „5-höz, 5-öt, (5)-ben”.

```
\ahusuffix{⟨szám⟩}{⟨toldalék⟩} ≡ \az{\husuffix{⟨szám⟩}{⟨toldalék⟩}}  
\Ahusuffix{⟨szám⟩}{⟨toldalék⟩} ≡ \Az{\husuffix{⟨szám⟩}{⟨toldalék⟩}}
```

Például

```
\Ahusuffix{\eqref{x}}{ban} definiált\dots
```

eredménye „Az (5)-ben definiált...”.

`\husuffixsaved`

Az előtte található `\husuffix`, `\ahusuffix`, `\Ahusuffix` parancsok által generált toldalékot tároló kifejthető parancs. Például

```
\husuffix{5}{bol} [5\husuffixsaved]
```

eredménye „5-ből [5-ből]”.

`\huazsilent{⟨kód⟩}`

Végrehajtja a `⟨kód⟩`-ot, de nem jeleníti meg az eredményt. Például

```
\huazsilent{\ahusuffix{5}{bol}}
```

nem jelenít meg semmit, de legenerálja a megfelelő névelőt illetve toldalékot és elmenti azokat a kifejthető `\azsaved` és `\husuffixsaved` parancsokba. Tehát ezután

```
\azsaved~5\husuffixsaved
```

eredménye „az 5-ből” lesz.

`\huazdef`

Ekvivalens a `\protected@edef` parancssal. A `\huazsilent` és `\huazdef` parancsok használatára a 4. szakaszban mutatunk példát.

2.2. Ignorált parancsok

Amikor a **huaz** csomag megkeresi a megfelelő határozott névelőt, akkor a szövegben a következő parancsokat a nem szöveget tároló argumentumaikkal együtt (ha van) figyelmen kívül hagyja:

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

2.3. A kereszthivatkozások kifejthető verziói

Ha a szöveg elején kereszthivatkozás van, akkor a határozott névelő megállapítása előtt a kereszthivatkozás 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 3.4. 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}
```

2.4. 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 két egyenértékű parancs:

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

Az `<átdefiniáló kód>` helyére egy parancsnak azt a definícióját kell írni, amelynek a határozott névelő megállapítása során kell érvényesülnie. Felsorolunk néhány kódpárt, melyek ekvivalensek egymással:

```
\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}.
```

Az eredmény „A mai óra anyaga a `LaTeX`.” lesz. Érdemes megemlíteni, hogy mind ezt a `\huazlike` paranccsal is meg lehet oldani a következő módon:

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

3. Néhány parancs használata a **huaz** csomaggal

3.1. 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}
```

A használatához először magyarosítani kell a következő kóddal a `hyperref` betöltése után a preambulumban:

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

Ezután már „1. szakasz” lesz az eredmény.

Az így módosított `\autoref`, `\autoref*`, `\autopageref`, `\autopageref*` parancsokhoz már lehet automatikus határozott névelőt is használni, ugyanis a **huaz** a névelő megállapításakor ezeket is kicseréli a kifejthető verziókra a 2.3. alszakaszban leírt módon. Így a következő kód eredménye „Az 1. szakasz az 1. oldalon kezdődik.” lesz:

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

3.2. A `\refstruc` parancs

A `magyar.1df` által definiált `\refstruc` parancs hasonló célokat szolgál mint az `\autoref`. Ez annyiban tud többet, hogy automatikus toldalékolása is lehetséges

```
\refstruc{<címke>+<toldalék>}
```

módon. Automatikus határozott névelőt a **huaz** használata nélkül

```
\az+\refstruc{<címke>+<toldalék>}
```

szintaxissal lehet megadni. A **huaz** ezt a módot nem támogatja, ehelyett használja a következőt:

```
\az{\refstruc{<címke>+<toldalék>}}
```

