The `apxproof` package

Pierre Senellart

`pierre@senellart.com`

http://github.com/PierreSenellart/apxproof

2017/05/31   v1.0.5-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 to 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}
```

When the content to put in appendix is an entire section, make sure that `\section` is the very first command that appears within the `toappendix` environment. It will disable the automatic production of a section heading.

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
$$\newtheoremrep\{\langle name\rangle\}[\langle counter\rangle]\{\langle title\rangle\}[\langle countersec\rangle]$$

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 environment should the numbering of these environments be inherited;

- ⟨*title*⟩ (e.g., `Theorem`) is the title that will be used to display this theorem environment;

- ⟨*countersec*⟩ (e.g., `section`) is an optional counter of a sectioning command indicating that counters for this theorem should be prefixed by this counter (and reset at each occurrence of the sectioning command).

⟨*counter*⟩ and ⟨*countersec*⟩ should not be used together. 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

By default, 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.
option   In order to use a single appendix for the main text and the bibliography, bibliography   one can specify the value `common` to the `bibliography` option when loading the package. (By default this option is set to `separate`.)

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.

`\appendixprelim` `\appendixprelim` contains arbitrary code that is executed just before the production of the appendix, which can be used to configure the way it is displayed. By default, this command contains `\clearpage\onecolumn` (the appendix is typeset on a new page in single-column mode) but redefining this option allows changing this behavior.

1.6 Advanced Features

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.

```
1    \RequirePackage{bibunits}
```

- environ to easily define the repeated theorem environments.

```
2    \RequirePackage{environ}
```

- etoolbox to define simple toggles.

```
3    \RequirePackage{etoolbox}
```

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

```
4    \RequirePackage{fancyvrb}
```

- ifthen for easier comparison of character strings.

```
5     \RequirePackage{ifthen}
```

- kvoptions to manage options passed to the package.

```
6     \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. In that case, we reestablish the existing proof environments, in case they had been customized (e.g., sig-alternate)

```
7     \@ifpackageloaded{amsthm}{
8       }{
9         \let\apx@oldamsthmproof\proof
10        \let\apx@oldamsthmendproof\endproof
11        \let\proof\undefined
12        \let\endproof\undefined
13      }
14    \RequirePackage{amsthm}
15    \ifdefined\apx@oldamsthmproof
16      \let\proof\apx@oldamsthmproof
17      \let\endproof\apx@oldamsthmendproof
18    \fi
```

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.

```
19 \makeatletter
```

We setup the processing of options using keyval facilities.

```
20 \SetupKeyvalOptions{
21   family=axp,
22   prefix=axp@
23 }
```

We declare the following options:

- appendix, with a default value of append (other possible values: strip, inline);

- bibliography, with a default value of separate (other possible value: common);

\axp@appendix

```
24 \DeclareStringOption[append]{appendix}
```

```
25 \DeclareStringOption[separate]{bibliography}
```

```
26 \ProcessLocalKeyvalOptions*
```

We check that the value of the `appendix` and `bibliography` options are valid, and add a message to the compilation log.

```
27 \ifthenelse{\equal{\axp@appendix}{append}}{
28   \message{apxproof: Appendix material appended to the document}
29 }{\ifthenelse{\equal{\axp@appendix}{strip}}{
30   \message{apxproof: Appendix material stripped}
31 }{\ifthenelse{\equal{\axp@appendix}{inline}}{
32   \message{apxproof: Appendix material inlined within the document}
33 }{
34   \errmessage{Error: unsupported option appendix=\axp@appendix for
35   package apxproof}
36 }}}
37 \ifthenelse{\equal{\axp@bibliography}{separate}}{
38   \message{apxproof: Separate bibliography for appendix material}
39 }{\ifthenelse{\equal{\axp@bibliography}{common}}{
40   \message{apxproof: Common bibliography for appendix and main text}
41 }{
42   \errmessage{Error: unsupported option bibliography=\axp@bibliography for
43   package apxproof}
44 }}
```

7.3 Macros Common to All Compilation Modes

`\axp@newtheorem`
`\@axp@newtheorem`
`\@@axp@newtheorem`

We introduce an intermediate `\axp@newtheorem` command to define a new theorem, differently depending on whether there is a section counter or not. This will be useful, in particular to allow changing this definition depending on the document class. This command uses two intermediary commands, `\@axp@newtheorem` and `\@@axp@newtheorem`, for the non-starred and starred versions.

