1 Introduction

Euclid established a style in his *Elements* that, for better or worse, has come down to us as the definitive one for presenting mathematical proofs. A *Theorem* or *Problem* (that is, something to be *proved* or *constructed*, respectively) is first stated, with the heading “Theorem” or “Problem”. After the proof or construction has been given, the original statement is re-iterated, followed by the letters “QED” (or “QEI” for a Problem).

QED stood for either *quod erat demonstrandum* or *quod est demonstratum*, which respectively mean “which was to be shown” and “which has been shown”. Modern custom is to use a little square instead of the letters QED.

It is also customary to put this square on the far right of the page, which is where this macro package comes in. It is surprisingly difficult to persuade TeX to do this. The obvious solution,

\[
\text{and so } 2n = p + q \text{ as required. } \hfill \square
\]

works 80% of the time, but fails if the preceding text happens to fill the line, in which case the square appears at the left of the next line.

This happens because, when TeX breaks the line at \hfill, it discards this stretchable space (called “glue”) [Texbook, page 97]. The reason for this behaviour is that, when a word occurs at the beginning or end of a line, we like to have it printed hard against the margin, so the space that would otherwise occur between words is lost when the line is broken there.

TeX and \LaTeX have been in standard use in the mathematical sciences for two decades now, so you would expect this problem to have been solved, and solved many times. However, on examining the documents and macro files that have come my way, I find that they either employ the naïve solution above, which doesn’t work, or they use code that is mine.

With the help of Chris Thompson of the Cambridge University Computing Service, I implemented the following solution for my Ph.D. thesis in 1986. As well as keeping the square on the right (though possibly on the next line), it also stops it from being left over (as a “widow”) on the next page.

\[
\text{\def\qed}{%}
\parfillskip=0pt \% so \par doesnt push \square to left
\]
If you already have a macro called \texttt{\textbackslash qed}, \texttt{\textbackslash endproof} or \texttt{\textbackslash pushright} which is laid out in a similar way to this, then it is almost certainly my code. Please use this macro package in place of this fragment of my code. Besides the basic case of the little square at the end of a piece of text, it also works with displays ($$ or the \texttt{displaymath} environment) and has hooks for compatibility with other situations.

\section{QED as a macro}

The basic use of the package is as follows:
\begin{verbatim}
\documentclass{article}
\title{A Proof of Goldbach's Conjecture}
\author{Paul Taylor}
\usepackage[open-square,define-standard-theorems]{QED}
\usepackage{amssymb}
\begin{document}
\maketitle
\begin{Theorem}
Every even number may be expressed as a sum of two primes.
\end{Theorem}
\Proof Trivial. $2n = p+q$. \qed
\end{verbatim}
In principle, the \texttt{\textbackslash qed} macro goes logically where you want the little square to appear in the text. \textbf{But this still doesn’t work} (and nor can it) because of the way \LaTeX{} works. Briefly, the \texttt{\textbackslash qed} macro alters the way in which the paragraph is formatted, but commands like \texttt{\textbackslash end\{itemize\}} close the paragraph themselves, leaving \texttt{\textbackslash qed} to the next paragraph, and therefore the next line.

The rule is that \texttt{\textbackslash qed} has to be put \emph{inside} the innermost environment instead, for example:

\begin{verbatim}
\begin{itemize}
  \item This case is easy too.
  \item In the final case we have
    \begin{equation}
      2n = p + q \quad \texttt{\textbackslash qed}
    \end{equation}
\end{itemize}
\end{verbatim}

If you still have to put \texttt{\textbackslash qed} outside the environment, it may be appropriate to add \texttt{\vskip\baselineskip} before \texttt{\textbackslash qed}, unless of course this would over-print. There is, however, still a danger that \LaTeX{} will break the page here (which seems to be unavoidable if the environment ends with a \texttt{\textbackslash penalty}).

Even if you don’t need to put \texttt{\textbackslash qed} inside an inner environment to avoid a blank line, it’s still a good idea to do so, because otherwise it’s possible that intervening end-of-environment material may allow \LaTeX{} to put the square on the next page.

