The phfthm package

Philippe Faist  philippe.faist@bluewin.ch

March 01, 2021

1 This document corresponds to phfthm v1.2, dated 2021/03/01. It is part of the phfqitltx package suite, see https://github.com/phfaist/phfqitltx.

phfthm—Goodies for theorems and proofs.

1 Introduction

1.1 Theorem environments

1.2 Proof environments

1.3 Theorem-heading environments

2 Quick start and package options

2.1 Predefined theorem environments

2.2 The proof environment

2.3 The theorem-heading environment

3 Theorem environments

3.1 Define theorem environments manually

3.2 Loading theorem sets manually

3.3 Theorem hooks

4 Proof environments

4.1 Manually define a proof environment

4.2 Proof hooks

5 Pairing theorems to proofs and proof-reference mechanism

5.1 On the theorem side

5.2 On the proof side

5.3 Customizing appearance of the proof reference text

6 Theorem-heading definition-like environments

6.1 Define theorem-heading environments manually

6.2 Available hooks for theorem-heading environments

7 Implementation

7.1 Generic Internal Stuff

7.2 Definitions for theorem environments

7.2.1 \phfMakeTheorem: definition of a new theorem environment

7.2.2 Default hooks for theorems

7.2.3 Proof-ref mechanism (on the theorem side)

7.3 Definitions for proof environments

7.3.1 Define a proof environment: \phfMakeProofEnv

7.3.2 Common hooks for proofs
1 Introduction

The \texttt{phfthm} package provides enhanced theorem and proof environments, based on the \texttt{amsthm} original versions. It allows for hooks to be placed, adds some default goodies and is highly customizable.

There are three generic types of environments provided: theorem environments, proof environments and “thmheading” environments.

1.1 Theorem environments

Theorem environments look like this:

\begin{theorem}[Gauss] For a closed surface $S$ enclosing a volume $V$, we have
\[ \oint_S \vec{u} \cdot dS = \int_V (\nabla \cdot \vec{u}) \, dV \ . \tag{1} \]

1.2 Proof environments

A proof environment might look like this:

\begin{proof}[Proof of Theorem 1] The proof of the theorem goes here.
\end{proof}
The enhanced theorem and proof environments provided by this package allow to pair theorems with proofs, automatically generating references from one to the other (see section 5).

1.3 Theorem-heading environments

Finally, theorem-heading environments are formatted like theorems, but the heading title is set as an argument to the environment. These environments are a nice alternative for definitions, and look like this:

\[ \text{Trace Distance. The 'trace distance' between } \rho \text{ and } \sigma \text{ is defined as} \]
\[ \delta(\rho, \sigma) = \frac{1}{2} \| \rho - \sigma \|_1, \]
where \( \| \cdot \|_1 \) is the Schatten-1 norm.

2 Quick start and package options

Example: Load the rich theorem set, with separate counters, and with proof-ref mechanism on and always displaying the proof reference in the margin:

\begin{verbatim}
\usepackage[thmset=rich,sepcounters=true,proofref={always,margin}]{phfthm}
\end{verbatim}

By default, some styles are tweaked a bit so that they appear nicely as documented below (for example, by using a filled square instead of a simple square for end-of-proof QED markers). Use the package option \texttt{resetstyle} to instruct \texttt{phfthm} not to proceed to these style adjustments; this allows you to enable features individually and selectively:

\begin{verbatim}
\usepackage[resetstyle,prooftitleitbf=true]{phfthm}
\end{verbatim}

2.1 Predefined theorem environments

You may load predefined theorem sets via the package option \texttt{thmset}. Theorem sets group common environments used in mathematical works such as Theorem, Proposition, Definition, etc.

Some package options control the way these environments are defined. If you would like more refined control over the appearance of these environments, or over which environments are defined, you may consider calling \texttt{\phfLoadThmSet} manually or defining individual environments with \texttt{\phfMakeTheorem}.

Possible theorem sets are:
Do not define any environment at package loading. You may of course invoke \phfLoadThmSet or \phfMakeTheorem manually at any later point.

Define the environments theorem, proposition, lemma and corollary as theorem-like environments, and definition as a definition-like environment.

Define the environments theorem, proposition, lemma, corollary, conjecture, remark as theorem-like environments, and definition as a definition-like environment.

In case you prefer to type less: the same environments are defined as the default set, but with shorter names. Define the environments thm, prop, lem, cor, conj, rem as theorem-like environments, and defn as a definition-like environment.

Provides the same environments as the default theorem set, as well as in addition: idea, question, claim, observation, and problem as theorem-like environments.

You may also load a theorem set at a later point after loading the phfthm package by invoking the \phfLoadThmSet macro, see subsection 3.2.

Further package options modify the style of the theorem-like and definition-like environments defined via the \thmset package option:

Use this package option to specify which style to use for theorem-like environments when loading the theorem set specified via the \thmset package option. The theorem style name should be one of plain, definition, remark, or any other \newtheoremstyle-defined theorem style (see documentation of amsthm).

Use this package option to specify which style to use for definition-like environments when loading the theorem set specified via the \thmset package option. The theorem style name should be one of plain, definition, remark, or any other \newtheoremstyle-defined theorem style (see documentation of amsthm).

Further options control various aspects of the environments defined by \thmset.
Each theorem environment defined with the \texttt{thmset} package option will use a separate counter if this option is set; otherwise (the default), there is a single counter which is shared by all those theorem environments.

Theorem counters will reset each time the parent counter is increased. Use this option for per-chapter or per-section numbering (use parent counter \texttt{parentcounter=chapter} or \texttt{parentcounter=section}). See also the \texttt{countpersection} package option.

The \texttt{proofref} package option allows to specify a comma-separated list of attributes to apply to the proof reference (“proof on page XYZ”) displayed along with the theorem. The following attributes may be specified:

\begin{itemize}
  \item \texttt{proofref={}}
  \item \texttt{proofref={default}}
    \begin{itemize}
      \item Do not change the default proof reference appearance.
    \end{itemize}
  \item \texttt{proofref=false}
    \begin{itemize}
      \item Deactivate the proof-ref mechanism.
    \end{itemize}
  \item \texttt{proofref={margin,...}}
    \begin{itemize}
      \item The proof reference is displayed in the margin, instead of after the theorem.
    \end{itemize}
  \item \texttt{proofref={marginbottom,...}}
    \begin{itemize}
      \item The proof reference is displayed in the margin, but aligned with the bottom of the theorem statement.
    \end{itemize}
  \item \texttt{proofref={longref,...}}
    \begin{itemize}
      \item The proof reference is displayed as a full sentence (“The proof of this \texttt{Theorem Name} can be found on page \texttt{XYZ}.”).
    \end{itemize}
  \item \texttt{proofref={off,...}}
    \begin{itemize}
      \item Turn off the proof reference mechanism completely for theorems defined with the \texttt{thmset} option.
    \end{itemize}
  \item \texttt{proofref={always,...}}
    \begin{itemize}
      \item Always display the proof reference, even if the proof is on the same page or on a nearby page.
    \end{itemize}
\end{itemize}

\textbf{Note:} this option has a global effect.

\begin{itemize}
  \item \texttt{proofref={onyifveryfar,...}}
    \begin{itemize}
      \item The proof reference is only displayed if the proof is at least two pages back, or four pages ahead.
    \end{itemize}
\end{itemize}

\textbf{Note:} this option has a global effect.
NOTE

The two package options \texttt{proofref=\{always\}} and \texttt{proofref=\{onlyifveryfar\}} apply to \textit{all} theorem environments which use the proof-ref mechanism, whether they have already been defined or not (see \texttt{\textbackslash phfProofrefPageBackTolerance} and \texttt{\textbackslash phfProofrefPageAheadTolerance}). All the other above options apply only to the theorem environments defined via the \texttt{thmset} package option.

2.2 The proof environment

By default, the \texttt{phfthm} package overrides the \texttt{proof} environment with a the package's own enhanced version. If you want to preserve the original \texttt{AMS} environment, you should use the \texttt{proofenv=false} package option.

\texttt{proofenv=\{true | false\}}

If set to \texttt{true}, then define an enhanced \texttt{proof} environment when loading this package. This will override any previously existing \texttt{proof} environment such as \texttt{AMS}.

If set to \texttt{false}, no action is taken at package loading time. You should then directly use the \texttt{\textbackslash phfMakeProofEnv} macro to define proof environments.

If you want finer control over how the proof environment is defined, or if you want to customize its appearance, you should use the \texttt{\textbackslash phfMakeProofEnv} macro directly (subsection 4.1).

If you set \texttt{proofenv=true}, there are a couple package options which alter the way the proof displays:

\texttt{smallproofs=\{true | false\}}

If set to \texttt{true}, then proofs display in a smaller font.

\texttt{qedsymbolblacksquare=\{true | false\}}

If set to \texttt{true}, the QED end-of-proof symbol (usually “□” with \texttt{amsthm}) is replaced by a filled square (“■”).

\texttt{prooftitleitbf=\{true | false\}}

If set to \texttt{true}, then the proof title (“Proof” or “Proof of Theorem 1”) is typeset in bold italic font.
2.3 The theorem-heading environment

By default, the \texttt{thmheading} environment is provided by the \texttt{phfthm} package:

\begin{thmheading}{Trace Distance}
The ‘trace distance’ between $\rho$ and $\sigma$ is defined as
\begin{equation}
\delta(\rho,\sigma) = \frac{1}{2} \left\| \rho - \sigma \right\|_1,
\end{equation}
where $\left\| \cdot \right\|_1$ is the Schatten-1 norm.
\end{thmheading}

\begin{table}[h]
\centering
\begin{tabular}{|l|}
\hline
\textbf{Trace Distance.} The ‘trace distance’ between $\rho$ and $\sigma$ is defined as \hfill (3) \\
\text{\textit{where $\left\| \cdot \right\|_1$ is the Schatten-1 norm.}} \\
\hline
\end{tabular}
\end{table}

You may also use \texttt{\label} and \texttt{\ref} as usual (\texttt{\ref} simply displays the given title).

Some package options control the way this environment is defined.

\begin{itemize}
\item \texttt{thmheading} = \texttt{true} | \texttt{false}
\end{itemize}

Define the environment \texttt{\begin{thmheading} ... \end{thmheading}} when loading the \texttt{phfthm} package, with reasonable default settings.

\begin{itemize}
\item \texttt{thmheadingstyle} = \texttt{(theorem style)}
\end{itemize}

If \texttt{thmheading} = \texttt{true} was specified, you may use this option to specify the theorem style to use for the \texttt{thmheading} environment. Possible values are \texttt{AMS} theorem style names (e.g. the base styles \texttt{plain}, \texttt{definition} or \texttt{remark}), or any other style defined with \texttt{\newtheoremstyle}.

If you want to define theorem-heading environments manually, see subsection 6.1.

3 Theorem environments

A theorem environment is based on the environment furnished by \texttt{amsthm}'s \texttt{\newtheorem} command, but with added goodies.

3.1 Define theorem environments manually

\begin{itemize}
\item \texttt{\phfMakeTheorem}
\end{itemize}

If you don't want to load a full theorem set (subsection 2.1), you can define
theorem environments individually with \phfMakeTheorem:

\phfMakeTheorem[(key-value options)]{(theorem environment name)}{(theorem name)}

This command defines a new environment (given as the first mandatory argument) which behaves as a theorem and is displayed as given by the second mandatory argument. For example, we might call
\phfMakeTheorem{theorem}{Theorem} to define the environment
\begin{theorem}...\end{theorem} which displays “Theorem N. . .”