`\newtheoremrep`
`\axp@newtheoremreptmp`

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 and fourth optional arguments, which is what `\@oparg` and `\axp@newtheoremreptmp` are used for. `\axp@newtheoremrep` is defined differently depending on the compilation mode.

```
45 \newcommand\newtheoremrep[1]{%
46   \@oparg{\axp@newtheoremreptmp{#1}}[]%
47 }
48 \def\axp@newtheoremreptmp#1[#2]#3{%
49   \@oparg{\axp@newtheoremrep{#1}[#2]{#3}}[]%
50 }
```

`proofsketch`

Simple `proofsketch` environment.

```
51   \newenvironment{proofsketch}{\begin{axp@oldproof}[Proof sketch]}{\end{axp@oldproof}}
```

<dl>
<dt>\thmhead</dt>
</dl>

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.

```
52 \AtBeginDocument{%
53   \def\thmhead#1#2#3{%
54     \thmname{#1}\thmnumber{\@ifnotempty{#1}{ }\@upn{#2}}%
55     \thmnote{ #3}}%
56 }
```

<dl>
<dt>\appendixrefname</dt>
<dt>\appendixbibliographystyle</dt>
<dt>\appendixbibliographyprelim</dt>
<dt>\appendixprelim</dt>
<dt>\appendixsectionformat</dt>
</dl>

We provide sensible defaults for these four 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.

```
57 \newcommand{\appendixrefname}{References for the Appendix}
58 \newcommand{\appendixbibliographystyle}{alpha}
59 \newcommand{\appendixbibliographyprelim}{}
60 \newcommand{\appendixprelim}{\clearpage\onecolumn}
61 \newcommand{\appendixsectionformat}[2]{Proofs for Section~#1 (#2)}
```

<dl>
<dt>axp@oldproof</dt>
</dl>

We save the definition of the existing `proof` environment.

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

7.4 Inline Compilation Mode

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

<dl>
<dt>\axp@newtheoremrep</dt>
</dl>

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.

```
65   \def\axp@newtheoremrep#1[#2]#3[#4]{%
66     \expandafter\let\csname #1\endcsname\undefined
67     \expandafter\let\csname c@#1\endcsname\undefined
68     \axp@newtheorem{#1}{#2}{#3}{#4}%
69     \NewEnviron{#1rep}[1][]{%
70       \begin{#1}[##1]\BODY\end{#1}}%
71     }
72   }
```

<dl>
<dt>inlineproof</dt>
<dt>nestedproof</dt>
<dt>appendixproof</dt>
</dl>

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

```
73   \let\inlineproof\proof
74   \let\endinlineproof\endproof
75   \let\nestedproof\proof
76   \let\endnestedproof\endproof
77   \let\appendixproof\proof
78   \let\endappendixproof\endproof
```

<dl>
<dt>toappendix</dt>
<dt>\noproofinappendix</dt>
<dt>\nosectionappendix</dt>
</dl>

In inline mode, this environment and these macros are no-ops.

```
79    \newenvironment{toappendix}{}{}
80    \let\noproofinappendix\relax
81    \let\nosectionappendix\relax

82 }
```

7.5   Append or Strip Compilation Modes

```
83 {
```

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.

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

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

```
86    \immediate\write\axp@proofsfile{%
87      \noexpand\makeatletter
88      \noexpand\let\noexpand\proof\noexpand\axp@oldproof
89      \noexpand\let\noexpand\endproof\noexpand\endaxp@oldproof
90      \noexpand\let\noexpand\section\noexpand\axp@oldsection
91    }
```

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

```
92    \def\FVB@VerbatimOut{%
93      \@bsphack
94      \begingroup
95        \FV@UseKeyValues
96        \FV@DefineWhiteSpace
97        \def\FV@Space{\space}%
98        \FV@DefineTabOut
99        \def\FV@ProcessLine{\immediate\write\axp@proofsfile}%
100       \let\FV@FontScanPrep\relax
101       \let\@noligs\relax
102       \FV@Scan}
103   \def\FVE@VerbatimOut{\endgroup\@esphack}
```

toappendix   The entire content of this environment is put in appendix, starting a new appendix section beforehand if needed.

```
104   \newenvironment{toappendix}
105     {\axp@writesection\VerbatimOut}
106     {\endVerbatimOut}
```

11

7.5.2 Definition of New Theorems

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

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

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

```
108    \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.