\section{Options}

The package may be customised using the following options in the \texttt{\textbackslash usepackage} declaration:

\begin{itemize}
  \item \texttt{\textbackslash black-square} \quad Use ■ for \texttt{\textbackslash qed}.
  \item \texttt{\textbackslash open-square} \quad Use □ for \texttt{\textbackslash qed}.
  \item \texttt{\textbackslash define-standard-theorems} \quad Define Answer, Axiom, Assumption, Conjecture, Construction, Convention, Corollary, Counterexample, Definition, Example, Exercise, Fact, Lemma, NB, Notation, Note, Observation, Problem, Proposition, PS, Remark, Question, Theorem, Thesis, Warning and their plurals.
  \item \texttt{\textbackslash roman-theorems} \quad Print the theorem statements in roman (upright).
  \item \texttt{\textbackslash italic-theorems} \quad Print the theorem statements in \emph{italic} (default).
  \item \texttt{\textbackslash theorems-as-commands} \quad Use \emph{commands} like \texttt{\textbackslash Theorem} to introduce a theorem state-
ment, instead of wrapping it in a \begin{Theorem} ... \end{Theorem} environment.

\[\text{auto-QED}\]
Make \end{Theorem} insert a \texttt{\textbackslash \textbf{\texttt{\textbackslash qed}}} automatically.

\[\text{explicit-QED}\]
Suppress automatic \texttt{\textbackslash qed} in \end{Theorem} (default).

\[\text{number-theorems}\]
(Default).

\[\text{dont-number-theorems}\]
Beware that the □ and ■ symbols come from the AMS symbols fonts, not the standard \LaTeX ones. This is why you need to \texttt{\textbackslash \usepackage{amssymb}}.

Besides the open-square and black-square options, you can also change what \texttt{\textbackslash qed} prints by redefining, for example,
\begin{verbatim}
\def\qedsymbol\c{\texttt{\textbackslash \textbf{\texttt{\textbackslash \bigcirc}}}}
\end{verbatim}
to get a big circle. The macros \texttt{\textbackslash QED}, \texttt{\textbackslash QEI} and \texttt{\textbackslash QEF} are also provided for classically minded mathematicians, and produce these three-letter acronyms (I and F stand for inveniendum and faciendum — to be found or done).

The \texttt{\textbackslash \textbf{\texttt{\textbackslash Proof}}} macro, of course, produces the word “Proof”, in bold, as a new un

\[\text{indented paragraph. You can change this by redefining, for example,}
\begin{verbatim}
\def \TheWordProof \{\kern\parindent\textsc{Démonstration}: }
\end{verbatim}
\[\text{to print it in SMALL CAPITALS with indentation and a colon.}

4 Proof environments

Given that a proof is enclosed in the pair \texttt{\textbackslash \textbf{\texttt{\textbackslash Proof}} ... \texttt{\textbackslash qed}}, you may like to use a \LaTeX environment instead:
\begin{verbatim}
\begin{Theorem}
\end{Theorem}
\begin{Proof}
\text{Easy.}
\end{Proof}
\end{verbatim}
The problem with this is that \texttt{\textbackslash \textbf{\texttt{\textbackslash end{Proof}}} comes after any \texttt{\textbackslash itemize} that might finish the proof, which is too late for \texttt{\textbackslash qed} to do its stuff.

In this case, you have to insert \texttt{\textbackslash qed} yourself within \texttt{\textbackslash \textbf{\texttt{\textbackslash itemize}}} as in Section 2. It will then print in the correct place. \texttt{QED.sty} will not repeat the square when \texttt{\textbackslash \textbf{\texttt{\textbackslash end{Proof}}} is executed.

In order to achieve this effect, and also to allow for nested Proof environments (for example the Lemmas leading up to a Theorem), \texttt{QED.sty} has to maintain an internal switch saying whether to print the square or not. This switch is turned on by \texttt{\textbackslash \textbf{\texttt{\textbackslash Proof}}} or \texttt{\begin{Proof}} and off when \texttt{\textbackslash qed} is executed. Nested Proof environments save the outer value of the switch. As a result of this,
\texttt{\textbackslash qed which doesn’t follow \textbackslash \textbf{\texttt{\textbackslash Proof}}} will not print the square!
Each properly nested \begin{Proof} \ldots \end{Proof} pair gets a single square at the end, on the far right hand side.

