The apxproof package

Pierre Senellart
pierre@senellart.com
http://github.com/PierreSenellart/apxproof

2016/10/29 v1.0.0-dev

Abstract

This package makes it easier to write articles where proofs and other material are deferred to the appendix. The appendix material is written in the LaTeX code along with the main text which it naturally complements, and it is automatically deferred. The package can automatically send proofs in the appendix, can repeat in the appendix the theorem environments stated in the main text, can section the appendix automatically based on the sectioning of the main text, and supports a separate bibliography for the appendix material.

1 Usage

The apxproof package is intended to simplify the writing of articles where some of the content needs to be deferred to an appendix. This is in particular useful for the submission of scientific articles to conferences or journals that limit the number of pages in the main text but allow an extra appendix, where proofs of theorems and other material can be added.

1.1 Basics

To use apxproof, first load it in the header of your document:

```
\usepackage{apxproof}
```

On its own, this does not do anything and should not change the appearance of your document. To add an appendix with some material from your document, use the `toappendix` environment:

toappendix

```
\begin{toappendix}
  ...
\end{toappendix}
```

The content will appear at the end of your document, in an automatically generated section that refers to the current section in the main text.

Example 1. *Throughout this documentation, all examples produce content deferred to the appendix, at the very end of this document.*

```
\begin{toappendix}
This content is in the appendix.
\end{toappendix}
```

1.2 Repeated Theorems and Proofs

In some scientific papers that include proofs, it is common to defer proofs to the appendix. This can easily be achieved using the `appendixproof` environment:

appendixproof

```
\begin{appendixproof}
  ...
\end{appendixproof}
```

This behaves like the `toappendix` environment, except that a proof environment is generated.

Example 2. *We now send a proof to the appendix:*

```
\begin{appendixproof}
This proof is in the appendix.
\end{appendixproof}
```

When deferring proofs to the appendix, an annoying problem is that the statement of the theorem remains in the main text; it is hard to read a proof that is far away from the statement it proves. apxproof solves this issue by allowing statements of theorems to be *repeated*: once in the main text, and once in the appendix before the proof of the statement. To use this feature, you can define a new *repeated theorem* environment using the `\newtheoremrep` command:

\newtheoremrep

```
\newtheoremrep{⟨name⟩}[⟨counter⟩]{⟨title⟩}
```

Usage is exactly the same as that of AMS LaTeX's `\newtheorem` macro: ⟨*name*⟩ (e.g., `theorem`) is the name of an environment that is created for this kind of theorem, ⟨*counter*⟩ (e.g., `definition`) is an optional counter describing from which kind of enviroment should the numbering of these environments be inherited, and ⟨*title*⟩ (e.g., `Theorem` is the title that will be used to display this theorem environment. What differs from `\newtheorem` is that, when the following is written:

```
\newtheoremrep{foobar}{Foobar}
```

then *two* environments are defined: the `foobar` environment, which behaves as if `\newtheorem` had been used, and the `foobarrep` environment, which results in the statement of this environment being repeated in the appendix.

proof

One interesting feature of apxproof is that in most situations, there is no need to use the `appendixproof` environment. Indeed, the `proof` environment is redefined by apxproof to automatically put the proof either in the main text (if it follows a regular theorem) or in the appendix (if it follows a repeated theorem).

Example 3. *Assume we have first defined a repeated theorem environment* `foobar` *as above. We can now use this theorem environment, first for a regular theorem in the main text, then for a theorem repeated in the main text and in the appendix:*

```
\begin{foobar}
This foobar is a regular one, in the main text.
\end{foobar}
\begin{proof}
This is the proof of the regular foobar.
\end{proof}
```

We obtain:

Foobar 1. *This foobar is a regular one, in the main text.*

Proof. This is the proof of the regular foobar. □

```
\begin{foobarrep}
This foobar is repeated in the appendix.
\end{foobarrep}
\begin{proof}
This is the proof of the repeated foobar.
\end{proof}
```