```
109    \def\axp@newtheoremrep#1[#2]#3[#4]{%
110      \expandafter\let\csname #1\endcsname\undefined
111      \expandafter\let\csname c@#1\endcsname\undefined
112      \axp@newtheorem{#1}{#2}{#3}{#4}%
113      \expandafter\pretocmd\csname #1\endcsname{\noproofinappendix}{}{}%
114      \axp@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. We distinguish the case when the theorem argument has a note and when it does not.

```
115    \NewEnviron{#1rep}[1][]{%
116      \addtocounter{axp@rpcounter}{1}%
117      \ifx\relax##1\relax
118        \begin{#1}\label{axp@r\roman{axp@rpcounter}}\BODY\end{#1}%
119      \else
120        \begin{#1}[##1]\label{axp@r\roman{axp@rpcounter}}\BODY\end{#1}%
121      \fi
```

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.

```
122      \global\toggletrue{axp@seenreptheorem}%
123      \global\expandafter\let\csname rplet\roman{axp@rpcounter}%
124                       \endcsname
125      \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.

```
126      \axp@writesection%
127      \immediate\write\axp@proofsfile{%
128        \noexpand\begin{axp@#1rp}
129          [\noexpand\ref{axp@r\roman{axp@rpcounter}}}%
130          \@ifnotempty{##1}{ \unexpanded{##1}}]%
131        \noexpand\let\noexpand\label\noexpand\@gobble%
132        \expandafter\noexpand\csname rplet\roman{axp@rpcounter}%
```

12

```
133                                      \endcsname
134          \noexpand\end{axp@#1rp}
135        }
136      }
137    }
```

7.5.3   Proof Environments

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

```
138    \newcommand\noproofinappendix{%
139      \global\togglefalse{axp@seenreptheorem}%
140    }
```

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

```
141    \newenvironment{appendixproof}
142      {%
143        \axp@writesection
144        \immediate\write\axp@proofsfile{%
145          \noexpand\begin{axp@oldproof}%
146        }%
147        \VerbatimOut
148      }
149      {%
150        \endVerbatimOut
151        \immediate\write\axp@proofsfile{%
152          \noexpand\end{axp@oldproof}%
153        }%
154        \noproofinappendix
155      }
```

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

```
156    \renewenvironment{proof}
157      {%
158        \iftoggle{axp@seenreptheorem}{%
159          \appendixproof
160        }{%
161          \axp@oldproof
162        }%
163      }
164      {%
165        \iftoggle{axp@seenreptheorem}{%
166          \endappendixproof
167        }{%
168          \endaxp@oldproof
169        }%
170      }
```

13

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

```
171    \let\inlineproof\axp@oldproof
172    \let\endinlineproof\endaxp@oldproof
173    \let\nestedproof\axp@oldproof
174    \let\endnestedproof\endaxp@oldproof
```

7.5.4 Section Management

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

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

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

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

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

```
177    \def\axp@section{\@ifstar\axp@@ssection\axp@@section}
178    \def\axp@@ssection#1{%
179      \@ifnextchar\section{}{\axp@oldsection*{#1}}%
180    }
181    \def\axp@@section#1{%
182      \@ifnextchar\section{}{\axp@oldsection{#1}}%
183    }
```

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

```
184    \let\axp@oldsection\section
185    \def\section{\@ifstar\@section\@@section}
186    \def\@section#1{%
187      \global\edef\axp@sectitle{#1}%
188      \axp@oldsection*{#1}%
189      \addtocounter{axp@seccounter}{1}%
190      \label{axp@s\roman{axp@seccounter}}%
191    }
192    \def\@@section#1{%
193      \global\edef\axp@sectitle{#1}%
194      \axp@oldsection{#1}%
195      \addtocounter{axp@seccounter}{1}%
196      \label{axp@s\roman{axp@seccounter}}%
197    }
```

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

```
198    \newcommand{\nosectionappendix}{
199      \global\def\axp@sectitle{}%
200    }
```

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

```
201    \newcommand\axp@writesection{%
202      \ifx\axp@sectitle\@empty
203      \else
204        \immediate\write\axp@proofsfile{%
205          \noexpand\def\noexpand\axp@tmp{%
206            \noexpand\ref{axp@s\roman{axp@seccounter}}%
207          }
208          \noexpand\axp@section{%
209            \noexpand\appendixsectionformat{\protect\noexpand\axp@tmp}%
210                                          {\axp@sectitle}%
211          }%
212        }%
213        \nosectionappendix
214      \fi
215    }
```

7.5.5   Append Compilation Mode

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

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

```
217      \let\axp@oldbibliography\bibliography
218      \renewcommand\bibliography[1]{%
219        \defaultbibliography{#1}%
220        \axp@oldbibliography{#1}%
221      }
```

After the end of the main text, we add the appendix (after the command
\appendixprelim is issued) within a bibunit environment so as to typeset a
separate bibliography for the appendix (unless the bibliography option is set to
common). There is an extra test to ensure an empty bibliography environment is
not produced.