The possible key-value options for the optional argument are:

counter=(\LaTeX counter | (empty))

The name of the \LaTeX counter to use for the theorem environment. If this is empty, then a new counter will be created which is specific to this theorem environment (the default). If not empty, then the theorem environment uses the given counter (or an alias thereof, see aliascounter).

If a counter is specified, the counter should already be defined with \LaTeX's \newcounter.

aliascounter=(true | false)

In some cases (e.g. if you're using hyperref's \autoref), it is important to have counters specific to each theorem environment (so you get “Theorem 5” or “Proposition 5” right). However, you may want different theorem environments to share a same logical counter (Say “Definition 1”, “Definition 2”, “Theorem 3”, “Proposition 4”). In this case, you should specify aliascounter=true.

When this option is on, then first we define an alias counter of the one given to the counter option, and then use the alias for the theorem environment. The alias is declared using the aliascnt package. The alias counter is automatically set up correctly for using \autoref.

Note that the aliascounter option only has an effect if the counter option is set to some non-empty value. If counter is set to a non-empty value, then aliascounter defaults to true.

thmstyle=(theorem style | (empty))

The theorem style to use to define this theorem environment. The value of this option should be a valid argument to AMS's \theoremstyle. If you leave this empty (the default), then the theorem style is not set explicitly and whatever default style is used.

defnostar=(true | false)

Set this to true if you want the corresponding non-starred theorem environment to be defined, e.g. \begin{theorem}...\end{theorem}.

Normal (non-starred) versions of the environments have an associated theorem number, as you expect by default.
defstar=(true | false)

Set this to true if you want the corresponding starred theorem environment to be defined, e.g. \begin{theorem*} ... \end{theorem*}.

Starred versions of the environments do not have an associated theorem number.

proofref=(true | false)

Enable or disable the proof-ref mechanism for this theorem environment (enabled by default).

proofrefstyle=(proof-ref style)

The style to use for the proof references. Here you may specify how the proof ref appears, for example (in the margin, long sentence, ...). Possible styles are proofrefstyle=default (the default), proofrefstyle=margin (display the proof ref in the margin of the page), proofrefstyle=marginbottom (display the proof ref in the margin of the page aligned with the bottom of the theorem statement) and proofrefstyle=longref (as by default but with a full sentence). See subsection 5.3 for how to further customize the appearance of the proof reference.

For example, you may use the following command invocation to define a theorem environment named “Remark” implemented as \begin{remark} ... \end{remark}, also with a starred verison \begin{remark*} ... \end{remark*}, using the plain AMS theorem style, and without the proof-ref mechanism:

\phfMakeTheorem[defstar=true,defnostar=true,thmstyle=plain,counter=,proofref=false]{remark}{Remark}

3.2 Loading theorem sets manually

You may load theorem sets at any time via the macro \phfLoadThmSet. This may be useful, for example, to load theorem sets only after you have defined a custom theorem style. The syntax of \phfLoadThmSet is:

\phfLoadThmSet{(options to \phfMakeTheorem for theorem-like environments)}{(options to \phfMakeTheorem for definition-like environments)}{(name of theorem set to load)}

The first and second argument to this macro are tokens to expand in front of \phfMakeTheorem for theorem-like or definition-like environments. For example:

\newcounter{mythmcounter}
\newtheoremstyle{mythmstyle}{...}
\phfLoadThmSet{...}
WARNING

The first and second arguments to \phfLoadThmSet must either be empty, or be enclosed in square braces.

\begin{itemize}
\item \texttt{\textbackslash theoremname}
\item \texttt{\textbackslash propositionname}
\item \texttt{\textbackslash lemmaname}
\item \texttt{\textbackslash corollaryname}
\item \texttt{\textbackslash conjecturename}
\item \texttt{\textbackslash remarkname}
\item \texttt{\textbackslash definitionname}
\item \texttt{\textbackslash ideaname}
\item \texttt{\textbackslash questionname}
\item \texttt{\textbackslash claimname}
\item \texttt{\textbackslash problemname}
\end{itemize}

The title of the theorem environments defined in theorem sets use the same scheme as figures, tables, etc. with regard to translations and babel: they use \texttt{\textbackslash theoremname}, \texttt{\textbackslash propositionname}, etc.

This package is language agnostic (with titles defined by default in English), and does not provide the titles for other languages. In order to support language switching with babel and \texttt{\selectlanguage}, you should add the relevant names to the corresponding \texttt{\textbackslash captions\langle language name\rangle} macro, for example:

\begin{verbatim}
\usepackage[french,...]{babel}
...
\addto\captionsfrench{%
  \def\theoremname{Th\textsuperscript{o}r\textsuperscript{e}me}%
  \def\propositionname{Proposition}%
  \def\lemmaname{Lemme}%
  \def\corollaryname{Corollaire}%
  \def\conjecturename{Conjecture}%
  \def\remarkname{Remarque}%
  \def\definitionname{D\textsuperscript{e}finition}%
  \def\ideaname{Id\textsuperscript{e}e}%
  \def\questionname{Question}%
  \def\claimname{Affirmation}%
  \def\problemname{Probl\textsuperscript{e}me}%
}
... \selectlanguage{french} ...
\end{verbatim}

\section*{3.3 Theorem hooks}

Any theorem environment automatically calls some hooks. There are hooks available per theorem environment as well as generic for all theorem environments.

The hook \texttt{\textbackslash phfthm\hook@start@\langle theorem environment name\rangle\{\langle theorem title\rangle\}} is called at the start of the environment. More precisely, it is called inside the original amsthm base environment; that is, after the heading was generated. It takes one mandatory argument, the optional title provided to the theorem
environment which may be empty. By default, the hook-defers to the global hook `\fthm@hook@startcommonnostar`.

The hook `\fthm@hook@start@\langle\text{starred theorem environment name}\rangle` is completely analogous, and is called for the starred environment. The only difference is that by default, it defers its call to `\fthm@hook@startcommonstar`.

The hook `\fthm@hook@startcommonnostar\langle\text{theorem environment}\rangle\langle\text{theorem optional given title}\rangle` collects the default definitions for non-starred environments (none by default) and continues to defer to `\fthm@hook@startcommon\langle\text{theorem environment}\rangle\langle\text{theorem optional given title}\rangle`. Analogously, the macro `\fthm@hook@startcommonstar\langle\text{theorem environment}\rangle\langle\text{theorem optional given title}\rangle` groups commands for starred environments (typically doesn't take care of `\label` stuff) and also defers to `\fthm@hook@startcommoncommonstar`.

The end hooks work very much analogously. `\fthm@hook@end@\langle\text{theorem environment name}\rangle` and `\fthm@hook@end@\langle\text{starred theorem environment name}\rangle` are called respectively for the non-starred and starred version of that theorem environment, and by default they defer to the common `\fthm@hook@endcommonnostar\langle\text{theorem environment name}\rangle` or `\fthm@hook@endcommonstar\langle\text{theorem environment name}\rangle`. Both these hooks defer their calls to `\fthm@hook@endcommon\langle\text{theorem environment}\rangle`. For theorems using the proof-reference mechanism, i.e. for which `\proofref=true` was specified to `\MakeTheorem` and which uses the `\label` hack (section 5), there is an additional hook. The hook `\fthm@hook@afterlabel@\langle\text{theorem environment name}\rangle` is called just after the `\label` command corresponding to the theorem is encountered (this should always be at the beginning of the theorem, see section 5). Depending on the proof-ref style, this hook may be used to generate the proof reference text (for example, with the margin proof-ref style). The hook is called after the theorem label is set. The label itself can be recovered from the value of the macro `\fthm@val@thmlabel`. By default, that hook calls the common hook `\fthm@hook@afterlabelcommon\langle\text{theorem environment name}\rangle`. (After the first occurrence of the command `\label`, the latter's definition is restored.)

# 4 Proof environments

Proof environments typeset mathematical proofs. The proof environment(s) provided by `phfthm` give some added functionality with respect to the AMS-default proof environment, such as supporting the proof-reference mechanism described in section 5.

A proof environment might look like the following:
Proof of Theorem 5. Let $\mathcal{T}_{X \rightarrow X'}$ be any trace nonincreasing completely positive map such that $\mathcal{T}_{X \rightarrow X'}(\Gamma_X)$ lies within the support of $\Gamma_{X'}$. Define the normalized state $\gamma_X = \Gamma_X / \text{tr} \Gamma_X$.

Now consider this and that … ■

The proof environments defined by this package wrap a given proof display environment (such as AMS' (amsthm)'s or IEEEtran's original proof environment) by adding functionality in the form of hooks. In the following, we refer to the “underlying proof display environment” as the original environment which is wrapped. It may be any \LaTeX environment whose task is to format the proof nicely.

4.1 Manually define a proof environment

\phfMakeProofEnv You may use the macro \phfMakeProofEnv to declare a new proof environment. The syntax is:

\phfMakeProofEnv[(key-value options)]{(proof environment name)}

This defines a new environment with the given name, which may be used to display proofs to theorems. The options may be:

\textbf{displayenv}=(name of \LaTeX environment)

Set a \LaTeX environment to use to actually format and display the proof. (The \phfMakeProofEnv command itself doesn't care about how the proof is displayed or formatted; rather it adds a goodies infrastructure in which stuff can be plugged in and provides options for such goodies.)

You may specify here the name of a \LaTeX environment, or give the special value \texttt{displayenv=*} to indicate the default appearance provided by phfthm, or leave the value empty \texttt{displayenv=} to signify that no underlying display environment should be invoked. (The latter may be useful if you are plugging a \phfMakeProofEnv-generated environment into a larger environment which already takes care of the display.)

\textbf{defaultproofname}=(default proof title)

Specify here the title to use (e.g. “Proof”) if no argument was given to the proof environment. If you do not specify any \texttt{defaultproofname}, or pass an empty value, then the value of \texttt{\proofname} is used.

\textbf{parselabel}=(true | false)

Specify whether the environment should parse its argument for some special information. If set to true, then the proof argument is passed on to a command (specified by the \texttt{parselabelcmd} option).

\textbf{parselabelcmd}=(\LaTeX macro)

If parselabel is set to true, then specify here a \LaTeX command which parses whatever it wants from the proof environment’s argument.
The macro should set the `\phfthm@val@displayargs` macro to tokens which will be expanded just after the invocation of the proof environment's display environment (`\displayenv`). It should also set `\phfthm@val@proofoflabel` (if appropriate) to the label corresponding to the theorem for which this is the proof of.

By default, the command `\phfthm@proof@parselabel` is used, which parses the proof environment's argument for a reference to a theorem in the context of a proof-ref mechanism (see section 5). The label is parsed to see if it is of the form `[*thm:reference]`, where `thm:reference` is the label pinned to a theorem.

\[\textbf{override=\{true | false\}}\]

Whether to override any existing environment with the same name as the new proof environment. If `true` is specified here, then `\renewenvironment` is used to define the proof environment, otherwise a simple `\newenvironment` is used.

\[\textbf{internalcounter=\{name of \LaTeX counter\}}\]

The name of the internal counter the proof environment should use. The count number is not displayed (by default at least), but it is only used to pin down anchors for PDF hyperlinks.