We now obtain:

Foobar 2. *This foobar is repeated in the appendix.*

Note that, since `hyperref` *is loaded, there is a hyperlink created from the statement of the theorem in the appendix to that in the main text.*

When the proof is deferred to the appendix, it is common practice to add a proof sketch in the main text. apxproof defines a simple `proofsketch` environment for this purpose:

`proofsketch`

```
\begin{proofsketch}
  ...
\end{proofsketch}
```

The proof sketch is typeset similarly to a proof, but is always in the main text. `inlineproof` Similarly, an `inlineproof` environment is provided so as to be able to have both a proof in the appendix (using the regualr `proof` environment, or alternatively the `appendixproof` environment) and a different proof in the main text (using the `inlineproof` environment).

Example 4. *Here are simple examples of proof sketches and inline proofs:*

```
\begin{proofsketch}
This is a proof sketch.
\end{proofsketch}
```

Proof sketch. This is a proof sketch. □

```
\begin{inlineproof}
This is an inline proof.
\end{inlineproof}
```

Proof. This is an inline proof. □

1.3 Bibliography

apxproof automatically adds a bibliography in the appendix with only the references cited in the appendix material. This allows for a clean separation of references used solely in the main text, and those used in the appendix.

Example 5. *Assume we have citations both in the main text and in the appendix.*

```
This is a citation in the main text~\cite{lamport86}.
\begin{toappendix}
This is a citation in the appendix~\cite{proofsAreHard}.
\end{toappendix}
```

This is a citation in the main text [1].

The bibliography in the appendix can use a different style and heading than the bibliography in the main text (and, by default, it does). See Section 1.5 for how to configure the appearance of that bibliography.

1.4 Mode

An optional ⟨*mode*⟩ can be specified when loading the package:

```
\usepackage[appendix=⟨mode⟩]{apxproof}
```

⟨*mode*⟩ can take one of the following three values:

`append` This is the default. Appendix material gathered by apxproof is appended to the main text.

`inline` In this mode, apxproof simply inline the content along with the main text.

`strip` This mode functions similarly to `append` except that the appendix is not appended at the end of the document. All appendix material is therefore removed.

1.5 Customization

apxproof provides a few macros that can be redefined (using `\renewcommand`) to customize the appearance of the appendix:

`\appendixsectionformat`

`\appendixsectionformat{`⟨***number***⟩`}{`⟨***title***⟩`}` is a macro that indicates how to format the section titles in the Appendix, given the number and title of the section in the main text. By default, they appear as “Proofs for Section ⟨*number*⟩ (⟨*title*⟩)”.

`\appendixrefname`

`\appendixrefname` contains the heading that is displayed before the bibliography. By default, this is “References for the Appendix”.

`\appendixbibliographystyle`

`\appendixbibliographystyle` contains the `.bst` bibliography style that is used in the bibliography in appendix. By default, this is `alpha`.

`\appendixbibliographyprelim`

`\appendixbibliographyprelim` contains arbitrary code that is executed just before the production of the bibliography in appendix, which can be used to configure the way it is displayed.

1.6 Advanced Commands

We now describe a few advanced macros and environments, whose usage is limited to special cases:

`nestedproof`

`nestedproof` is an environment that can be used within a `proof` environment deferred in the appendix; this is required because, for technical reasons, no `proof` environment can be nested within a deferred `proof` environment.

`\noproofinappendix`

`\noproofinappendix` can be used inside repeated theorems that are not followed by a `proof` or `appendixproof` theorem; the point is to ensure that a further `proof` environment cannot be mistakenly understood as a proof of the repeated theorem. It should not be needed in most situations as apxproof tries figuring out when a proof follows a repeated theorem automatically, but may occasionally be needed in complex scenarios.

`\nosectionappendix`

`\nosectionappendix` is to be used inside a section that *does* contain appendix material, but for which a section in the appendix should not be created. This should be rarely needed. When this command is present, appendix material is appended to the end of the previously created section.

