The **subref** package

Jesse A. Tov

`tov@ccs.neu.edu`

This document corresponds to **subref** v0.2, dated 2011/04/06.

1 Introduction

The purpose of this package is to make it easy to refer to labels nested within hierachies of references. For example, suppose we declare theorem with multiple parts and subparts:

```
\begin{thm}\labeli[thm:ex] This theorem has four parts:
  \begin{enumerate}
    \item\labelii[thm:ex-p1] This is part 1.
    \item\labelii[thm:ex-p2] This is part 2.
    \item\labelii This is part 3, which has two subparts:
      \begin{enumerate}
        \item\labelii[thm:ex-first] This is the first sub-part.
        \item\labelii[thm:ex-second] This is the second sub-part.
        \end{enumerate}
      \item\labelii[thm:ex-p4] This is part 4.
  \end{enumerate}
\end{thm}
Now we can refer to the whole theorem (\ref{thm:ex}) or a part
(\ref{thm:ex-p2}) or even a subpart (\ref{thm:ex-first}).
```

We use the commands `\labeli`, `\labelii`, `\labeliii`, and `\labeliv` to label up to four levels of structure. These commands need to be used in the proper order, because, for example, `\labelii` saves information that `\labeliii` uses to generate three-level labels. The commands take an optional argument which assigns a name to the label, but it can be omitted, or the normal `\label` command may be used. That is, `\labeli[labname]` is equivalent to `\labeli\label{labname}`.

Note that we used `\labelii` for items in both the main and nested `enumerate` environments. This is because nested `enumerate`s already include multiple levels of item names in the references that they produce.

The output of the above example is:

> **Theorem 1.** *This theorem has four parts:*
>
> > *1. This is part 1.*

2. This is part 2.

3. This is part 3, which has two subparts:

    (a) This is the first sub-part.

    (b) This is the second sub-part.

4. This is part 4.

Now we can refer to the whole theorem (1) or a part (1.2) or even a subpart (1.3a).

2   Command Reference

\labeli [⟨*label-name*⟩,...]
\labelii [⟨*label-name*⟩,...]
\labeliii [⟨*label-name*⟩,...]
\labeliv [⟨*label-name*⟩,...]

These macros keep track of nested references. Each sets the current reference, which is captured by \label, to include all levels up to the current level. For example, \labeliii sets the current reference to include the reference saved by the closest prior call to \labelii and appends to that the reference at the current location. Each saves the current reference for use by the next level.

Each takes an optional argument which is a list of label names to use for the current reference. This is merely a convenience:

$$\labeln[⟨\textit{label-name}⟩_1, \ldots, ⟨\textit{label-name}⟩_k] =$$
$$\labeln\label\{⟨\textit{label-name}⟩_1\} \ldots \label\{⟨\textit{label-name}⟩_k\}$$

\fmtsublabeli {⟨*label-segment*⟩}
\fmtsublabelii {⟨*label-segment*⟩}
\fmtsublabeliii {⟨*label-segment*⟩}
\fmtsublabeliv {⟨*label-segment*⟩}

These are used by subref to format each segment of a multiple-level label. Redefine them to change how labels are rendered. For example, the default rendering of a two level label is 2.3, but to get 2(3) instead, write:

```
\renewcommand{\fmtsublabelii}[1]{(#1)}
```

3   Implementation

\fmtsublabeli   These are used to style macro references at levels 1 through 4:

\fmtsublabelii

\fmtsublabeliii

\fmtsublabeliv

```
1 \providecommand*\fmtsublabeli[1]{#1}
2 \providecommand*\fmtsublabelii[1]{.#1}
3 \providecommand*\fmtsublabeliii[1]{.#1}
4 \providecommand*\fmtsublabeliv[1]{.#1}
```

`\labeli` Captures the current label as the level i label. Higher label levels are undefined, so that we don't get weird behavior such as mixing multiple levels. The current label is defined to be the level i label only.

```
 5 \newcommand\labeli{%
 6   \global\let\subref@labeli\@currentlabel
 7   \global\let\subref@labelii\@undefined
 8   \global\let\subref@labeliii\@undefined
 9   \global\let\subref@labeliv\@undefined
10   \global\def\@currentlabel{%
11     \string\fmtsublabeli{\subref@labeli}%
12   }%
13   \subref@optlabels
14 }
```

`\labelii` Captures the current label as the level ii label. Higher label levels are undefined, so that we don't get weird behavior such as mixing multiple levels. The current label is defined to be the level i and ii labels together.

```
15 \newcommand\labelii{%
16   \global\let\subref@labelii\@currentlabel
17   \global\let\subref@labeliii\@undefined
18   \global\let\subref@labeliv\@undefined
19   \global\def\@currentlabel{%
20     \string\fmtsublabeli{\subref@labeli}%
21     \string\fmtsublabelii{\subref@labelii}%
22   }%
23   \subref@optlabels
24 }
```

`\labeliii` Like `\labelii`, but captures three label levels.

```
25 \newcommand\labeliii{%
26   \global\let\subref@labeliii\@currentlabel
27   \global\let\subref@labeliv\@undefined
28   \global\def\@currentlabel{%
29     \string\fmtsublabeli{\subref@labeli}%
30     \string\fmtsublabelii{\subref@labelii}%
31     \string\fmtsublabeliii{\subref@labeliii}%
32   }%
33   \subref@optlabels
34 }
```

`\labeliv` Like `\labelii`, but captures four label levels.

```
35 \newcommand\labeliv{%
36   \global\let\subref@labeliv\@currentlabel
37   \global\def\@currentlabel{%
38     \string\fmtsublabeli{\subref@labeli}%
39     \string\fmtsublabelii{\subref@labelii}%
40     \string\fmtsublabeliii{\subref@labeliii}%
41     \string\fmtsublabeliii{\subref@labeliv}%
42   }%
```

```
43    \subref@optlabels
44 }
```

`\subref@optlabels` Takes an optional argument which is a comma-separated list of label names, and invokes `\label` for each. We use this at the end of each of the new `\label`n macros so that each takes a list of label names as an optional argument.

```
45 \newcommand\subref@optlabels[1][\relax]{%
46   \ifx#1\relax\else
47     \@for\@subref@each:=#1\do{\expandafter\label\expandafter{\@subref@each}}%
48   \fi
49   \ignorespaces
50 }
```

Change History

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.