The counter should already be defined with `\newcounter`.

\[\textbf{proofofname=\{\LaTeX macro\}}\]

Specify here a macro which will be called with a single argument. The macro produces the text to display when the proof environment is parsed as the proof of a specific theorem or proposition (or other theorem environment). The argument which will be given to it is the title of what the proof is of (e.g. “Theorem 3”). Typically, the macro should produce something like “Proof of Theorem XYZ.”

By default, the global macro `\proofofname` is used.

\[\textbf{\proofname} \text{ Text to use to display "Proof." This should be already defined by the \LaTeX system, and \texttt{babel} should already provide translations in different languages.}\]

\[\textbf{\proofofname} \text{ The globally defined macro `\proofofname` specifies the default way of displaying "Proof of Theorem 5." It is originally defined as something like}\]

\[\newcommand\proofofname[1]{\proofname{} of \#1}\]

You may override this to obtain something fancier, if you wish to display the document in a different language. For instance, you might want to display (“Démonstration (Théorème 5)” in french:

\[\textbf{\def\proofofnameinparentheses#1\{\proofname\ (#1)\}}\]

\[\\addto\captionsfrench{\%

13\]


4.2 Proof hooks

The proof hooks are relatively straightforward. All hooks presented here take no argument.

Information about the argument of the proof, both the raw argument and the possibly parsed reference, are available as macros to some of the hooks (but don't change these values unless you know what you're doing). The macro \phfthm@val@proofarg contains the raw argument to the proof environment, and is available to all hooks. If you use the default proof environment argument parsing (which you must have enabled when calling \phfMakeProofEnv), then additionally the macros \phfthm@val@prooftitle and \phfthm@val@proofofname are available containing, respectively, the label of the theorem which is referenced, and the displayable reference to it (e.g. “Theorem 5”). The last two macros are available to all hooks except the first one (..@start).

The hooks named ..@start... are called within the call to \begin{<proof environment>}.  

\phfthm@hookproof@...@start

The hook named \phfthm@hookproof@(<environment name>@@start is called at the very beginning of the proof environment.

\phfthm@hookproof@...@startafterdisplay

The hook named \phfthm@hookproof@(<environment name>@@startafterdisplay is invoked immediately after the beginning of the underlying “display” environment (the environment used to display the proof contents).

\phfthm@hookproof@...@startlast

The hook named \phfthm@hookproof@(<environment name>@@startlast is called after we are sure that an anchor has been pinned down for the proof. This hook is called last within the commands in \begin{<proof environment>}.

\phfPinProofAnchor

By the way, the macro \phfPinProofAnchor may be used within the hooks to pin down an anchor for referring to the proof (especially via the proof-ref mechanism). Just call it anywhere appropriate (a good idea is calling it after leaving v-mode before displaying the title, in order to avoid placing it just before a page break). If you do not call this macro, it is automatically called for you just before the ...@startlast hook.
The two following hooks are called within the call to `\end{proof environment}`.

- `\phfthm@hookproof@...@end` The hook `\phfthm@hookproof@{environment name}@end` is called before the proof display environment is closed.

- `\phfthm@hookproof@...@final` The hook `\phfthm@hookproof@{environment name}@final` is called after the proof environment display is finished, as the very last.

All proof hooks call are defined by default to defer their call to a common hook. The common hooks each take one argument (the proof environment name). They are named `\phfthm@hookproof@startcommon{environment name}`, `\phfthm@hookproof@startafterdisplaycommon{environment name}`, `\phfthm@hookproof@startlastcommon{environment name}`, `\phfthm@hookproof@endcommon{environment name}`, and `\phfthm@hookproof@finalcommon{environment name}`. They are all defined to be empty by default.

### 5 Pairing theorems to proofs and proof-reference mechanism

One of the goodies provided by the `phfthm` package is the proof-ref mechanism, where in a theorem environment, the text “see proof on page …” is displayed to direct the reader to the location of the corresponding proof. The mechanism is deactivated by default, but can be enabled with a simple package option.

This only works if the proof is given the label of the corresponding theorem or proposition. For example:

```latex
\begin{theorem}[Gauss]
\label{thm:Gauss}
For a closed surface $S$ enclosing a volume $V$, we have
\begin{equation}
\oint_S \vec u \cdot d\vec S = \int_V (\vec\nabla \cdot \vec u) \, dV.
\end{equation}
\end{theorem}
```

```latex
\begin{proof}[*thm:Gauss]
\end{proof}
```

The above example might produce the following output:
**Theorem 17** (Gauss). For a closed surface $S$ enclosing a volume $V$, we have

$$\oint_S \vec{u} \cdot d\vec{S} = \int_V (\vec{\nabla} \cdot \vec{u}) \, dV .$$

(41)

*Proof on page XXX.*

... 

**Proof of Theorem 17.**  ...

A positive side-effect is that in your LaTeX source, if you consistently label your proofs then you always know what theorem a proof refers to. If you would like to simply output “Proof.” instead of “Proof of Theorem X.”, for instance, because the proof immediately succeeds the theorem in question, you may use the syntax \[**thm:label\] instead. In this case, the proof-reference mechanism will know that this is the proof of the theorem labeled thm:label and will avoid undefined references. For example, the following code:

\begin{theorem}[Gauss]
\label{thm:Gauss}
For a closed surface $S$ enclosing a volume $V$, we have
\begin{equation}
\oint_S \vec{u} \cdot d\vec{S} = \int_V (\vec{\nabla} \cdot \vec{u}) \, dV .
\end{equation}
\end{theorem}

\begin{proof}[**thm:Gauss]
...
\end{proof}

might output:

**Theorem 17** (Gauss). For a closed surface $S$ enclosing a volume $V$, we have

$$\oint_S \vec{u} \cdot d\vec{S} = \int_V (\vec{\nabla} \cdot \vec{u}) \, dV .$$

(41)

**Proof.**  ...

5.1 On the theorem side

On the theorem side, the proof-ref mechanism works by hacking into the definition of \LaTeX\'s \label. The \label command should be placed first within the theorem (see example above). It is important, in theorems which use the proof-ref mechanism (on by default), to always have a corresponding label: Indeed, you may experience weird results if you don't have a theorem label, but then have labels for other objects in the theorem such as equations or itemize items.

Once the corresponding proof is detected (a proof environment with an optional argument of the form \[\text{[\textit{thm:the}\text{-}label]}\] for the same label thm:the-label
as specified to the theorem, see subsection 5.2), then a text is generated (by default “Proof on page …”) and placed after the theorem. The appearance of this text is customizable (subsection 5.3).

More precisely, the hack with the \label command works as follows: At the beginning of the theorem, the \label command is redefined so that at its first occurrence, it stores its argument as the theorem’s label to use for the proof reference, it then pins down a BiX label as the original \label command would do, and finally it calls the \afterlabel theorem hook (see 3.3). After the first occurrence of \label, the \label command is restored to its original BiX meaning in case there are other objects within the theorem which are to be referred to.

TIP
The \label hack is only active within theorem environments where the proof-ref mechanism has been enabled. Outside these environments, the \label macro retains its original BiX definition.

\noproofref If, for any reason, you do not want to make sure you don’t have any text “Proof on page …” appearing (for example there is no corresponding proof because the theorem is obvious), then you should call \noproofref immediately inside the theorem:

\begin{theorem}
   \noproofref
   Theorem text ...
\end{theorem}

The command \noproofref temporarily disables the proof-ref mechanism (and restores \label to BiX’s original meaning) for the current theorem.

5.2 On the proof side

On the proof side, you just need to specify for which theorem this is the proof of. For that (unless you override the defaults and plug in your own magic parsing; see subsection 4.1), you should specify an optional argument to the proof which is of the following form: \begin{proof}[*\langle label\rangle], where \langle label\rangle is the label name you have associated with the theorem in question (see example above).

This has two effects: it sets the proof to display “Proof of …,” and also does some background dark magic to display, at the location of the corresponding theorem, some text like “Proof on page …,” where the page number corresponds to the page on which this proof is located.

You may also use the syntax \begin{proof}[**\langle label\rangle], which has the effect of producing the simple output “Proof.” (this is desirable e.g. if the proof immediately follows the theorem statement) but which still informs the proof-reference
mechanism that this proof is associated with the given theorem labeled by 
\langle \text{label} \rangle. This can avoid undefined references, and can simplify your job if you want to move proofs around in your document.

(It would be nice, of course, to have the package automatically output “Proof” only if the proof immediately follows the theorem and output “Proof of XXX” if the proof is further away. Unfortunately this is hard to detect in \LaTeX and I currently have no plans to try to implement this.)

5.3 Customizing appearance of the proof reference text

Here we explain the workings of the \texttt{\textbackslash phfthm@proofrefstyle@...} macros and how they are called. It allows you to define new proof-ref styles, for example.

When the option \texttt{proofref=true} is given to \texttt{\textbackslash phfMakeTheorem} to define an environment (say \texttt{mytheoremenv}), then the hook \texttt{\textbackslash phfthm@hook@start@mytheoremenv} will automatically include the following calls:

- The macro \texttt{\textbackslash phfthm@proofrefstyle@〈proof-ref style〉@setup} is called (for the proof-ref style given via the \texttt{\textbackslash proofenvstyle=〈style name〉} key-value option to \texttt{\textbackslash phfMakeTheorem});

- The macro \texttt{\textbackslash phfthm@def@label@thmlabel} is invoked, implementing the hack on the \texttt{\textbackslash label} macro;

- The macro \texttt{\textbackslash phfthm@proofref@impl@start} is called. This macro is expected to be defined after calling \texttt{\textbackslash phfthm@proofrefstyle@〈proof-ref style〉@setup}.

Furthermore, the \texttt{\textbackslash phfthm@hook@afterlabel@mytheoremenv} hook will include a call to \texttt{\textbackslash phfthm@proofref@impl@afterlabel{〈label of the theorem〉}}. Again, the latter macro is expected to be defined after calling \texttt{\textbackslash phfthm@proofrefstyle@〈proof-ref style〉@setup}.

Finally, the hook \texttt{\textbackslash phfthm@hook@end@mytheoremenv} includes a call to \texttt{\textbackslash phfthm@proofref@impl@end{〈label of the theorem〉}}. Again, the latter macro is expected to be defined after calling \texttt{\textbackslash phfthm@proofrefstyle@〈proof-ref style〉@setup}.

Hence, to define a new proof-ref style, you simply need to define a macro called \texttt{\textbackslash phfthm@proofrefstyle@〈PROOF-REF-STYLE-NAME〉@setup}. This macro should include commands to locally define the macros \texttt{\textbackslash phfthm@proofref@impl@start}, \texttt{\textbackslash phfthm@proofref@impl@afterlabel}, and \texttt{\textbackslash phfthm@proofref@impl@end}.

Different proof-ref styles may work similarly and want to share most of the code. A good idea is to build upon the default proof-ref style, which is highly modular and can be instantiated in different flavors. For an example, check the margin
proof-ref style which does precisely that. For more documentation, check out the implementation of the default proof-ref style in subsection 7.4.3.

6 Theorem-heading definition-like environments

A theorem-heading environment is an environment which displays in the same way as a theorem environment, but where the title may be any text (say, “Trace Distance” instead of, e.g., “Theorem 5”).

By default, the phfthm package provides the thmheading environment (see subsection 2.3).