## 5 Theorem styles and environments

You may feel that it is logical to put the \texttt{Proof} environment \textit{within} the \texttt{Theorem} environment. \texttt{QED.sty} supports this. In the case where the \texttt{Proof} is absent (because it is obvious or because the main Theorem is stated after its proof), you can arrange for the square to be generated by \texttt{\end{Theorem}}, by selecting the \texttt{auto-QED} option in \texttt{\usepackage}.

I, on the other hand, regard the words \texttt{Theorem \textit{etc. as headings}}, and prefer not to wrap my enunciations in \LaTeX environments at all. I just write

\begin{verbatim}
\Theorem $2n = p+q$. 
\Proof Obvious. \qed
\end{verbatim}

This behaviour is declared with the \texttt{theorems-as-commands} option in \texttt{\usepackage}. This also suppresses \textit{italic} typesetting of the statement, because there is no \texttt{\end{Theorem}} to turn it off.

You can also suppress italicisation or numbering with the \texttt{\usepackage} options \texttt{roman-theorems} and \texttt{dont-number-theorems}.

If you want to use Mittelbach’s \texttt{theorem.sty} package, you have to load that first, then \texttt{QED.sty}. This is because of the way in which the two packages interact, in particular regarding the internal macro \texttt{\@thm}. The options \texttt{theorems-as-commands}, \texttt{roman-theorems} and \texttt{dont-number-theorems} will not work in this case.

## 6 Control of page breaking

This is not really concerned with theorems and proofs, but this problem seems to arise in the same context.

Whilst \LaTeX is extremely good at breaking paragraphs into lines nicely, it often makes a mess of page breaking. This is because the program was written in the early 1980s, when computers had kilobytes rather than megabytes of memory, so \LaTeX generates pages of DVI as soon as it can.

Its mechanisms for breaking both lines and pages involve \texttt{\penalty} items in its horizontal and vertical lists. The \LaTeX commands \texttt{\nobreak} and \texttt{\goodbreak} generate \LaTeX \texttt{\penalty} commands, which are supposed to control (line and) page breaking.

In my experience, these don’t work.

The macros \texttt{\goalbreak} and \texttt{\allowlines} fix this problem by testing how full the page already is, and generating a \texttt{\newpage} if necessary.

- \texttt{\goalbreak{50pt}} or \texttt{\goalbreak{2cm}} ensure that this much space is available on the current page.

- \texttt{\allowlines{5}} measures the space in lines instead (using \texttt{\baselineskip}).
You may want to use this just before a short Theorem statement in order to keep it on the same page as its proof. (In fact, QED.sty uses a small \penalty to encourage this anyway.)

The macro \closeupaline is useful after \Proof when displayed material follows immediately:
\Proof\closeupaline
\ $$ 2n = p + q $$

7 Modifying other \La\TeX\ environments

This section is for \La\TeX\ experts.

The \qed macro will only work in those environments which provide access to the right hand side of the page. In other words, wherever you think “\hfill\square” ought to work then \qed will. (Of course it is just because that naive method doesn’t work that this macro file is needed at all.) The macro was designed to solve the problem in the case of “unrestricted horizontal mode” (i.e. at the end of a paragraph); it also works in $$ displays by pretending to be an equation number. In most other cases it basically does “\hfill\square” or “\hfill$\square$” and hopes for the best.

Unfortunately some environments (we shall use picture as an example) do not allow this at all. In this case, the (re)designers of the environment may wish to handle the placing of the little square for themselves.

Each individual environment (both the standard ones and those defined by the user) needs to co-operate in order to make \qed work. It is not the business of this package to rewrite \La\TeX\ in this way, but the following mechanism may be inserted into other environments.

- Put \WillHandleQED in the \begin part of the environment definition. This notifies QED.sty that the environment is going to ...
- put \HandleQED anywhere in the \end part of the environment definition (even inside dollars, braces or another environment), where you would like the square to be printed if \qed is used.

For example:
\newenvironment{qedpicture}\
{\begin{picture}\WillHandleQED}\
{\put(0,0){\HandleQED}\end{picture}}

defines a version of the picture environment in which the \qed command causes a square to be printed at the origin of picture’s coordinate system — wherever \qed occurs within the body.