3.3. A \told, \atold és \Atold parancsok

Ezeket a parancsokat a magyar.ldf definiálja automatikus számtoldalékolás céljából. A használatának szintaxisa:

```
\told<szám>+<toldalék>{ }
```

Amennyiben nem kapcsolja be a magyar.ldf az=no opcióját, akkor automatikus határozott névelőt a következő módon rakhat elé függetlenül attól, hogy a **huaz** csomagot betöltötte-e vagy sem:

```
\atold<szám>+<toldalék>{ }  
\az{\told<szám>+<toldalék>{ }}  
\Atold<szám>+<toldalék>{ }  
\Az{\told<szám>+<toldalék>{ }}
```

3.4. Az \eqref* és \refeq* parancsok

A \ref és \pageref parancsoknak definiált a csillagos verziójuk is, amely alap esetben 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.

4. A **huaz** csomag használata a PDF könyvjelzőjében

A **huaz** csomag parancsait a `\part`, `\chapter`, `\section`, `\subsection`, stb. parancsokban használva a PDF-ben helyes eredményt kapunk, de `hyperref` vagy `bookmark` csomaggal együtt használva a PDF könyvjelzője nem lesz helyes a következő esetekben:

1. `\part` vagy `\chapter` parancsokba írunk automatikus határozott névelő vagy automatikus toldalék parancsot.

2. `\section`, `\subsection`, stb. parancsokba írunk legalább két automatikus határozott névelő vagy legalább két automatikus toldalék parancsot.

A könyvjelzőhöz kifejthető parancsokat kell alkalmazni, ezért az automatikus határozott névelő esetében az `\azsaved`, míg toldalék esetében a `\husuffixsaved` tárolóparancs kerül a bookmarkba. Ez jól működik `\section`, `\subsection`, stb. esetén, mert a tárolóparancs legenerálódik mire a bookmarkba kerül. Viszont `\part` és `\chapter` esetében ezek legenerálása előtt kerülnek a bookmarkba. Ez okozza az 1. problémát.

Amikor több automatikus névelő vagy toldalék parancs is van egy strukturális parancsban, akkor az utolsó által generált `\azsaved` illetve `\husuffixsaved` érvényesül a korábbiakban is. Ez okozza az 2. problémát.

Az 1. megoldásához gondoskodni kell arról, hogy a tárolóparancsok még a `\part` illetve `\chapter` előtt létrejőjenek. Ezt a következő példákban látható módon lehet megtenni:

```
\huazsilent{\Aref{x}}
\part{\Aref{x} ...}
```

```
\huazsilent{\Ahusuffix{\ref{x}}{ban}}
\chapter{\Ahusuffix{\ref{x}}{ban} ...}
```

A 2. probléma a következő példákban látható módon oldható meg:

```
\huazsilent{\Aref{x}}
\huazdef\tempa{\azsaved~\ref{x}}
\huazsilent{\aref{y}}
\huazdef\tempb{\azsaved~\ref{y}}
\chapter{\tempa\ ... \tempb\ ...}
```

Ha például az `\Aref{x}` eredménye „Az 1” illetve az `\aref{y}` eredménye „a 3” lenne, akkor az előző kód esetén a PDF-ben és annak bookmarkjában is a helyes „Az 1 ... a 3 ...” lesz.

```
\huazsilent{\Ahusuffix{\ref{x}}{bol}}
\huazdef\tempa{\azsaved~\ref{x}\husuffixsaved}
\huazsilent{\ahusuffix{\ref{y}}{bol}}
\huazdef\tempb{\azsaved~\ref{y}\husuffixsaved}
\section{\tempa\ ... \tempb\ ...}
```

Ha például az `\Ahusuffix{\ref{x}}{bol}` eredménye „Az 1-ből” illetve az `\ahusuffix{\ref{y}}{bol}` eredménye „a 3-ból” lenne, akkor az előző kód esetén a PDF-ben és annak bookmarkjában is a helyes „Az 1-ből ... a 3-ból ...” lesz.