6.1 Define theorem-heading environments manually

A new theorem-heading environment can be defined by calling \phfMakeThmheadingEnvironment. The syntax is:

\phfMakeThmheadingEnvironment[(key-value options)]{(environment name)}

The key-value options may be any combination of the following:

\texttt{thmstyle=(theorem style name)}

The theorem style to use to display the environment. You may specify here any default AMS style (plain, remark or definition), or any other \newtheoremstyle-defined style.

\texttt{internalcounter=(name of counter)}

The name of a counter which will internally track environment instances. By default, a common internal counter is used for all theorem-heading environments (named phfthmheadingcounter). The counter must be already defined (see \LaTeX's \newcounter).

You can also use \label and \ref (the latter simply displays the given title).

6.2 Available hooks for theorem-heading environments

The hook \phfthm@hook@thmheading@...@start{(title)} is invoked at start, within an internal environment created with \newtheorem. This hook accepts one argument, the title of the theorem-heading.

The hook \phfthm@hook@thmheading@...@end is called at the end, but still within the internal theorem environment.

Replace the dots with the name of the theorem-heading environment (such as thmheading).
By default, these hooks simply call the common hooks \phfthm@hook@thmheading@start and \phfthm@hook@thmheading@end. These common hooks are empty by default.

## 7 Implementation

First, load some packages. General toolboxes:

1 \RequirePackage{xkeyval}
2 \RequirePackage{etoolbox}

To define alias counters for theorems, load aliascnt:

3 \RequirePackage{aliascnt}

And finally, load the AMS math and theorem (amsmath, amsthm) packages:

4 \RequirePackage{amsmath}
5 \RequirePackage{amsthm}

### 7.1 Generic Internal Stuff

\phfthm@internal@execattribs Internal command: execute all definitions given in list of attributes. This was copy-pasted from a similar definition in the phfntote package.

#1 = prefix to look for attributes
#2 = name of what #1 represents, to use in message in case attribute is not found
#3 = list of attributes

6 \def\phfthm@internal@execattribs#1#2#3{%
7 \@for
next:=#3\do{%
8 \ifcsname #1\next\endcsname%
9 \csname #1\next\endcsname%
10 \else%
11 \PackageWarning{phfthm}{Unknown #2: '\next'. Ignoring.}
12 \fi
13 }
14 }

### 7.2 Definitions for theorem environments

7.2.1 \phfMakeTheorem: definition of a new theorem environment

First, define some key-value syntax accepted by the \phfMakeTheorem command.
\phfMakeTheorem

Define a new theorem environment. The syntax is \phfMakeTheorem
\texttt{[(options)]{(theorem environment name)}{(Theorem Display Name)}}. For
example: \phfMakeTheorem\texttt{[counter=thmcounter]{}{prop}\{Proposition\}}

Handle the \texttt{[options]}. First, ensure that the defaults are set, and then, parse the
input.

\setkeys{phfmkthm}{#1}

Now, react to whatever was given in the options.

Set the theorem style, if requested.\footnote{The construct \texttt{\if\relax\detokenize...\relax} tests whether \ldots is empty: see \url{http://tex.stackexchange.com/a/53091/32188}}

If requested, define the default, unstarred version of the theorem. Use
\texttt{\newtheorem} for that, which we make sure to call appropriately depending
on whether a separate counter is requested or not. Make sure also to define
\texttt{\autorefname} for \texttt{\autoref}. If an alias counter is requested, create it and
pass that one to \texttt{\newtheorem}.

At this point, we create a theorem named \texttt{phfthm@...} using \texttt{\newtheorem}
(because we still want to add calls to hooks).
— in case we use a separate given counter (if counter= is given empty or not specified). Also check if we want a parent counter (per-section or per-chapter numbering) and take that into account:

\ifrelax\detokenize\expandafter{\cmdKV@phfmkthm@parentcounter}\relax%  
\edef{x}{\noexpand\newtheorem{phfthm@#2}{#3}}%  
\else  
\edef{x}{\noexpand\newtheorem{phfthm@#2}{#3}{\expandonce\cmdKV@phfmkthm@parentcounter}}%  
\fi  
\x  
\csdef{phfthm@#2autorefname}{#3}%  
\else%  
\ifKV@phfmkthm@aliascounter%  
in case we make a distinct alias counter, eg. for use with \autoref:

\newaliascnt{#2}{\cmdKV@phfmkthm@counter}%  \newtheorem{phfthm@#2}[#2]{#3}%  \aliascntresetthe{#2}%  \csdef{#2autorefname}{#3}%  \else%  
in case we directly instruct \newtheorem to use the other counter (does not work with \autoref):

\newtheorem{phfthm@#2}[\cmdKV@phfmkthm@counter]{#3}%  \fi%  \fi%  

And also define the actual theorem environment, adding calls to hooks.

\newenvironment{#2}{1}{2}{1}{
\begin{phfthm@#2}{##1}%  \begingroup%  \csname phfthm@hook@start@#2\endcsname{##1}%  \}%  \csname phfthm@hook@end@#2\endcsname%  \endgroup%  \end{phfthm@#2}%

Define hooks specific to this theorem with sensible defaults. If proof-ref is on, call the appropriate callbacks. Then, call the common hooks (see \phfthm@hook@startcommonnonostar, \phfthm@hook@afterlabelcommon and \phfthm@hook@endcommonnonostar, detailed in subsection 3.3).

\csedef{phfthm@hook@start@#2}{1}{2}{1}{%  \ifKV@phfmkthm@proofref%  \expandafter\noexpand\csname phfthm@proofreffstyle\cmdKV@phfmkthm@proofreffstyle@setup\endcsname%  \noexpand\phfthm@def@label@thmlabel{#2}%
If requested, define the starred version of the theorem. We call \texttt{\newtheorem*} to define the base theorem environment (which we call \texttt{phfthm@...}), after which as above we define the actual environment which also calls the relevant hooks.

Finally, define the default hooks specific to the starred version of the theorem (see subsection 3.3).

\begin{verbatim}
\ifKV@phfmkthm@defstar%
\newtheorem*[phfthm@#2*]{#3}{#4}
\newenvironment{#2*}[1][1]{
\begin{phfthm@#2*}[##1]
\begingroup
\csname phfthm@hook@start@#2*\endcsname{##1}
}{
\csname phfthm@hook@end@#2*\endcsname
\endgroup
\end{phfthm@#2*}
\fi%
\end{verbatim}

% 7.2.2 Default hooks for theorems
Common default hooks definitions for start of the theorems.

For all three of these hooks, we have \#1 = theorem name, e.g. proposition and \#2 = full (optional) title of proposition, if given, or empty.

Make sure to invoke the \label re-definition hack only for non-starred theorems/propositions; indeed, if no theorem label is set we don't want to interfere with labels set to inner equations, itemizes etc. Hence, call \phfthm@def@label@thmlabel only in the “nostar” hook.

\def\phfthm@hook@startcommonnostar#1#2{\phantomsection\addtocounter{eqn}{1}\label{#1}#2}\def\phfthm@hook@startcommonstar#1#2{\phantomsection\addtocounter{eqn}{1}\label{#1}#2}\def\phfthm@hook@startcommon#1#2{\phantomsection\addtocounter{eqn}{1}\label{#1}#2}

Furthermore, in any case, set the \postdisplaypenalty to avoid an orphan line on a new page after display equation.
\postdisplaypenalty=10000\relax

Further hooks, for after the theorem main \label command (\phfthm@hook@afterlabelcommon) and for the end of the theorem.
\def\phfthm@hook@afterlabelcommon#1{\phantomsection\addtocounter{eqn}{1}\phfthm@hook@afterlabelcommon{#1}}\def\phfthm@hook@endcommonnostar#1{\phantomsection\addtocounter{eqn}{1}\phfthm@hook@endcommon{#1}}\def\phfthm@hook@endcommonstar#1{\phantomsection\addtocounter{eqn}{1}\phfthm@hook@endcommon{#1}}\def\phfthm@hook@endcommon#1{\phantomsection\addtocounter{eqn}{1}\phfthm@hook@endcommon{#1}}

### 7.2.3 Proof-ref mechanism (on the theorem side)

These macros enable the proof-ref mechanism (section 5). The theorem's label is stored upon calling \label, because we (locally) hack into the definition of \label. (After the first usage of \label its meaning is restored.)

\def\phfthm@def@label@thmlabel#1{\ifdefined\phfthm@old@label\PackageWarning{phfthm}{Internal inconsistency: \string\phfthm@def@label@thmlabel called twice for the same theorem environment!}\else\let\phfthm@old@label\label\edef\label{\noexpand\phfthm@thmlabel{#1}}\edef\phfthm@thmlabel{#1}\fi}

Main macro to invoke at the beginning of the theorem environment, so that the theorem label is stored in a local macro once \label is invoked. This hacks the \label macro locally. Here, \#1 = the theorem environment name, e.g. proposition.
The first call to \label within the theorem redirects to the macro \phfthm@thmlabel. (Applies to theorem environments for which \phfthm@def@label@thmlabel was called, which is the default).

Here \#1 = theorem environment name, e.g. proposition; and \#2 = the label value (argument to the \label macro).

First, store the label value into a macro called \phfthm@val@thmlabel.

Then, call the original \label macro to do what \LaTeX would normally do for a \label{...} call.

Restore the old \label definition, in case there are other items in the theorem environment such as equations, itemizes etc. which may themselves have \label’s.

Invoke the \phfthm@hook@afterlabel@thmname hook for this theorem environment.

Finally, ignore any spaces following the \label command. (Maybe we should have done something with \@bsphack and \@esphack but oh well...)

7.3 Definitions for proof environments

Improved, smarter proof environments.

Save old proof environment provided by amsthm.

And provide a default name for proofs (this should normally already be provided by amsthm).
The default counter for proofs. The value of this counter is typically not displayed, we just use it to pin down anchors for labels for cross references.

Utility: to see if an argument was specified (possibly empty) to the proof environment.

7.3.1 Define a proof environment: \texttt{\phfMakeProofEnv}

Make a proof environment. Syntax: \texttt{\phfMakeProofEnv [\langle options\rangle] \{\langle proof environment name\rangle\}}.

Parse the key-value options. First, make sure that all the defaults are set, then parse the options.

The meaning of the options are detailed in subsection 4.1.

The general idea here is first to pre-process all the options, and save all the useful information in macros named \texttt{\phfthm@prfenv@<proof-environment-name>@val@<something>}. 

26
Then, we can define the begin/end environment macros which will recall the saved information.

Take care of the display environment to use. Recall that if displayenv=*, we use our own default; if displayenv= (empty), there is no display environment. Here, we set \phfthm@prfenv<proof-environment-name>@val@displayenv to the name of the environment to use (possibly empty), for later reference.

156 \def\phfmkprf@tmp@star{*}\
157 \ifx\cmdKV@phfmkprf&displayenv\phfmkprf@tmp@star\relax\fi
158 \def\cmdKV@phfmkprf&displayenv{phfthm@proof@defaultdisplayenv}
160 \cslet{phfthm@prfenv#2@val@displayenv}\cmdKV@phfmkprf&displayenv\

Process the default proof name. If none is given, use \proofname and pass no option to the underlying display environment whenever the proof environment is called with no option. Here, we set \phfthm@prfenv<proof-environment-name>@val@defaultproofnameargs and \phfthm@prfenv<proof-environment-name>@val@setdefaultprooftitle; the former is the tokens to put in front of the proof environment invocation in case no explicit proof title is given to the proof environment while the latter contains the command to set \phfthm@val@prooftitle to the default proof name.

