L'extension thmbox*

Emmanuel Beffara
manu@beffara.org

24 avril 2005

Résumé

Cette extension définit l'environnement `thmbox` servant à présenter des théorèmes, définitions et objets similaires dans des boîtes décoratives avec des cadres et d'autres éléments esthétiques variés. La commande standard `\newtheorem` est redéfinie pour utiliser ce format.

1 Documentation

✖The package is loaded like any other, by writing

`\usepackage[`*options*`]{thmbox}`

The option `nothm` prevents the command `\newtheorem` from being redefined, so theorems defined with this command keep their traditional aspect. All other options are considered as default formatting options, they can be redefined at any time using the command `\thmboxoptions`. The argument of this macro is a list of `key=value` pairs in the `keyval` style, as defined in section 1.2.

1.1 Examples

The package defines an environment `thmbox` that is used as follows :

```
\begin{thmbox}[L]{The title}
  Some text, some more text, a
  sufficient amount to get a
  full box with several lines.
\end{thmbox}
```

The title

Some text, some more text, a sufficient amout to get a full box with several lines.

The argument `[L]` indicates the style of the box. The two other defined styles are `[M]` and `[S]`, which make respectively :

*Ce fichier a pour numéro de version 2.0 et a été mis à jour le 24/04/2005. Son titre original est « *thmbox package* ».

The title

Some text, some more text, a sufficient amout to get a full box with several lines.

The title

Some text, some more text, a sufficient amout to get a full box with several lines.

Any other option from the previous list can be used in the optional argument.

By default, loading the package `thmbox` replaces the definition of the LaTeX command `\newtheorem`. This feature can be turned off by the option `nothm`. The new version has the same syntax as the standard one with an extra optional argument at the beginning. This argument can be used to specify formatting options for the optional argument of the `thmbox` environment. For instance, saying

```
\newtheorem[L]{thm}{Theorem}[section]
```

will produce the following aspect for the environment `thm` :

```
\begin{thm}
  Any continuous function over
  $\mathbf{R}$ is measurable.
\end{thm}
```

Theorem 1.1

Any continuous function over $\mathbf{R}$ *is measurable.*

And with a title, with the defaut style, we get :

```
\begin{thm}[G\"odel] \label{tg}%
  Any theory that contains first
  order arithmetics is undecidable.
\end{thm}
```

Theorem 1.2 (*Gödel*)

Any theory that contains first order arithmetics is undecidable.

The default style for theorem boxes is "M", so writing

```
\newtheorem{cor}[thm]{Corollary}
```

will lead to this :

```
\begin{cor}
  Second order arithmetics
  is undecidable.
\end{cor}
```

Corollary 1.3

Second order arithmetics is undecidable.

The package also redefines the `proof` environment. The text inside such an enivronment is written smaller, with extra margins, with a black square sign at the end. The aspect is the following :

```
\begin{proof}
  This is a consequence of the
  inclusion of PA into AF2,
  since Peano arithmetics is
  undecidable.
\end{proof}
```

Démonstration : This is a consequence of the inclusion of PA into AF2, since Peano arithmetics is undecidable. ■

The `proof` environment takes an optional argument :

```
\begin{proof}[of \ref{tg}]
  This is a rather technical
  story of encodings.
\end{proof}
```

Démonstration of 1.2 : This is a rather technical story of encodings. ■

The `example` environment has mostly the same behaviour as `proof` :

```
\begin{example}
  An approach consists in
  encoding Turing machines.
\end{example}
```

Bla bla bla.

Example : An approach consists in encoding Turing machines.

Ploum plam.

Its optional argument can be used to write something else than "Example". The alternative method is to redefine `\examplename`.

```
\begin{example}[Idea]
  One could also proceed by
  encoding $\lambda$-calculus.
\end{example}
```

Idea : One could also proceed by encoding λ-calculus.

> As an extra, on the model of the `thmbox` environment, the package provides an environment `leftbar` that formats its contents with an extra margin and a running vertical rule in the left.

1.2 Options

The following general options are available :

`style=`⟨*letter*⟩ indicates which style should be used when drawing the boxes. The letter may be one of the following :

`S` : a vertical bar on the left of the text

`M` : a bar on the left and a short horizontal bar at the bottom of the text

`L` : a vertical bar on each side and a horizontal bar at the bottom

The default value for this parameter is `M`. The options `S`, `M`, `L` are shortcuts for `style=S`, `style=M` and `style=L`.

`cut=`⟨*bool*⟩ indicates if boxes may be cut at page breaks (true by default), `nocut` is equivalent to `cut=false`

The following options are used to change style of the header and contents :

`underline=`⟨*bool*⟩ indicates if the title of boxes should be underlined (true by default), `nounderline` is equivalent to `underline=false`

`headstyle=`⟨*text*⟩ defines how the header of the theorems is formatted. In the text, `#1` represents the environment name (i.e. "Theorem") and `#2` represent the number. The default value is "`\bfseries\boldmath#1 #2`".

`titlestyle=`⟨*text*⟩ defines how the optional title of theorems is formatted. In the text, `#1` represents the title. The default value is "` (\textit{#1})`".

