The **secnum** package

Gau, Syu

*Last Update: 2020/02/02*

**Abstract**

The package **secnum** provides a macro `\setsecnum` which allows user to format section numbering intuitively.

**Contents**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Usage</td>
</tr>
<tr>
<td>B</td>
<td>Process</td>
</tr>
<tr>
<td>C</td>
<td>Implementation</td>
</tr>
</tbody>
</table>

**A Usage**

Before using the macro, load the package in preamble.

\usepackage{secnum}

Then, one can format the section numbering by using the macro `\setsecnum` in preamble.

\setsecnum{⟨num format⟩}

A typical `⟨num format⟩` is like this:

1.1.1

It consists of some syntax abbrs of numbering formats, reffering the follows,

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>a</td>
</tr>
<tr>
<td>I</td>
<td>i</td>
</tr>
<tr>
<td>l</td>
<td>l</td>
</tr>
</tbody>
</table>

| \Alph | \alph | \Roman | \roman | \arabic |

and some separators, which can be any character except the abbrs and special characters such as barces “Ω”, comma “,”, space “¯”, etc. 
B Process

The process of the macro \texttt{setsecnum} can be explained as follows.

Step 1. The main function eats the input, saying \texttt{I.1.a}, and stores it in a token list.

Step 2. Replace abbrs by macros. In our example, it results “\texttt{Roman.\arabic.\alph}”

Step 3. Split this token list into a sequence by macros. In our example, it results “\texttt{Roman}”, “\texttt{\arabic}” and “\texttt{.\alph}”.

Step 4. Store those codes in indivdual containers.

Step 5. Use them to renew \texttt{thesection}, \texttt{thesubsection}, \texttt{thesubsubsection} etc. provided there is no \texttt{chapter}.

C Implementation

The following is the implementation. Users can ignore.