161 \if\relax\detokenize\expandafter{\cmdKV@phfmkprf@defaultproofname}\relax\%\n162 \csdef{phfthm@prfenv#2@val@defaultproofnameargs}{}%
163 \csdef{phfthm@prfenv#2@val@setdefaultprooftitle}{%}
164 \def\phfthm@prfenv#2@val@defaultproofnameargs{}%
166 \csedef{phfthm@prfenv#2@val@setdefaultprooftitle}{%}
168 \noexpand\def\noexpand\phfthm@val@setdefaultprooftitle{%}
170 \expandonce{\cmdKV@phfmkprf@defaultproofname}}%\n171 \fi
172 \csedef{phfthm@prfenv#2@val@parselabelandmakdisplayargs}##1{%
173 \csedef{phfthm@prfenv#2@val@parselabel}{%}
174 \expandonce\cmdKV@phfmkprf@parselabelcmd{##1}%
176 \noexpand\phfthm@proof@noparselabel{##1}%\n177 \fi
178 \noexpand\if\noexpand\relax\noexpand\detokenize\%
179 \noexpand\expandafter{\noexpand\phfthm@val@prooftitle}\noexpand\relax\%
180 \noexpand\def\noexpand\phfthm@val@displayargs{}\%
181 \noexpand\else\%
182 \noexpand\expandafter{\noexpand\phfthm@val@displayargs{}}%
184 \noexpand\csname phfthm@prfenv#2@val@proofofname\endcsname%
185 \noexpand\phfthm@val@prooftitle%\n186 \noexpand\fi
Store the macro which creates the “Proof of…” text (proofofname option).

\cslet{phfthm@prfenv@#2@val@proofofname}\cmdKV@phfmkprf@proofofname%

Create the macro which will take care of pinning down the label for the proof-ref (see section 5). This macro first ref-steps the internal counter and then pins down a label, if appropriate.

\csdef{phfthm@prfenv@#2@val@pinproofanchor}{% 
\csname phfthm@prfenv@#2@val@refstepinternalcounter\endcsname% 
\if\relax\detokenize\expandafter{\phfthm@val@proofoflabel}\relax\else% 
\edef\phfthm@tmp@larg{{proof:\phfthm@val@proofoflabel}}% 
\expandafter\label\phfthm@tmp@larg% 
\fi\]%

The command to ref-step the internal proof counter. Use the value of the internalcounter command option.

\csedef{phfthm@prfenv@#2@val@refstepinternalcounter}{% \noexpand\refstepcounter{\cmdKV@phfmkprf@internalcounter}}%

Make macros \phfthm@prfenv@<proof-environment-name>@val@displayenvbegincmd and \phfthm@prfenv@<proof-environment-name>@val@displayenvendcmd, which essentially expand to \begin{<the-display-env>} and \end{<the-display-env>} for the display environment given in the option displayenv.

\if\relax\detokenize\expandafter{\cmdKV@phfmkprf@displayenv}\relax% \csdef{phfthm@prfenv@#2@val@displayenvbegincmd}##1{}% \csdef{phfthm@prfenv@#2@val@displayenvendcmd}##1{}% \else \csedef{phfthm@prfenv@#2@val@displayenvbegincmd}##1{% \noexpand\begin{\csname phfthm@prfenv@#2@val@displayenv\endcsname}##1}%; \csedef{phfthm@prfenv@#2@val@displayenvendcmd}##1{% \noexpand\end{\csname phfthm@prfenv@#2@val@displayenv\endcsname}##1}%; \fi

See if we need to call \newenvironment or \renewenvironment, depending on the value of the override option.

\def\phfthm@tmp@defcmd{\newenvironment}%; \ifKV@phfmkprf@override\def\phfthm@tmp@defcmd{\renewenvironment}\fi%

Finally, (re-)define the environment. The default value of the optional argument is the token \phfthm@NOPROOFARG, which indicates that no argument was provided.

Start by storing the value of the argument into a macro, and then call the “start” hook (see proof hooks in subsection 4.2).
First, parse the optional argument into proof label (maybe) and proof title. If no optional argument was given, don't give any argument to the underlying display environment. If an empty argument was given, set some defaults; otherwise, use the necessary command to potentially parse the label and create the proper arguments for the underlying display environment.

Define the \texttt{\phfPinProofAnchor} command (locally) in case the display formatting environment takes care of where to place the anchor already.

Start the proof's display environment. Don't be fooled here by the curly braces after \texttt{x}, it only protects the argument to the \texttt{\phfthm@prfenv@#2@val@displayenvbegincmd} command itself: the \texttt{\phfthm@val@displayargs} are still just tokens which will be expanded in front of the \texttt{\begin{<proof-display-env>}} command.

And call the corresponding hook:
If required, pin anchor after the proof-display-environment. (\phfPinProofAnchor auto-destructs after first use, so it's safe to potentially call it a second time here). Then, call the corresponding hook.

\phfPinProofAnchor%
 expansafter\noexpand\csname phfthm@hookproof@#2@startlast\endcsname%
}%

Now, the definitions for the "end" part of the environment. Just call the relevant hooks and close the display environment.

{%
 expansafter\noexpand\csname phfthm@hookproof@#2@end\endcsname%
 \csname phfthm@prfenv@#2@val@displayenvendcmd\endcsname
 expansafter\noexpand\csname phfthm@hookproof@#2@final\endcsname%
}%

Finally, define the default values of the proof-environment-specific hooks. These just call the corresponding global hooks (see subsection 4.2).

\csdef{phfthm@hookproof@#2@start}{\phfthm@hookproof@startcommon{#2}}%
\csdef{phfthm@hookproof@#2@startafterdisplay}{%
 \phfthm@hookproof@startafterdisplaycommon{#2}}%
\csdef{phfthm@hookproof@#2@startlast}{\phfthm@hookproof@startlastcommon{#2}}%
\csdef{phfthm@hookproof@#2@end}{\phfthm@hookproof@endcommon{#2}}%
\csdef{phfthm@hookproof@#2@final}{\phfthm@hookproof@finalcommon{#2}}%
}

7.3.2 Common hooks for proofs

The hooks are documented in subsection 4.2.

\def\phfthm@hookproof@startcommon#1{}
\def\phfthm@hookproof@startafterdisplaycommon#1{}
\def\phfthm@hookproof@startlastcommon#1{}
\def\phfthm@hookproof@endcommon#1{}
\def\phfthm@hookproof@finalcommon#1{}

7.3.3 Default display environment for proofs

phfthm@proof@defaultdisplayenv Provide an environment which displays a proof in a similar fashion as \textit{AMS}, but with some small additional features.

\newenvironment{phfthm@proof@defaultdisplayenv}[1][\proofname]{%
 \par
 \pushQED{\qed}%
 \normalfont \topsep6p\@plus6p\@relax
 \trivlist\item\relax
 \phfPinProofAnchor
These macros may be overridden to change the proof title appearance.

These macros parse the argument of the proof environment to see if it is of the form `<some-label>` (see section 5).

Call \texttt{\phfthm@proof@parselabel\{\textit{proof environment argument}\}} to parse the argument string. This macro will set \texttt{\phfthm@val@proofoflabel} and \texttt{\phfthm@val@prooftitle} to appropriate values (respectively, the label name of the corresponding theorem and a representative title such as “Theorem 6”).

If the argument is [*thm-label] we want “Proof of \texttt{\ref{thm-label}}.”. If the argument is [**thm-label] we want “Proof.” but we’re informing the proofref-system that this is the proof of \texttt{\langle thm-label \rangle} (e.g., to avoid undefined references to proofs). Otherwise if we get [Some Custom Theorem] then we want “Proof of Some Custom Theorem.”.

The use of \texttt{\detokenize} here is a trick to make sure that all chars in the label text have a non-active category (e.g. we would have problems, e.g., if in the label “thm:gauss” the “:” is an active char—such as in French):
\def\phfthm@proof@noparselabel#1{\def\phfthm@val@proofoflabel{}\def\phfthm@val@prooftitle{#1}}

\phfthm@proof@noparselabel Enjoys the same syntax as $\phfthm@proof@parselabel$, i.e., it is a drop-in replacement for the latter, except that it invariably sets $\phfthm@val@proofoflabel$ to an empty value and $\phfthm@val@prooftitle$ to the argument itself. You could use this as a $\text{parselabelcmd}$ macro if you didn't want to parse the label.

In order to look up what we are a proof of, we use $\text{\autoref}$ provided by the $\text{hyperref}$ package. Crucially, it is NOT the job of $\text{phfthm}$ to load this package, and we should only use them if they are available (we don't want to depend on it). If it is not available, then we fall back to the regular $\text{\ref}$ command.

\begin{verbatim}
\def\phfthm@autoref\{\ref\}
\AtBeginDocument{\@ifpackageloaded{hyperref}{\def\phfthm@autoref{\autoref}}{}}
\end{verbatim}

\section{Implementation of the proof-ref machinery}

\subsection{Small general stuff}

$\text{\proofonname}$ The macro $\text{\proofonname}$ displays “Proof on ….” Here, \#2 is the full page reference and \#1 is the label name of the referenced theorem.

\providecommand\proofonname[2]{Proof on #2.}

$\text{\proofrefsizesize}$ Format the proof reference “Proof on page ….”. This macro is meant to set the font size (or other font properties), but it may also be defined to take one argument, the proof reference text.

\def\proofrefsizesize{\footnotesize}
Use \noproofref inside a theorem to signify that no proof reference should be attempted.

The implementation just defines \phfthm@val@noproofref. If this macro is defined, then no proof ref should be generated for the current thmlabel. Also, restore \label to its original definition in case it was overridden.

\def\noproofref{%
  \def\phfthm@val@noproofref{1}%
  \ifdef\phfthm@old@label \let\label\phfthm@old@label \fi%
}%

\phfthm@proofref@warnnolabel

Produce a warning that no label was provided in order to infer the proof reference.

\def\phfthm@proofref@warnnolabel{%
  \PackageWarning{phfthm}{No label provided for proof reference!}%
}%

\phfthm@proofref@expandthmlabeltoarg

Utility to expand the value of \phfthm@val@thmlabel as an argument to a callback command. #1 = the macro to relay the call to.

\def\phfthm@proofref@expandthmlabeltoarg#1{%
  \ifdef\phfthm@val@noproofref\relax%
    \else%
      \providecommand\phfthm@val@noproofref\relax%
      \else%
        Then make sure \phfthm@val@thmlabel is defined (maybe empty), and then either call the callback macro #1 with the value of \phfthm@val@thmlabel as argument, or generate a warning if that value is empty.

