The apxproof package

Pierre Senellart
pierre@senellart.com

2016/10/27 v1.0.0-dev

Abstract

This package facilitates the writing of scientific article with proofs deferred to the appendix.

1 Usage

TODO

2 Implementation

We now describe the entire code of the package, in literate programming fashion.

2.1 Dependencies

We first load a few package dependencies:

- bibunits to add a second bibliography for the appendix material.

```
\RequirePackage{bibunits}
```

- environ to easily define the repeated theorem environments.

```
\RequirePackage{environ}
```

- etoolbox to define simple toggles.

```
\RequirePackage{etoolbox}
```

- fancyvrb for the bulk of the work of exporting appendix material in an auxiliary file.

```
\RequirePackage{fancyvrb}
```

- ifthen for easier comparison of character strings.

```
\RequirePackage{ifthen}
```

- kvoptions to manage options passed to the package.

```
\RequirePackage{kvoptions}
```

- amsthm for its `\newteorem` macro. Some document classes (e.g., lipics) preload the amsthm: this is fine, `\RequirePackage{amsthm}` will simply have no effect. On the other hand, some other document classes (e.g., llncs or sig-alternate) define a `proof` environment that conflicts with amsthm, so we have to undefine this environment before loading amsthm.

```
\@ifpackageloaded{amsthm}{
  }{
    \let\proof\undefined
    \let\endproof\undefined
  }
\RequirePackage{amsthm}
```

2.2 Option Processing

Many names througout the package use an arobase (`@`) to avoid name conflict with user-defined names. To simplify the compilation of the documentation, we simply make it a regular character in all the rest.

```
\makeatletter
```

We setup the processing of options using keyval facilities; the only declared options is named `appendix`, with a default value of `append`:

```
\SetupKeyvalOptions{
  family=AXP,
  prefix=AXP@
}
\DeclareStringOption[append]{appendix}
\ProcessLocalKeyvalOptions*
```

We check that the value of the `appendix` option is valid, and add a message to the compilation log.

```
\ifthenelse{\equal{\AXP@appendix}{append}}{
  \message{apxproof: Appendix material appended to the document}
}{\ifthenelse{\equal{\AXP@appendix}{strip}}{
  \message{apxproof: Appendix material stripped}
}{\ifthenelse{\equal{\AXP@appendix}{inline}}{
  \message{apxproof: Appendix material inlined within the document}
}{
  \errmessage{Error: unsupported option appendix=\AXP@appendix for
  package apxproof}
}}}
```

2.3 Macros Common to All Compilation Modes

`\newtheoremrep` We define the high-level `\newtheoremrep` to have the same syntax as amsthm's `\newtheorem`. For this purpose, we need a little trick to deal with the second

optional argument, which is what `\@oparg` is used for. `\axp@newtheoremrep` is defined differently depending on the compilation mode

```
\newcommand\newtheoremrep[1]{%
  \@oparg{\axp@newtheoremrep{#1}}[]%
}
```

`proofsketch` Simple `proofsketch` environment.

```
\newenvironment{proofsketch}
  {\vskip3pt\noindent\emph{Proof sketch.} }
  {\hfill\qed\vskip3pt}
```

`\thmhead` We redefine AMS-LaTeX's `\thmhead` to use a format where the repeated version of a theorem, using a theorem note, can look exactly like the original version of the theorem and its theorem counter.

```
\def\thmhead#1#2#3{%
  \thmname{#1}\thmnumber{\@ifnotempty{#1}{ }\@upn{#2}}%
  \thmnote{ #3}}
```

2.4 Inline Compilation Mode

```
\ifthenelse{\equal{\AXP@appendix}{inline}}{
```

`axp@newtheoremrep` In the inline mode, `axp@newtheoremrep` undefines the existing theorem environment if it has already been defined (e.g., by the document class), invokes `\newtheorem` and creates a repeated theorem environment that behaves exactly as the regular theorem environment.

```
  \def\axp@newtheoremrep#1[#2]#3{%
    \expandafter\let\csname #1\endcsname\undefined
    \expandafter\let\csname c@#1\endcsname\undefined
    \newtheorem{#1}[#2]{#3}%
    \NewEnviron{#1rep}[1][]{%
      \begin{#1}[##1]\BODY\end{#1}%
    }
  }
```