The width of the saved \qed symbol is available for testing as the dimension \wd\QEDbox.

To make this compatible with not loading QED.sty, add the following to macro files which use \WillHandleQED and \HandleQED:
\% ifx\WillHandleQED\undefined
Also, \UnHandleQED restores normal processing; this may be useful for the minipage environment, which has its own notion of “right hand side” of the page.

8 Implementation

Underlying \qed with its switchable behaviour and configurable symbol is a macro \pushright{text} that always prints its argument, on the far right. The argument to \pushright is in text mode, so \qed invokes it with \{${\text{\qedsymbol}}$}. It does the following case analysis:

- outer horizontal text mode (in a paragraph) uses \parag@pushright{#1}
- inner horizontal (text or maths) mode (\hbox) uses \hbox@pushright{#1}
- outer maths mode (display) uses \TeX’s \eqno mechanism but has to suppress a \LaTeX error check.
- outer vertical mode (between paragraphs on the current page) uses \page@pushright{#1};
- inner vertical mode (\vbox) uses \vbox@pushright{#1};

The interested reader can follow the comments in the file QED.sty itself to see how these cases are handled.

Why can’t we do all of this with \endProof? Because by that stage the damage has been done: the environments, paragraphs or displays have already been closed and their \hbox’es added to the current vertical list, and cannot be removed for alteration if this is the current page.

Ideally we should be able to add the symbol to the last line after the paragraph, display or whatever other structure has been completed and we’ve returned to vertical mode.

The reason why this isn’t possible is that \TeX forbids the use of \lastbox and similar commands (which can be used to take \hbox and \vbox lists apart) on the current page (outer vmode). It does this because its “galley” implementation (shipping complete pages to the DVI file as soon as possible) might output the item before we have a chance to remove it from the page.

In principle there is a more sophisticated method, which could be used in inner vertical mode (i.e. inside \TeX’s \vbox, \vadjust or \insert or \LaTeX’s \footnote, \parbox, minipage, figure or table). In these circumstances \lastbox is allowed.
We would have to strip and restore glue and penalties (\texttt{\textbackslash insert}, \texttt{\textbackslash mark} and \texttt{\textbackslash write} are obstacles to this) to get access to the last line (box), which may still not have the correct baseline. We could check its natural width against the page width, but if we re-set it, centering and (right) justification might be disturbed.

This method could even be used to implement the \texttt{Proof} environment by collecting the entire proof in its own \texttt{\textbackslash vbox}, modifying that and \texttt{\textbackslash unvbox}'ing it to the current page. This approach may lead to the following problems:

- footnotes, floats, marginpars, labels and index items may be lost;
- overflow of \TeX{}’s main memory, especially if \texttt{\textbackslash end\{Proof\}} is missing;
- I don’t know how page-breaking would be handled.

I don’t think this is worth the trouble. The package has, since January 1994, included an error message

\begin{verbatim}
QED.sty could be improved in this case: please ask
that appears when \texttt{\textbackslash qed} is used in inner vertical mode, but nobody has contacted me as a
result of it.
\end{verbatim}

9 Availability, compatibility, copyright and (no) warranty

\texttt{QED.sty} is Copyright © 1993-2003 Paul Taylor.

The package may be obtained from the Comprehensive \TeX{} Archive Network at
\url{www.ctan.org}
or one of its numerous mirror sites. Within the directory structure there, the file is called
\texttt{macros/generic/proofs/taylor/QED.sty}
but on your own computer you may call it \texttt{QED.sty}, \texttt{QED.tex}, \texttt{qed.sty} or \texttt{qed.tex}.

You may use this file for any purpose. You may copy it on condition that you do so in exactly the form in which it was supplied, make no charge for the copy and impose the same conditions on the recipient.

This software is supplied “as is” without warranty, express or implied, including but not limited to merchantability and fitness for any particular purpose. No liability is accepted by the author, his present or former employers or any third party for any direct or consequential damages, however caused. In all cases you must keep several backup versions of your files in a place to which your machine cannot write, and you must check documents visually before submitting the final version to a publisher or expensive printer.