Preparations

This package uses \LaTeX{}X3. Therefore, the packages expl3 and \texttt{xparse} are needed and should use \texttt{\ProvidesExplPackage} rather than \texttt{\ProvidesPackage}.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesExplPackage{secnum}{2020/02/02}{}{An intuitive way to format section numbering}
\RequirePackage{xparse}
```

\texttt{l__syu_secnum_tl} The two variables are used to store the formatting information.
\texttt{l__syu_secnum_seq} The following variables are used to store the individal formatting codes.

\texttt{g__syu_chapter_tl} This \texttt{(integer)} encodes if \texttt{thechapter} is defined.
\texttt{g__syu_section_tl} If \texttt{thechapter} is defined, it is 1.
\texttt{g__syu_subsection_tl} Otherwise, it is 0.
\texttt{g__syu_subsubsection_tl}
\texttt{g__syu_paragraph_tl}
\texttt{g__syu_subparagraph_tl}
\texttt{g__syu_if_thechapter_int} If \texttt{\if_cs_exist:N thechapter}
\texttt{\int_gset:Nn g__syu_if_thechapter_int 1}\texttt{\else:}
\texttt{\int_gset:Nn g__syu_if_thechapter_int 0}\texttt{\fi:}
Main function

\setsecnum
Here is the definition of the main function \setsecnum.

\DeclareDocumentCommand{\setsecnum}{m}{
    \tl_set:Nn \l__syu_secnum_tl {#1}
}

Store the input in.

\tl_set:Nn \l__syu_secnum_tl {#1}

Replace syntax abbrs by corresponding macros.

\__syu_secnum_unabbr:N \l__syu_secnum_tl

Split into a sequence by macros.

\__syu_split_by_macros:NN \l__syu_secnum_tl \l__syu_secnum_seq

Read formatting information.

\__syu_secnum_from_seq:N \l__syu_secnum_seq

Set the seconumdepth and tocdepth.

\setcounter{seconumdepth}{ \seq_count:N \l__syu_secnum_seq }
\setcounter{tocdepth}{ \seq_count:N \l__syu_secnum_seq }

Format numberings.

\__syu_secnum:

Unabbravation

\__syu_secnum_unabbr:N
This function replace the abbrs in a ⟨tl var⟩ by expansions.

\cs_new_protected:Npn \__syu_secnum_unabbr:N #1 {
    \regex_replace_all:nnN {A} {\c{Alph}} #1
    \regex_replace_all:nnN {a} {\c{alph}} #1
    \regex_replace_all:nnN {I} {\c{Roman}} #1
    \regex_replace_all:nnN {i} {\c{roman}} #1
    \regex_replace_all:nnN {1} {\c{arabic}} #1
}

Split to sequence

\__syu_split_by_macros:NN
This function split a ⟨tl var⟩ into a ⟨sequence⟩ by macros.

\cs_new_protected:Npn \__syu_split_by_macros:NN #1 #2 {
    \tl_clear:N \l_tmpa_tl
    \seq_clear:N #2
    \tl_map_inline:Nn #1 {
        \tl_put_right:Nn \l_tmpa_tl ##1
        \__syu_if_macro:nT ##1 {
            \seq_put_right:NV #2 \l_tmpa_tl
            \tl_clear:N \l_tmpa_tl
        }
    }
}

3
But how to see if an (item) in the token list is a macro?

\__syu_macro_tl
This \texttt{tl var} stores the first five characters of the meaning of any macro, i.e. \texttt{macro} (watch out its catcode). The idea is to creat a \texttt{tl var} and then set its value to be the first five characters of its meaning.

\begin{verbatim}
\tl_new:N \__syu_macro_tl
\tl_set:Nx \__syu_macro_tl { \meaning \__syu_macro_tl }
\tl_gset:Nx \__syu_macro_tl { \tl_range:Nnn \__syu_macro_tl {1}{5} }
\end{verbatim}

\texttt{\__syu_if_macro:nT} \texttt{\__syu_if_macro:nF} \texttt{\__syu_if_macro:nTF}
Then, define a conditional testing if the input is a macro. Note that I use \texttt{if\_meaning} rather than \texttt{\tl_if_eq:NNTF}.

\begin{verbatim}
\prg_new_protected_conditional:Npnn \__syu_if_macro:n #1 { T , F , TF }
{ \group_begin: \tl_set:Nx \l_tmpa_tl { \meaning #1 }
  \tl_set:Nx \l_tmpa_tl { \tl_range:Nnn \l_tmpa_tl {1}{5} }
\exp_after:wN \group_end:
  \if_meaning:w \l_tmpa_tl \__syu_macro_tl
    \prg_return_true:
  \else:
    \prg_return_false:
  \fi: }
\end{verbatim}

\texttt{\__syu_secnum_from_seq:N}
Read the formatting info from given \texttt{sequence}.

\begin{verbatim}
\cs_new_protected:Npn \__syu_secnum_from_seq:N #1
{ \tl_gset:Nx \g__syu_chapter_tl { \seq_item:Nn #1 { \g__syu_if_thechapter_int } }
  \tl_gset:Nx \g__syu_section_tl { \seq_item:Nn #1 { 1 + \g__syu_if_thechapter_int } }
  \tl_gset:Nx \g__syu_subsection_tl { \seq_item:Nn #1 { 2 + \g__syu_if_thechapter_int } }
  \tl_gset:Nx \g__syu_subsubsection_tl { \seq_item:Nn #1 { 3 + \g__syu_if_thechapter_int } }
  \tl_gset:Nx \g__syu_paragraph_tl { \seq_item:Nn #1 { 4 + \g__syu_if_thechapter_int } }
  \tl_gset:Nx \g__syu_subparagraph_tl { \seq_item:Nn #1 { 5 + \g__syu_if_thechapter_int } }
}
\end{verbatim}

Read formatting info
Formatting

`\__syu_secnum`: Formatting section numbering.

```latex
\cs_new:Nn \__syu_secnum:n
{\__syu_secnum:n}

When `\thechapter` is defined, start from it.

```latex
\if_cs_exist:N \thechapter
  { % g__syu_chapter_tl (chapter)
    \renewcommand*{\thechapter}
    % g__syu_section_tl (section)
    \renewcommand*{\thesection}
    \{ \thechapter
    \{ g__syu_section_tl (section) \}
\else
  \renewcommand*{\thesection}
  \{ g__syu_section_tl (section) \}
\fi:
```

Otherwise start from `\thesection`.

```latex
\else:
  \renewcommand*{\thesection}
  \{ g__syu_section_tl (section) \}
\fi:
```

The rest levels.

```latex
\renewcommand*{\thesubsection}
  \{ \thesection
    \{ g__syu_subsection_tl (subsection) \}
  \renewcommand*{\thesubsubsection}
    \{ \thesubsection
      \{ g__syu_subsubsection_tl (subsubsection) \}
  \renewcommand*{\theparagraph}
    \{ \thesubsubsection
      \{ g__syu_paragraph_tl (paragraph) \}
  \renewcommand*{\thesubparagraph}
    \{ \theparagraph
      \{ g__syu_subparagraph_tl (subparagraph) \}
\end{verbatim}

\end{flushleft}

}\end{verbatim}

\end{document}