`inlineproof` `nestedproof` `appendixproof` In the inline mode, these environments behave like the regular `proof` environment.

```
  \let\inlineproof\proof
  \let\endinlineproof\endproof
  \let\nestedproof\proof
  \let\endnestedproof\endproof
  \let\appendixproof\proof
  \let\endappendixproof\endproof
```

`toappendix` `\noproofinappendix` `\nosectionappendix` In the inline mode, this environment and these macros are no-op.

```
  \newenvironment{toappendix}{}{}
  \let\noproofinappendix\relax
  \let\nosectionappendix\relax
```

Finally, some class-specific behavior so that theorems created by `\newtheoremrep` appear with the correct style. For now, only the styling for ACM document class (e.g., sig-alternate) needs to be adapted.

```
\ifdefined\@acmtitlebox
  \newtheoremstyle{mystyle}
    {3pt}
    {3pt}
    {\itshape}
    {}
    {\scshape}
    {.}
    {.5em}
    {}
  \theoremstyle{mystyle}
\fi
}
```

2.5 Append or Strip Compilation Modes

```
{
```

We now deal with the case where apxproof really does something useful: either append the appendix material to the document, or strip it entirely.

```
\newwrite\axp@proofsfile
\immediate\openout\axp@proofsfile=\jobname.axp
\immediate\write\axp@proofsfile{%
  \noexpand\makeatletter
  \noexpand\let\noexpand\proof\noexpand\axpold@proof
  \noexpand\let\noexpand\endproof\noexpand\endaxpold@proof
  \noexpand\let\noexpand\section\noexpand\axpold@section
}
\def\FVB@VerbatimOut{%
  \@bsphack
  \begingroup
    \FV@UseKeyValues
    \FV@DefineWhiteSpace
    \def\FV@Space{\space}%
    \FV@DefineTabOut
    \def\FV@ProcessLine{\immediate\write\axp@proofsfile}%
    \let\FV@FontScanPrep\relax
    \let\@noligs\relax
    \FV@Scan}
\def\FVE@VerbatimOut{\endgroup\@esphack}

\let\axpold@proof\proof
\let\endaxpold@proof\endproof

\newtoggle{axp@seenreptheorem}