2 Supported Document Classes

Because apxproof modifies sectioning commands, bibliographies, and proofs, it may not work straight away with arbitrary document classes. It has currently been tested with and is supported for the following document classes:

- LaTeX standard document classes (e.g., `article.cls`)

- KOMA-Script (e.g., `scrartcl.cls`)
- ACM SIG Proceedings (e.g., `sig-alternate.cls`)
- Springer's Lecture Notes in Computer Science (e.g., `llncs.cls`)
- Schloß Dagstuhl's Leibniz International Proceedings in Informatics (e.g., `lipics.cls`)

Other classes may work out of the box. Adding support for specific classes is possible and can be requested from the author of this package.

3 Known Issues and Limitations

We report here some issues we are currently aware of:

- When using hyperref, the appendix in the bibliography is not hyperlinked. This is to avoid possible issues with multiply defined bibliography entries.
- `appendixproof`, `proof`, `toappendix` environments cannot be nested. This is a limitation of the fancyvrb package that apxproof relies on. Note the existence of the `nestedproof` environment for nested proofs.
- apxproof is incompatible with a separate use of the fancyvrb package. This is because apxproof redefines some internal mechanisms of fancyvrb.

Issues not listed here should be reported to the author.

4 License

5 Contact

- `https://github.com/PierreSenellart/apxproof`
- Pierre Senellart <pierre@senellart.com>

Bug reports and feature requests should preferably be submitted through the *Issues* feature of GitHub.

6 Acknowledgments

Thanks to Antoine Amarilli for feedback and proofreading.

7 Implementation

We now describe the entire code of the package, in a literate programming fashion. Throughout the package, we use the `axp@` prefix to identify local macros and environment names, which are not meant to be used by the final user.

7.1 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}
```

7.2 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 option 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}
}}}
```

7.3 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\textit{Proof sketch.} \upshape}
  {\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
\appendixsectionformat

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}{}
\newcommand{\appendixsectionformat}[2]{Proofs for Section~#1 (#2)}
```

Finally, some class-specific behavior ensures that the 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
```

7.4 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
```

```
}
```

7.5 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.

7.5.1 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 is put in appendix, starting a new appendix section beforehand if needed.

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

7.5.2 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` (patching the `\begin{foobar}` so as not to expect a proof in the appendix) 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}%
  \expandafter\pretocmd\csname #1\endcsname{\noproofinappendix}{}{}%
  \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}
    }
  }
}
```

7.5.3 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
```

7.5.4 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@section
\axp@@ssection
\axp@@section

This command behaves similarly to `\axp@oldsection`, except that it first tests whether a `\section` follows, and if so, does not produce anything. This is useful to avoid producing empty sections in the appendix. As usual, we have to process starred and unstarred version separately.

```
\def\axp@section{\@ifstar\axp@@ssection\axp@@section}
\def\axp@@ssection#1{%
  \@ifnextchar\section{}{\axp@oldsection*{#1}}%
}
\def\axp@@section#1{%
  \@ifnextchar\section{}{\axp@oldsection{#1}}%
}
```

\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@section{%
        \noexpand\appendixsectionformat{\protect\noexpand\axp@tmp}%
                                       {\axp@sectitle}%
      }%
    }%
    \nosectionappendix
  \fi
}
```

7.5.5 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}
 }
}{}
```

7.5.6 Class-Specific Behavior

We conclude with some class-specific behavior.

\@getcl@ss
\@getclass
\@currentclass

We first use a little trick to store the 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}%
 }
```

```
243   }
244 }
```

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.

References

[1] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison–Wesley Pub. Co., Reading, MA, 1986.

A Proofs for Section 1 (Usage)

This content is in the appendix.

Proof. This proof is in the appendix. □

Foobar 2. *This foobar is repeated in the appendix.*

Proof. This is the proof of the repeated foobar. □

This is a citation in the appendix [Unk16].

References for the Appendix

[Unk16] Unknown. Proofs are hard, 2016.