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. Throughout the package, we use the `axp@` prefix to identify local macros and environment names, not meant to be used by the final user.

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 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 throughout 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
```

`\axp@appendix` 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}}
```

`\appendixrefname` `\appendixbibliographystyle` `\appendixbibliographyprelim` We provide sensible defaults for these three user-customizable macros. Even though they are only useful in append mode, we define them for all modes so that a `\renewcommand` works in all cases.

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

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

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

2.4 Inline Compilation Mode

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

`\axp@newtheoremrep` In 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 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 inline mode, this environment and these macros are no-ops.

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

```
}
```

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.

2.5.1 Auxiliary File for the Appendix

`\axp@proofsfile` We open a new auxiliary file, with extension `.axp`, where the appendix material will be dumped.

```
\newwrite\axp@proofsfile
\immediate\openout\axp@proofsfile=\jobname.axp
```

`proof` `\section` At the beginning of this file, we make `@` a regular character (since it will be used in several places for internal names) and reestablish the original definition of the `proof` environment and the `\section` macro.

```
\immediate\write\axp@proofsfile{%
  \noexpand\makeatletter
  \noexpand\let\noexpand\proof\noexpand\axp@oldproof
  \noexpand\let\noexpand\endproof\noexpand\endaxp@oldproof
  \noexpand\let\noexpand\section\noexpand\axp@oldsection
}
```

`\FVB@VerbatimOut` `\FVE@VerbatimOut` We modify the internal behavior of the fancyvrb package to write to the `\axp@proofsfile`, without closing it and re-opening it on every write.

```
\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}
```

`toappendix` The entire content of this environment in put in appendix, possibly after starting a new appendix section if needed.

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

2.5.2 Definition of New Theorems

`axp@seenreptheorem` Used to indicate whether a repeated theorem was just typeset, without its proof.

```
\newtoggle{axp@seenreptheorem}
```

`axp@rpcounter` Sequentially incremented for every repeated theorem, used to create labels.

```
\newcounter{axp@rpcounter}
```

`axp@newtheoremrep` When called with first argument `foobar`, we first undefine the existing `foobar` environment (and its counter) if it has already been defined (e.g., by the document class), then invoke `\newtheorem` for the regular version of the theorem `foobar` and `\newtheorem*` for an internal version `axp@foobarrp` that will be used in the appendix to restate the existing theorem.

```
\def\axp@newtheoremrep#1[#2]#3{%
  \expandafter\let\csname #1\endcsname\undefined
  \expandafter\let\csname c@#1\endcsname\undefined
  \newtheorem{#1}[#2]{#3}%
  \newtheorem*{axp@#1rp}{#3}%
```

We then define a `foobarrep` environment that increments the `axp@rpcounter` and typeset the regular `foobar` theorem with a label derived from the counter.

```
  \NewEnviron{#1rep}[1][]{%
    \addtocounter{axp@rpcounter}{1}%
    \begin{#1}[##1]\label{axp@r\roman{axp@rpcounter}}\BODY\end{#1}%
```

We set the `axp@seenreptheorem` toggle to indicate that we are looking for the proof of the theorem, then store in a macro the content of the theorem's body.

```
    \global\toggletrue{axp@seenreptheorem}%
```

```
      \global\expandafter\let\csname rplet\roman{axp@rpcounter}\endcsname
      \BODY
```

Possibly after starting a new appendix section if needed, we typeset a repeated version of the theorem using the `axp@foobarrp` environment and a reference to the previously defined label. We ignore any use of `\label` in this environment to avoid multiply defined labels.

```
      \axp@writesection%
      \immediate\write\axp@proofsfile{%
        \noexpand\begin{axp@#1rp}
          [\noexpand\ref{axp@r\roman{axp@rpcounter}}\@ifnotempty{##1}{\noexpand##1}]%
          \noexpand\let\noexpand\label\noexpand\@gobble%
          \expandafter\noexpand\csname rplet\roman{axp@rpcounter}\endcsname
        \noexpand\end{axp@#1rp}
      }
    }
  }
```

2.5.3 Proof Environments

`axp@oldproof` We save the definition of the existing `proof` environment.

```
  \let\axp@oldproof\proof
  \let\endaxp@oldproof\endproof
```

`\noproofinappendix` Utility macro that toggles `axp@seenreptheorem` to false.