```

```
\def\noproofinappendix{%
  \global\togglefalse{axp@seenreptheorem}%
}

\newenvironment{appendixproof}
  {%
    \axp@writesection
    \immediate\write\axp@proofsfile{%
      \noexpand\begin{axpold@proof}%
    }%
    \VerbatimOut
  }
  {%
    \endVerbatimOut
    \immediate\write\axp@proofsfile{%
      \noexpand\end{axpold@proof}%
    }%
    \noproofinappendix
  }

\renewenvironment{proof}
  {%
    \iftoggle{axp@seenreptheorem}{%
      \appendixproof
    }{%
      \axpold@proof
    }%
  }
  {%
    \iftoggle{axp@seenreptheorem}{%
      \endappendixproof
    }{%
      \endaxpold@proof
    }%
  }

\let\inlineproof\axpold@proof
\let\endinlineproof\endaxpold@proof

\let\nestedproof\axpold@proof
\let\endnestedproof\endaxpold@proof

\newcommand{\appendixrefname}{References for the Appendix}
\newcommand{\appendixbibliographystyle}{alpha}
\newcommand{\appendixbibliographyprelim}{}

\ifthenelse{\equal{\AXP@appendix}{append}}{
  \AtEndDocument{%
    \clearpage\onecolumn\appendix
    \appendixbibliographyprelim
```

```
    \begin{bibunit}[\appendixbibliographystyle]%
    \immediate\closeout\axp@proofsfile\input{\jobname.axp}%
    \renewcommand{\refname}{\appendixrefname}%
    \putbib
    \end{bibunit}%
  }
}{}

\let\axpold@bibliography\bibliography
\renewcommand\bibliography[1]{%
  \defaultbibliography{#1}%
  \axpold@bibliography{#1}%
}

\newcounter{axp@rpcounter}
\newcounter{axp@seccounter}

\def\axp@newtheoremrep#1[#2]#3{%
  \newtheorem*{axp@#1rp}{#3}%
  \expandafter\let\csname #1\endcsname\undefined
  \expandafter\let\csname c@#1\endcsname\undefined
  \newtheorem{#1}[#2]{#3}%
  \NewEnviron{#1rep}[1][]{%
    \addtocounter{axp@rpcounter}{1}%
    \begin{#1}[##1]\label{axp@r\roman{axp@rpcounter}}\BODY\end{#1}%
    \global\toggletrue{axp@seenreptheorem}%
    \global\expandafter\let\csname rplet\roman{axp@rpcounter}\endcsname\BODY%
    \axp@writesection%
    \immediate\write\axp@proofsfile{%
      \noexpand\begin{axp@#1rp}[\noexpand\ref{axp@r\roman{axp@rpcounter}}\@ifnotempty{##1}{\
      \noexpand\let\noexpand\label\noexpand\@gobble%
      \expandafter\noexpand\csname rplet\roman{axp@rpcounter}\endcsname\noexpand\end{axp@#1r
      }
  }
}

\def\axp@sectitle{}

\let\axpold@section\section
\def\section{\@ifstar\@section\@@section}
\def\@section#1{%
  \global\edef\axp@sectitle{#1}%
  \axpold@section*{#1}%
  \addtocounter{axp@seccounter}{1}%
  \label{axp@s\roman{axp@seccounter}}%
}
\def\@@section#1{%
  \global\edef\axp@sectitle{#1}%
  \axpold@section{#1}%
  \addtocounter{axp@seccounter}{1}%
```

```
    \label{axp@s\roman{axp@seccounter}}%
  }
  \ifdefined\@acmtitlebox
  \fi

  \newcommand{\nosectionappendix}{
    \global\def\axp@sectitle{}%
  }

  \newcommand\axp@writesection{%
    \ifx\axp@sectitle\empty
    \else
      \immediate\write\axp@proofsfile{\noexpand\def\noexpand\tmp{\noexpand\ref{axp@s\roman{axp
      (\axp@sectitle)}}%
      \nosectionappendix
    \fi
  }

  \newenvironment{toappendix}
    {\axp@writesection\VerbatimOut}
    {\endVerbatimOut}
```

2.5.1 Class-Specific Behavior

We conclude with some class-specific behavior.

`\@getcl@ss` `\@getclass` `\@currentclass`

We first use a little trick to store the current document class in macro `\@currentclass`, from http://tex.stackexchange.com/a/43541.

```
  \def\@getcl@ss#1.cls#2\relax{\def\@currentclass{#1}}
  \def\@getclass{\expandafter\@getcl@ss\@filelist\relax}
  \@getclass
```

ACM Document Classes

```
  \ifdefined\@acmtitlebox
```

`\thebibliography` `\refname` `\appendixrefname`

The section title of the bibliography is in uppercase in these document classes. In addition, the `\thebibliography` macro hard-codes twice the section title, so we un-hardcode it so that it can be modified in the appendix.

```
    \patchcmd{\thebibliography}{References}{\protect\refname}{}{}
    \patchcmd{\thebibliography}{References}{\protect\refname}{}{}
    \newcommand{\refname}{REFERENCES}
    \renewcommand{\appendixrefname}{REFERENCES FOR THE APPENDIX}
```

`\section` `\@@section`

These document classes redefine `\section` in a weird way, adding the possibility of an optional argument. We redefine this in a sane way.

```
    \def\section{\@ifstar\@section{\@dblarg{\@@section}}}
    \def\@@section#1#2{%
      \global\edef\axp@sectitle{#2}%
      \axpold@section{#2}%
      \addtocounter{axp@seccounter}{1}%
```

```
231       \label{axp@s\roman{axp@seccounter}}%
232     }

233   \fi
```

lipcs

```
234    \ifthenelse{\equal{\@currentclass}{lipics}}{
```

\appendixbibliographyprelim

The default bibliography in the lipics document class formatting is not compatible with the alpha bibliography style. We fix this here.

```
235     \renewcommand{\appendixbibliographyprelim}{%
236       \global\let\oldbiblabel\@biblabel
237       \def\@biblabel{\hspace*{-2em}\small\oldbiblabel}%
238     }

239   }

240 }
```

Change History

v1.0.0-dev

General: Initial version 1

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.