        \providecommand\phfthm@val@thmlabel{}
        \edef\phfthm@tmpa\{}\phfthm@val@thmlabel\}
        \expandafter\notblank\phfthm@tmpa{%
          \expandafter#1\phfthm@tmpa%
        }{\phfthm@proofref@warnnolabel% no label provided}
    \fi%
%}

7.4.2 Utilities for interacting with \autoref labels

In this context, we also need some generic utilities for interacting with \autoref labels.
The macro \phfthm@autorefnameof extracts the name of the counter which generated this reference (e.g. “section” or “theorem”).

\begin{verbatim}
def\phfthm@autorefnameof#1{
    Extract the counter part of the reference section.NN, which is 4th element in the r@label macro. (Code extracted from hyperref.sty.)
    \expandafter\ifx\csname r@#1\endcsname\relax%
    \textbf{??}%
    \else%
        \expandafter\expandafter\expandafter\phfthm@HyPsd@autorefname\csname r@#1\endcsname{}{}{}{}\@nil%
    \fi%
}
def\phfthm@HyPsd@autorefname#1#2#3#4#5\@nil{% 
    \ifx\#4\%
        \else%
            \phfthm@HyPsd@@autorefname\#4.\@nil%
        \fi%
}
def\phfthm@HyPsd@@autorefname#1.#2\@nil{% 
    \ltx@IfUndefined{#1autorefname}{% 
        \ltx@IfUndefined{#1name}{% 
            \csname#1name\endcsname% 
        }{% 
            \csname#1autorefname\endcsname% 
        }% 
    }{% 
        \csname#1autorefname\endcsname% 
    }% 
}
\end{verbatim}

A minimal pageref macro, which just extracts the page number on which the given label is located.

\begin{verbatim}
def\phfthm@min@pageref#1{% 
    \ifcsname r@#1\endcsname% 
        \expandafter\expandafter\expandafter\expandafter
            \expandafter\expandafter\expandafter\@car
                \expandafter\expandafter\expandafter\@gobble
                    \csname r@#1\endcsname{}
            \else% 
                0%
            \fi%
}
\end{verbatim}

The dark magic going on here is beyond me. The code was copied from hyperref.sty, in \"\def\HyPsd000@pageref\ldots\" and seems to work.

\begin{verbatim}
def\phfthm@min@pageref#1{% 
    \ifcsname r@#1\endcsname% 
        \expandafter\expandafter\expandafter\expandafter
            \expandafter\expandafter\expandafter\@car
                \expandafter\expandafter\expandafter\@gobble
                    \csname r@#1\endcsname{}
            \else% 
                0%
            \fi%
}
\end{verbatim}
7.4.3 Default proof-ref style, with basic machinery

Now we define the relevant callbacks for the default style. See documentation in subsection 5.3. Recall a proof-ref style just needs to define \texttt{\phfthm@proofrefstyle@<stylename>@setup}, which in turn should just define the callbacks \texttt{\phfthm@proofref@impl@start}, \texttt{\phfthm@proofref@impl@afterlabel} and \texttt{\phfthm@proofref@impl@end}. For our default style, these callbacks further call other callbacks of the form \texttt{\phfthm@proofref@impl@...}, such that these definitions can be re-used to create new styles. The main proof-ref generation routine is \texttt{\phfthm@proofrefstyle@default@main}, which can be used for either the \texttt{...@afterlabel} or the \texttt{...@end} callback.

\begin{verbatim}
\phfthm@proofrefstyle@default@-fmt
\end{verbatim}

Format and display the proof reference. \#1 = the theorem's label (e.g. \texttt{prop:1}); \#2 = the full reference (e.g. "page XYZ").

This macro is the default value of the callback \texttt{\phfthm@proofref@impl@fmt}, which is called by the default style itself.

Use correct spacing for right-aligning the reference.² If there is room on the current line, just right-align the proof-ref text; if not, add it on a separate line. [We can achieve this with the sequences \texttt{\hfil\null\hfil}: if there is space, it all fits on the same line, if not, the line breaks at the \texttt{\null} point.]

\begin{verbatim}
\texttt{\def\phfthm@proofrefstyle@default@fmt#1#2{%}
\texttt{\parfillskip=0pt\relax%}
\texttt{\hfil\null\hfil\null\hfil%}
\texttt{\hbox{\proofrefsize{\proofonname{#1}{#2}}}}\par%}
\texttt{362}}
\end{verbatim}

\begin{verbatim}
\phfthm@proofrefstyle@default@-fmtfarback
\phfthm@proofrefstyle@default@-fmtfarahead
\phfthm@proofrefstyle@default@-fmtcloseby
\end{verbatim}

These macros are the default values of the callbacks \texttt{\phfthm@proofref@impl@fmtfarback}, \texttt{\phfthm@proofref@impl@fmtfarahead}, and \texttt{\phfthm@proofref@impl@fmtcloseby}, which are called by the default style itself. These callbacks define how to format and (possibly not) display the proof reference depending on whether the proof is “far behind” (several pages back), “far ahead” (several pages ahead) or “close by” (neither far back nor far ahead), as defined by \texttt{\phfProofrefPageBackTolerance} and \texttt{\phfProofrefPageAheadTolerance}.

\begin{verbatim}
\texttt{\def\phfthm@proofrefstyle@default@fmtfarback#1#2{%}
\texttt{\phfthm@proofref@impl@fmtfarback#1#2}}
\texttt{\def\phfthm@proofrefstyle@default@fmtfarahead#1#2{%}
\texttt{\phfthm@proofref@impl@fmtfarahead#1#2}}
\texttt{\def\phfthm@proofrefstyle@default@fmtcloseby#1#2{}}
\end{verbatim}

\begin{verbatim}
\phfProofrefPageBackTolerance
\phfProofrefPageAheadTolerance
\end{verbatim}

The macros \texttt{\phfProofrefPageBackTolerance} and \texttt{\phfProofrefPageAheadTolerance} define how many pages back or ahead the proof should be in order to consider it “far back” or “far ahead.”

²Thanks \url{http://tex.stackexchange.com/a/43239/32188}!
Either value may be set to -1 to force the proof to be considered “far back” or “far ahead.”

Define the internal counter which allows to check on which page we are at the place of the proof reference. This is used by \texttt{phfthm@proofrefstyle@default@main}.

The main proof-ref generation routine. The argument \texttt{#1} is the current label of the theorem; the referenced label is \texttt{proof:#1}.

Check to see if the proof is far away ahead or back (as defined by the tolerance macros above). Depending on each case, call the corresponding callbacks.

If the proof is “far back,” call the corresponding callback.

If the proof is “far ahead,” call the corresponding callback.

Otherwise, it is close by.

\footnotetext{\url{http://tex.stackexchange.com/a/2526} to test whether ref is on same page. Note that was problematic, probably due to hyperref. I needed to use my own \texttt{phfthm@min@pageref} without any hyper linking mechanism in place.}
The main set-up macro for the default proof-ref style. It sets all the call-backs to the default ones.

\def\phfthm@proofrefstyle@default@setup{\let\phfthm@proofref@impl@start\relax\let\phfthm@proofref@impl@afterlabel@gobble\let\phfthm@proofref@impl@end\phfthm@proofrefstyle@default@main\let\phfthm@proofref@impl@fmtfarback\phfthm@proofrefstyle@default@fmtfarback\let\phfthm@proofref@impl@fmtfarahead\phfthm@proofrefstyle@default@fmtfarahead\let\phfthm@proofref@impl@fmtcloseby\phfthm@proofrefstyle@default@fmtcloseby\let\phfthm@proofref@impl@fmt\phfthm@proofrefstyle@default@fmt}\\

7.4.4 Other proof-ref styles: margin and marginbottom

These styles simply use the same mechanism as the default style, but plug in different sub-callbacks.

Here's a command that actually places the marginpar:

\def\phfthmproofref@placemarginpar[2]{\marginpar[\raggedleft #1]{\raggedright #2}}\\

In the marginbottom proof-ref style, we place an invisible \marginpar at the top of the proof so that the proof-ref margin note will not be shifted too high when we try to align it to the bottom of the theorem statement. To work around a weird bug in Rev\LaTeX, where I get the error message 'command sequence undefined: @captype', I have found that a fix is to override this local invisible \marginpar to locally make sure the \@captype is defined. The wild fix I've found—no guarantees, there are dark magic forces at work here—is the following, which you can include in your document preamble if you're banging your head against this problem and you're desperate:

\makeatletter\renewcommand\phfthmproofref@placephantommarginpar{\begingroup\csdef{@captype}{figure}\marginpar{\vspace{-\baselineskip}\rule{0pt}{0pt}}\endgroup%
The margin style

Set-up macro for the “margin” proof-ref style (displays the proof reference in the margin of the page).

The proof reference should be displayed directly at the top, not at the end of the theorem, so plug in \phfthm@proofref@default@main onto ...@afterlabel and not onto ...@end. Don't forget that these macros accept one argument, the theorem label.

Define the formatting callback to put the note in the margin of the page using a \marginpar. We need \leavevmode to make sure it's aligned properly vertically with the paragraph.\footnote{See \url{http://tex.stackexchange.com/a/16161/32188}}

The marginbottom style

Set-up macro for the “marginbottom” proof-ref style (displays the proof reference in the margin of the page).
The proof reference should be displayed directly at the bottom, not at the end of
the theorem, so plug in \phfthm@proofref@default@main onto \ldots@end and
not onto \ldots@afterlabel. Don’t forget that these macros accept one argument,
the theorem label. In \ldots@afterlabel we place a dummy marginpar with
an invisible rule one \baselineskip upwards, so that we force any proof-ref
marginpar not to extend above the top of the theorem statement.

\def\phfthm@proofref@impl@afterlabel##1{\%
  \phfthmproofref@placephantommarginpar
  \}
\let\phfthm@proofref@impl@end\phfthm@proofrefstyle@default@main

Define the formatting callback to put the note in the margin of the page using
a \marginpar. We use a trick to align the margin note with the bottom of the
paragraph.\footnote{https://tex.stackexchange.com/a/388018/32188}

\def\phfthm@proofref@impl@fmt##1##2{\%
  \phfthmproofref@marginbottom@domarginpar{##1}{##2}\%
  \}
\def\phfthm@proofref@marginbottom@domarginpar#1#2{\%
  \setbox\phfthm@tmp@boxa=\hbox to \marginparwidth{\parbox[b]{\marginparwidth}{\raggedleft\sloppy
    \proofrefsize{\strut\proofonname{#1}{#2}\strut}}}%
  \parbox[b]{\marginparwidth}{\raggedleft\sloppy
    \proofrefsize{\strut\proofonname{#1}{#2}\strut}}}%
  \phfthmproofref@placemarginpar{\box\phfthm@tmp@boxa}{\box\phfthm@tmp@box}\%
  \}
\newbox\phfthm@tmp@boxa
\newbox\phfthm@tmp@box

\section{Thmheading definition-like environments}

\subsection{Manually define a thmheading environment}

Define the key-value options accepted by \phfMakeThmheadingEnvironment.

\define@cmdkey{phfthmmkthmheading}{thmstyle}\{}
\define@cmdkey{phfthmmkthmheading}{internalcounter}\{}
\newcounter{phfthmheadingcounter}\%

\footnote{https://tex.stackexchange.com/a/388018/32188}
\phfMakeThmheadingEnvironment

Creates a new environment \begin{thmheading}{Title}...\end{thmheading} for customizing the heading on-the-fly (see documentation in section 6). Useful for an alternative formatting of definitions. The syntax is:

\phfMakeThmheadingEnvironment[(key-value options)]{(environment name)}

You can also use \label and \ref (the latter simply displays the given title).