```
222      \AtEndDocument{
223        \appendixprelim
224        \appendix
225        \ifthenelse{\equal{\axp@bibliography}{separate}}{
226        \begin{bibunit}[\appendixbibliographystyle]
227        }{}
228          \immediate\closeout\axp@proofsfile
229          \input{\jobname.axp}
230        \ifthenelse{\equal{\axp@bibliography}{separate}}{
231          \renewcommand{\refname}{\appendixrefname}
```

```
232        \let\axp@oldthebibliography\thebibliography
233        \renewcommand\thebibliography[1]{%
234          \ifx\relax#1\relax\else\axp@oldthebibliography{#1}\fi}
235        \appendixbibliographyprelim
236        \putbib
237      \end{bibunit}
238    }{}
239  }
240 }{}
```

7.5.6 Class-Specific Behavior

We conclude with some class-specific behavior.

ACM Document Classes

```
241    \ifdefined\@acmtitlebox
```

We first redefine the `proofsketch` environment, which is used differently in the base class.

```
242    \renewenvironment{proofsketch}{\begin{axp@oldproof}[sketch]}{\end{axp@oldproof}}
```

We adjust the styling of theorems for the needs of apxproof.

```
243    \newtheoremstyle{mystyle}
244      {6pt}
245      {6pt}
246      {\itshape}
247      {10pt}
248      {\scshape}
249      {.}
250      {.5em}
251      {}
252    \theoremstyle{mystyle}
```

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

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

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

```
257    \def\section{\@ifstar\@section{\@dblarg{\@@section}}}
258    \def\@@section[#1]#2{%
259      \global\edef\axp@sectitle{#2}%
260      \axp@oldsection{#2}%
261      \addtocounter{axp@seccounter}{1}%
262      \label{axp@s\roman{axp@seccounter}}%
263    }
```

16

```
264    \fi
```

lipcs
```
265    \ifdefined\lipics@opterrshort
```

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

```
266    \renewcommand{\appendixbibliographyprelim}{%
267      \global\let\@oldbiblabel\@biblabel
268      \def\@biblabel{\hspace*{-2em}\small\@oldbiblabel}%
269    }
```

```
270    \fi
```

llncs and other Springer document classes
```
271    \@ifclassloaded{llncs}{
```

proofsketch We first redefine the proofsketch environment, which is used differently in the base class.

```
272      \renewenvironment{proofsketch}{\begin{axp@oldproof}[sketch]}{\end{axp@oldproof}}
```

```
273    }{}
274      \ifdefined\spnewtheorem
```

\@axp@newtheorem It is necessary to use \spnewtheorem instead of \newtheorem in Springer document
\@@axp@newtheorem classes to obtain standard formatting.

```
275  \def\@axp@newtheorem#1#2#3#4{%
276    \ifx\relax#4\relax
277      \spnewtheorem{#1}{#3}{\bfseries}{\itshape}%
278    \else
279      \spnewtheorem{#1}{#3}[#4]{\bfseries}{\itshape}%
280    \fi
281  }
282  \def\@@axp@newtheorem#1#2{%
283    \spnewtheorem*{#1}{#2}{\upshape\bfseries}{\itshape}%
284  }
```

We have to redefine the macro \@thmcountersep for proper sectioned counters.
```
285  \def\@thmcountersep{.}
```

We remove the parentheses added by default for theorem notes, which are not compatible with the use of theorem notes by apxproof.

```
286  \def\@Opargbegintheorem#1#2#3#4{#4\trivlist
287      \item[\hskip\labelsep{#3#1}]{#3 #2\@thmcounterend\ }}
288  \def\@spopargbegintheorem#1#2#3#4#5{\trivlist
289      \item[\hskip\labelsep{#4#1\ #2}]{#4 #3\@thmcounterend\ }#5}
```

```
290    \fi
```

```
291  }
```

17

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.

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.