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

`appendixproof` We dump the content of this in appendix, within an original `proof` environment, possibly after creating a new appendix section.

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

`proof` This environment either puts the proof in appendix, if we are after a repeated theorem without its proof, or inlines it otherwise.

```
\renewenvironment{proof}
  {%
    \iftoggle{axp@seenreptheorem}{%
      \appendixproof
    }{%
      \axp@oldproof
    }%
  }
  {%
    \iftoggle{axp@seenreptheorem}{%
      \endappendixproof
    }{%
      \endaxp@oldproof
    }%
  }
```

`inlineproof` `nestedproof` These two environments are synonyms for the original `proof` environment.

```
\let\inlineproof\axp@oldproof
\let\endinlineproof\endaxp@oldproof
\let\nestedproof\axp@oldproof
\let\endnestedproof\endaxp@oldproof
```

2.5.4 Section Management

`axp@seccounter` Sequentially incremented for every section, used to create labels.

```
\newcounter{axp@seccounter}
```

`\axp@sectitle` Saves the title of the last encountered section.

```
\def\axp@sectitle{}
```

`\axp@oldsection` `\section` `\@section` `\@@section` We redefine the `\section` command to create a label based on `axp@seccounter` and to store its title in `\axp@sectitle`. Two definitions are necessary to cover the starred and unstarred use of `\section`, though most likely the former is not going to be used (since no section number will appear to refer to that section in the appendix).

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

\nosectionappendix We remove the current section title, to indicate no section should be created in the appendix.

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

\axp@writesection If `\axp@sectitle` is not empty, we create a new section in the appendix, referring to the main text section.

```
\newcommand\axp@writesection{%
  \ifx\axp@sectitle\@empty
  \else
    \immediate\write\axp@proofsfile{%
      \noexpand\def\noexpand\axp@tmp{\noexpand\ref{axp@s\roman{axp@seccounter}}}
      \noexpand\axp@oldsection{%
        Proofs for Section\noexpand~\noexpand\protect\noexpand\axp@tmp{}
        (\axp@sectitle)%
      }%
    }%
    \nosectionappendix
  \fi
}
```

2.5.5 Append Compilation Mode

```
\ifthenelse{\equal{\axp@appendix}{append}}{
```

\axp@oldbibliography \bibliography Thanks to bibunits's `\defaultbibliography` macro, we set the appendix bibliography source to be the same as that of the main text.

```
  \let\axp@oldbibliography\bibliography
  \renewcommand\bibliography[1]{%
    \defaultbibliography{#1}%
    \axp@oldbibliography{#1}%
  }
```

After the end of the main text, we add the appendix (on a new page, set in single-column mode) within a `bibunit` environment so as to typeset a separate bibliography for the appendix.

```
  \AtEndDocument{
    \clearpage\onecolumn\appendix
    \begin{bibunit}[\appendixbibliographystyle]
      \immediate\closeout\axp@proofsfile
      \input{\jobname.axp}
      \renewcommand{\refname}{\appendixrefname}
      \appendixbibliographyprelim
      \putbib
    \end{bibunit}
  }
}{}
```

2.5.6 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 them in a sane way.

```
  \def\section{\@ifstar\@section{\@dblarg{\@@section}}}
  \def\@@section[#1]#2{%
    \global\edef\axp@sectitle{#2}%
    \axp@oldsection{#2}%
    \addtocounter{axp@seccounter}{1}%
    \label{axp@s\roman{axp@seccounter}}%
  }

\fi
```

lipcs

```
 \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.

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

}
}
```

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.