\newcommand\phfMakeThmheadingEnvironment[2][]{% }

Parse the options. First set defaults, and then parse the input string.

\begin{verbatim}
def\cmdKV@phfthmmkthmheading@thmstyle{plain}\
def\cmdKV@phfthmmkthmheading@internalcounter{phfthmheadingcounter}\
\setkeys{phfthmmkthmheading}{#1}\
\end{verbatim}

And now, produce the relevant definitions:

\begin{verbatim}
csdef{phfthm@thmheading@#2@val@title}{$\langle$No Title Given$\rangle$}\
\theoremstyle{\cmdKV@phfthmmkthmheading@thmstyle}\
\newtheorem*{phfthm@internal@thmheading@#2}{% 
\csname phfthm@thmheading@#2@val@title\endcsname}\
\end{verbatim}

We use \newtheorem* to create an unnumbered theorem. The fixed title is just a single token, the macro which will be set to the relevant title at the last moment.

\begin{verbatim}
\newenvironment{#2}[1][]{\csdef{phfthm@thmheading@#2@val@title}{##1}\letcs\thephfthmheadingcounter{phfthm@thmheading@#2@val@title}\
\csname phfthm@hook@thmheading@#2@start\endcsname{##1}\
\end{verbatim}

Define the actual environment.

\begin{verbatim}
\csname phfthm@internal@thmheading@#2\endcsname\
\hspace*{0pt}\refstepcounter{\cmdKV@phfthmmkthmheading@internalcounter}\
\csname phfthm@hook@thmheading@#2@start\endcsname{##1}\
\end{verbatim}

Relay call to the internal AMS-defined “theorem:”

\begin{verbatim}
\csname phfthm@internal@thmheading@#2\endcsname\
\hspace*{0pt}\refstepcounter{\cmdKV@phfthmmkthmheading@internalcounter}\
\csname phfthm@hook@thmheading@#2@start\endcsname{##1}\
\end{verbatim}

Pin down an anchor. The use of \hspace*{0pt} is explained at http://tex.stackexchange.com/a/88493/32188 (see especially the first comment).

\begin{verbatim}
\hspace*{0pt}\refstepcounter{\cmdKV@phfthmmkthmheading@internalcounter}\
\csname phfthm@hook@thmheading@#2@start\endcsname{##1}\
\end{verbatim}

Also, let’s add some flexibility in the hspace:

\begin{verbatim}
\hskip 0em plus 0.5em minus 0em\% 
\ignorespaces\% 
\end{verbatim}

Now, the END part of the environment: just call the callback and close the internal AMS-defined theorem.
Also define the relevant callbacks, which just relay their calls to the default callbacks.

\csdef{phfthm@hook@thmheading@#2@start}##1{\phfthm@hook@thmheading@start{##1}}
\csdef{phfthm@hook@thmheading@#2@end}{\phfthm@hook@thmheading@end}

Provide as well the obsolete command \phfthmMakeThmheadingEnvironment which was provided in earlier versions of this package:

\def\phfthmMakeThmheadingEnvironment{\phfMakeThmheadingEnvironment}

\begin{verbatim}
\phfthm@hook@thmheading@start
\phfthm@hook@thmheading@end
\end{verbatim}

Global callbacks which are called for all thmheading-type environments defined with \phfMakeThmheadingEnvironment (unless their hooks have been changed in order for them not to call these global hooks).

\def\phfthm@hook@thmheading@start#1{}
\def\phfthm@hook@thmheading@end{}

### 7.6 Theorem sets

Here, we define the theorem sets proposed by the package for quick loading.

We first define the names. These are defined in any case regardless of whether we are loading a theorem set or of which theorem set we are loading.

\def\theoremname{Theorem}
\def\propositionname{Proposition}
\def\lemmaname{Lemma}
\def\corollaryname{Corollary}
\def\conjecturename{Conjecture}
\def\remarkname{Remark}
\def\definitionname{Definition}
\def\ideaname{Idea}
\def\questionname{Question}
\def\claimname{Claim}
\def\observationname{Observation}
\def\problemname{Problem}

As we define the theorem sets, remember the names in a comma-separated list which we can display in help text. The \phfthm@def@thmset replaces the \def command and expects the definitions to follow immediately.

\def\phfthm@def@thmset@optlist{}
In definitions of theorem sets, use these macros to define a new theorem-like environment (theorem, proposition, corollary, etc.) or definition-like environment (definition, remark). The macros \phfthm@val@mkthmoptarg@theorem and \phfthm@val@mkthmoptarg@defn are defined by \phfLoadThmSet.

The default set (empty name, or name “empty”) provides no theorem. (The first line uses \def directly so that we don’t include an empty item in the list of available choices.)

Theorem set simple:

Theorem set default:

Theorem set shortnames:
The macro \phfLoadThmSet loads a theorem set. See documentation at subsection 3.2.

#1 = options to \phfMakeTheorem for theorem-like environments
#2 = options to \phfMakeTheorem for definition-like environments
#3 = name of the theorem set to load

\newcommand\phfLoadThmSet[3]{%
  \ifcsname phfthm@thmset@#3\endcsname%
  \edef\phfthm@val@mkthmoptarg@theorem{#1}%
  \edef\phfthm@val@mkthmoptarg@defn{#2}%
  \csname phfthm@thmset@#3\endcsname%
  \else%
    \PackageWarning{phfthm}{Unknown theorem set: ‘#3’!}%
  \fi%
}

For compatibility with my earlier versions of phfthm, also provide the obsolete
\phfthmLoadThmSet:

\def\phfthmLoadThmSet{\phfLoadThmSet}

\section{Package option handling}

The machinery is in place, now define and parse the package options.

\subsection{Declaring the package options}

The package options all use the keyval parsing mechanism using the xkeyval package.

Recall when using \define@XXXkey that the optional argument after the second mandatory argument is the value which is assumed if the key is given with no explicit value; it is not the initial default value.
The \texttt{resetstyle} package option

An option to reset all options so that the package provides only stand-alone definitions and is not invasive (see section 2).

This option does not expect any argument (i.e., you should specify \texttt{usepackage[resetstyle,\{other options\}]{phfthm}}, and not \texttt{usepackage[resetstyle=true,\{other options\}]{phfthm}}).

\begin{verbatim}
545 \define@key{phfthmpkg}{resetstyle}[]\{%
546 \KV@phfthmpkg@smallproofsfalse\%
547 \KV@phfthmpkg@qedsymbolblacksquarefalse\%
548 \KV@phfthmpkg@prooftitleitbffalse\%
549 \KV@phfthmpkg@sepcountersfalse\%
550 \KV@phfthmpkg@countpersectionfalse\%
551 \KV@phfthmpkg@proofterrfalse\%
552 \if\relax\detokenize{#1}\relax\else\%
553 \PackageError{phfthm}{'resetstyle' does not take any argument.}{You
554 specified the 'resetstyle' argument and provided a value to it
555 ('resetstyle=...'). However the 'resetstyle' option does not accept
556 any value argument.}
557 \fi\%
558 \}
\end{verbatim}

Options for loading theorem sets

Define the various package options for the loading of predefined theorem sets (subsection 2.1).

The \texttt{sepcounters} option, off by default.

\begin{verbatim}
559 \define@boolkey{phfthmpkg}{sepcounters}[true]{}
560 \KV@phfthmpkg@sepcountersfalse
\end{verbatim}

The \texttt{countpersection} option, off by default.

\begin{verbatim}
561 \define@boolkey{phfthmpkg}{countpersection}[true]{}
562 \KV@phfthmpkg@countpersectionfalse
\end{verbatim}

The \texttt{proofref} option. The proof-ref is on initially by default.

\begin{verbatim}
563 \newif\ifKV@phfthmpkg@proofref
564 \KV@phfthmpkg@proofreftrue
565 \def\cmdKV@phfthmpkg@proofreff@style{default}
\end{verbatim}

Actually define the option itself. Here we do some customized parsing of the value of the \texttt{proofref=...} option, to treat the cases \texttt{proofref=} (empty argument) and \texttt{proofref=false} separately.

\begin{verbatim}
566 \define@key{phfthmpkg}{proofref}[]\{%
567 \ifblank{#1}\{%
\end{verbatim}

44
If a blank argument provided, set some sensible defaults with proofref on:

\ifstrequal{#1}{false}{KV@phfthmpkg@proofreffalse}{KV@phfthmpkg@proofreftrue
\def\cmdKV@phfthmpkg@proofref@style{#1}}%

Otherwise, check to see if the value is false, in which case deactivate the proof-ref mechanism, or else, activate it and set the given style value as documented in subsection 2.1.

\ifstrequal{#1}{false}{KV@phfthmpkg@proofreffalse}{KV@phfthmpkg@proofreftrue
\def\cmdKV@phfthmpkg@proofref@style{#1}}%

The \textbf{thmset} option. We subtly construct the command \define@choicekey{phfthmpkg}{thmset}{\val}{\phfthm@def@thmset@optlist}, but with the last macro (option list) expanded.

\edef\@tmpb{{\phfthm@def@thmset@optlist}}\expandafter\@tmpa\@tmpb{\xdef\cmdKV@phfthmpkg@thmset{\val}}%

By default we should load the default set.

\def\cmdKV@phfthmpkg@thmset{default}

The options \textbf{theoremstyle} and \textbf{definitionstyle} set which theorem style to use for theorems and definitions, when loading the given thmset.

\define@cmdkey{phfthmpkg}{theoremstyle}{}\define@cmdkey{phfthmpkg}{definitionstyle}{}

Proof environment options

Define the package options \textbf{proofenv}, \textbf{smallproofs}, \textbf{qedsymbolblacksquare} and \textbf{prooftitleitbf} (subsection 2.2).

\define@boolkey{phfthmpkg}{proofenv}{true}\define@boolkey{phfthmpkg}{smallproofs}{true}
\define@boolkey{phfthmpkg}{qedsymbolblacksquare}{true}\define@boolkey{phfthmpkg}{prooftitleitbf}{true}
Set the initial default values for these options.

593 \KV@phfthmpkg@smallproofstrue
594 \KV@phfthmpkg@qedsymbolblacksquaretrue
595 \KV@phfthmpkg@proofenvtrue
596 \KV@phfthmpkg@prooftitleitbfalse

Options for a theorem-like heading environment

Define the \textbf{thmheading} and \textbf{thmheadingstyle} package options, documented in \textit{section 6}.

597 \define@boolkey{phfthmpkg}{thmheading}[true]{}
598 \define@cmdkey{phfthmpkg}{thmheadingstyle}{}

The \textit{thmheading} environment is provided by default; it's a stand-alone definition anyway. The style defaults to the \textit{plain} style.

599 \KV@phfthmpkg@thmheadingtrue
600 \def\cmdKV@phfthmpkg@thmheadingstyle{plain}

7.7.2 Parsing the package options

The usual stuff (\texttt{xkeyval}-flavored).

601 \DeclareOptionX*{%
602 \PackageWarning{phfthm}{Invalid option: \texttt{\CurrentOption}}%
603 }%\ProcessOptionsX<phfthmpkg>

7.7.3 Execute package options-controlled actions

\textit{Loading a theorem set}

First, we need to take into account the options which alter the way the theorem sets will be loaded (separate counters, proof-ref, etc.).

Take care of the proof-ref stuff. First, define the possible styles (note that these are not the same as the values to the \texttt{proofrefstyle} argument to the \texttt{\phfMakeTheorem} command).

605 \def\phfthm@val@mkthmopt@proofrefstyle{}
606 \ifKV@phfthmpkg@proofref
607 \def\phfthm@proofref@style@default{}
608 \def\phfthm@proofref@style{}%

Note that \texttt{proofref=always} and \texttt{proofref=onlyifveryfar} have a global effect, because they set \texttt{\phfProofrefPageBackTolerance} and \texttt{\phfProofrefPageAheadTolerance} (see documentation in \textit{subsection 2.1}).
\def\phfthm@proofref@style@always{
  \def\phfProofrefPageBackTolerance{-1}
  \def\phfProofrefPageAheadTolerance{-1}
}\def\phfProofref@style@onlyifveryfar{
  \def\phfProofrefPageBackTolerance{2}
  \def\phfProofrefPageAheadTolerance{4}
}\def\phfProofref@style@margin{
  \def\phfthm@val@mkthmopt@proofrefstyle{proofrefstyle=margin}
}\def\phfProofref@style@marginbottom{
  \def\phfthm@val@mkthmopt@proofrefstyle{proofrefstyle=marginbottom}
}\def\phfProofref@style@longref{
  \par{\raggedleft\proofrefsize{\proofonname}}\par}
\def\x{\par{\raggedleft\proofrefsize{\proofonname}}\par}
\def\phfProofref@style@off{
  \def\phfthm@val@mkthmopt@proofrefstyle{proofref=false}
}\def\proofonname##1##2{The proof of this \phfthm@autorefnameof{##1} can
  be found on ##2.}
\def\proofrefsize##2{\proofrefsize{##2}}
Now execute the given styles. Construct the command
\phfthm@internal@execattribs{phfthm@proofref@style@}{ProofRef Style}{\cmdKV@phfthmpkg@proofref@style},
but with the last macro expanded.
\def\x{\par{\raggedleft\proofrefsize}}
\def\phfProofref@style@off{\def\phfthm@val@mkthmopt@proofrefstyle{proofref=false}}
Prepare the proof-ref option in case we don't want any proof reference mechanism on.

Changed in v1.1 [2019/03/12]: Bug fix: package option proofref=false didn't have any effect.
\fi
Take care of counters. In any case, define a common counter, in case we use a
common counter for all theorem types. (The counter is defined in any case, to
avoid breaking other code which might use it if suddenly the user decides to use `sepcounters=true` for their document.)

\newcounter{phfthmcounter}
\setcounter{phfthmcounter}{0}

Prepare an argument to \texttt{\phfMakeTheorem} according to the `countpersection` and `sepcounters` options. First determine the parent counter in case `countpersection` is set (chapter if chapter counter exists, else section).

\ifKV@phfthmpkg@countpersection
  \edef\phfthm@val@mkthmopt@counteropts{parentcounter=\phfthm@tmp@parentcountername}
\else
  \def\phfthm@val@mkthmopt@counteropts{counter=phfthmcounter}
\fi
\ifKV@phfthmpkg@sepcounters
  \edef\phfthm@val@mkthmopt@counteropts{\phfthm@val@mkthmopt@counteropts,\phfthm@val@mkthmopt@proofrefstyle,thmstyle=\cmdKV@phfthmpkg@theoremstyle}
\else
  \edef\phfthm@val@mkthmopt@counteropts{\phfthm@val@mkthmopt@counteropts,\phfthm@val@mkthmopt@proofrefstyle,thmstyle=\cmdKV@phfthmpkg@definitionstyle}
\fi

Finally, load the theorem set defined by the options. The first argument regroups the options for theorem environments (Theorem, Proposition, Corollary, ...); the second argument regroups the options for definition environments (Definition); the third argument is the theorem set name itself.

\phfLoadThmSet%{(\phfthm@val@mkthmopt@counteropts,phfthm@val@mkthmopt@proofrefstyle,thmstyle=\cmdKV@phfthmpkg@theoremstyle)%}\phfLoadThmSet%{(\phfthm@val@mkthmopt@counteropts,\phfthm@val@mkthmopt@proofrefstyle,\phfthm@val@mkthmopt@counteropts,\phfthm@val@mkthmopt@proofref=false,thmstyle=\cmdKV@phfthmpkg@definitionstyle)%}\phfLoadThmSet%{\cmdKV@phfthmpkg@thmset}

(Note the absence of the proof-ref for definitions.)

\textit{Define the proof environment}

If requested, define the proof environment (subsection 2.2). First, make sure we take into account the options `smallproofs`, `qedsymbolblacksquare` and
Go ahead and define the proof environment. Because we have already loaded amsthm, we need to override the existing proof environment.

\ifKV@phfthmpkg@proofenv
\phfMakeProofEnv[override=true]{proof}
\fi

Define the theorem-heading environment

Define the thmheading environment, if requested.

\ifKV@phfthmpkg@thmheading
\phfMakeThmheadingEnvironment[%
[thmstyle=\cmdKV@phfthmpkg@thmheadingstyle]{thmheading}]
\fi
Change History

v1.0
General: Initial version .................................................. 1

v1.1
General: Bug fix: package option proofref=false didn’t have any effect .... 47

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.

Symbols

* (cmd. opt.) ........................................... 12
\@car ........................................... 351
\@endpefalse ........................................... 266
\@for ........................................... 7
\@gobble ................................. 352, 396, 416
\@ifnextchar ........................................... 278, 281
\@ifpackageloaded .................................. 301
\@nil ........................................... 329, 332, 335, 338, 353
\@plus ........................................... 259
\@tmpa ........................................... 579, 581
\@tmpb ........................................... 580, 581
\\ ........................................... 333
{} (pkg. opt.) ........................................... 5

A
aliascnt ........................................... 8, 20
\aliascntresetthe .................................. 50
aliascounter (cmd. opt.) ........... 8
always (pkg. opt.) ................. 5, 6
amsmath ........................................... 20
amsthm ........................................... 2, 4, 6, 7, 10, 12, 20, 25, 49
\appto ........................................... 495
\apptocmd ........................................... 673
\AtBeginDocument .................................. 300
\autopageref ......................... 381, 385, 387
\autoref ........................................... 301

B
babel ........................................... 10, 13
\baselineskip ........................................... 411
\baselinestretch ........................................... 674
\begin ........................................... 57, 92, 204
\begingroup ........................................... 58, 93
\bfseries ........................................... 685
\blacksquare ........................................... 680
\box ........................................... 443, 445

C
claim (environment) ......................... 4
\claimname ........................................... 10, 490, 531
\cmdKV@phfmkprf@defaultproofname ......... 150, 161, 167, 170
\cmdKV@phfmkprf@displayenv ................. 149, 157, 158, 160, 199
\cmdKV@phfmkprf@internalcounter
\cmdKV@phfmkprf@parselabelcmd ............... 151, 198
\cmdKV@phfmkprf@proofofname ................. 154, 174
\cmdKV@phfmkthm@counter ......................... 152, 188
\cmdKV@phfmkthm@parentcounter ................. 25, 38, 48, 53
\cmdKV@phfmkthm@proofofstyle ................. 30, 39, 42
\cmdKV@phfmkprf@proofrefstyle ................. 31, 68
<table>
<thead>
<tr>
<th>Command</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\qed</td>
<td>258</td>
</tr>
<tr>
<td>\qedsymbol</td>
<td>681</td>
</tr>
<tr>
<td>\qedsymbolblacksquare</td>
<td>6, 45, 49</td>
</tr>
<tr>
<td>\remark</td>
<td>4</td>
</tr>
<tr>
<td>\raggedleft</td>
<td>405, 437, 627</td>
</tr>
<tr>
<td>\raggedright</td>
<td>407, 440</td>
</tr>
<tr>
<td>\rallge</td>
<td>457</td>
</tr>
<tr>
<td>\ref</td>
<td>299</td>
</tr>
<tr>
<td>\refstepcounter</td>
<td>198, 372, 465</td>
</tr>
<tr>
<td>\relax</td>
<td>33, 38</td>
</tr>
<tr>
<td>\setcounter</td>
<td>55</td>
</tr>
<tr>
<td>Command</td>
<td>Pages</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>\remarkname</td>
<td>10, 486, 515, 524</td>
</tr>
<tr>
<td>\renewcommand</td>
<td>681</td>
</tr>
<tr>
<td>\renewenvironment</td>
<td>209</td>
</tr>
<tr>
<td>\RequirePackage</td>
<td>1, 2, 3, 4, 5, 679</td>
</tr>
<tr>
<td>\resetstyle</td>
<td>3, 44</td>
</tr>
<tr>
<td>\rich (pkg. opt.)</td>
<td>4</td>
</tr>
<tr>
<td>\rule</td>
<td>411</td>
</tr>
<tr>
<td>\setbox</td>
<td>436, 439</td>
</tr>
<tr>
<td>\setcounter</td>
<td>640</td>
</tr>
<tr>
<td>\setkeys</td>
<td>32, 155, 456</td>
</tr>
<tr>
<td>\shortnames</td>
<td>4</td>
</tr>
<tr>
<td>\simple (pkg. opt.)</td>
<td>4</td>
</tr>
<tr>
<td>\sloppy</td>
<td>437, 440</td>
</tr>
<tr>
<td>\space</td>
<td>6, 45, 48</td>
</tr>
<tr>
<td>\string</td>
<td>119, 675</td>
</tr>
<tr>
<td>\strut</td>
<td>438, 441</td>
</tr>
<tr>
<td>\text</td>
<td>681</td>
</tr>
<tr>
<td>\textbf</td>
<td>326</td>
</tr>
<tr>
<td>\the</td>
<td>378</td>
</tr>
<tr>
<td>\theorem (environment)</td>
<td>4</td>
</tr>
<tr>
<td>\theoremmname</td>
<td>10, 481, 503, 510, 519</td>
</tr>
<tr>
<td>\theoremmstyle</td>
<td>35, 458</td>
</tr>
<tr>
<td>\theoremmstyle (pkg. opt.)</td>
<td>4, 45</td>
</tr>
<tr>
<td>\thefhfhmheadingcounter</td>
<td>463</td>
</tr>
<tr>
<td>\thefhfhmInternalProofrefCounter</td>
<td>373, 376</td>
</tr>
<tr>
<td>\thmheading</td>
<td>7, 46</td>
</tr>
<tr>
<td>\thmheading (environment)</td>
<td>7</td>
</tr>
<tr>
<td>\thmheadingstyle (pkg. opt.)</td>
<td>7, 46</td>
</tr>
<tr>
<td>\thmset</td>
<td>3–6, 45</td>
</tr>
<tr>
<td>\thmstyle (cmd. opt.)</td>
<td>8, 19</td>
</tr>
<tr>
<td>\tiny</td>
<td>681</td>
</tr>
<tr>
<td>\topsep</td>
<td>259</td>
</tr>
<tr>
<td>\trivlist</td>
<td>260</td>
</tr>
<tr>
<td>\true (cmd. opt.)</td>
<td>8, 11, 18</td>
</tr>
<tr>
<td>\true (pkg. opt.)</td>
<td>6, 7</td>
</tr>
<tr>
<td>\val</td>
<td>579, 582</td>
</tr>
<tr>
<td>\vspace</td>
<td>411</td>
</tr>
<tr>
<td>\x</td>
<td>40, 42, 44, 233, 234, 633, 635</td>
</tr>
<tr>
<td>\xkeyval</td>
<td>43, 46</td>
</tr>
<tr>
<td>\z@</td>
<td>411</td>
</tr>
</tbody>
</table>