`bodystyle=`⟨*text*⟩ defines how the text of theorems and similar environments is formatted, it is inserted before the text in those environments. The default value is "`\slshape\noindent`".

The following options define the various spacings :

`leftmargin=`⟨*dim*⟩, `rightmargin=`⟨*dim*⟩ defines the horizontal space between the margin of the surrounding text and that of the text inside the box (defaut value is `\parskip` for the left margin and 0pt for the right margin)

`hskip=`⟨*dim*⟩, `vskip=`⟨*dim*⟩ defines the horizontal and vertical space between the frame of the box and its contents (default value is 0.4em for both)

`thickness=`⟨*dim*⟩ defines the thickness of the bars (defaut value is 0.6pt)

1.3 Change log

Version 2.0 (2005/04/24)
- first CTAN release
- replaced `preskip` and `postskip` by `left`/`rightmargin` and `h`/`vskip`
- new option `headstyle`

Version 1.91 (2004/06/08)
- translated everything to english
- cleaned interface, switched to `keyval`, added parameters

Version 1.3 (2002/09/09)
- added option `nothm`

Version 1.2 (2002/05/24)
- added option `nocut`

Version 1.1 (2002/03/08)
- proper LaTeX interface with `\newtheorem`

Version 1.0 (around 2001)
- first version as an independent package

2 Implementation

First of all, we have to repect the protocol of LaTeX 2ε packages, by checking the format and introducing ourselves :

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{thmbox}[2005/04/24 v2.0 fancy theorem outlining]
```

2.1 Formatting options

`\thmboxoptions` Parameters are set using the `keyval` mechanism. All options are in the set named `thmbox`. The macro `\thmboxoptions` can be used to change the default parameters at any time.

```
\RequirePackage{keyval}
\newcommand\thmboxoptions{\setkeys{thmbox}}
```

`\thmbox@style` The style of the current box is stored by a `\let` in the macro `\thmbox@style`, it can be changed using the `style` option.

```
\let\thmbox@style=M
\define@key{thmbox}{style}{\let\thmbox@style=#1}
```

For compatibility with previous versions, we provide shortcuts for each style :

```
\define@key{thmbox}{S}[]{\let\thmbox@style=S}
\define@key{thmbox}{M}[]{\let\thmbox@style=M}
\define@key{thmbox}{L}[]{\let\thmbox@style=L}
```

`\thmbox@leftmargin` `\thmbox@rightmargin` `\thmbox@hskip` `\thmbox@vskip` `\thmbox@thickness` Next we have presentation parameters. The variables `\thmbox@leftmargin` and `\thmbox@rightmargin` are the extra margins to put between the surrounding text and the text of the box. The thickness of the frame is defined by `\thmbox@thickness`, its spacing with the contents is `\thmbox@hskip` horizontally and `\thmbox@vskip` vertically.

```
\newdimen\thmbox@leftmargin  \thmbox@leftmargin=\parindent
\newdimen\thmbox@rightmargin \thmbox@rightmargin=0pt
\newdimen\thmbox@hskip       \thmbox@hskip=.4em
\newdimen\thmbox@vskip       \thmbox@vskip=.4em
\newdimen\thmbox@thickness   \thmbox@thickness=.6pt
```

These parameters can be set using the keyval interface :

```
\define@key{thmbox}{leftmargin}{\thmbox@leftmargin=#1\relax}
\define@key{thmbox}{rightmargin}{\thmbox@rightmargin=#1\relax}
\define@key{thmbox}{hskip}{\thmbox@hskip=#1\relax}
\define@key{thmbox}{vskip}{\thmbox@vskip=#1\relax}
\define@key{thmbox}{thickness}{\thmbox@thickness=#1\relax}
```

`\ifthmbox@cut` The boolean `\ifthmbox@cut` indicates whether the boxes may be cut on page breaks or if they must be kept in one block. By default, cutting boxes is allowed.

```
\newif\ifthmbox@cut
\thmbox@cuttrue
```

This can be changed using the `cut` option, the option `nocut` is equivalent to `cut=false`.

```
\define@key{thmbox}{cut}[true]{%
  \expandafter\let\expandafter\ifthmbox@cut\csname if#1\endcsname}
\define@key{thmbox}{nocut}[]{\thmbox@cutfalse}
\DeclareOption{cut}{\thmbox@cuttrue}
\DeclareOption{nocut}{\thmbox@cutfalse}
```

`\ifthmbox@underline` The boolean `\ifthmbox@underline` indicates if the title of boxes should be underlined. It is activated by default and can be changed using the option `underline`.

```
\newif\ifthmbox@underline
\thmbox@underlinetrue
\define@key{thmbox}{underline}[true]{%
  \expandafter\let\expandafter\ifthmbox@underline\csname if#1\endcsname}
\define@key{thmbox}{nounderline}[]{\thmbox@underlinefalse}
\DeclareOption{underline}{\thmbox@underlinetrue}
\DeclareOption{nounderline}{\thmbox@underlinefalse}
```

`\thmbox@headstyle` The macro `\thmbox@headstyle` defines how the header of theorems is formatted, the arguments are the kind of enviromnent (i.e. "theorem") and its number.

```
\newcommand\thmbox@headstyle[2]{\bfseries\boldmath#1 #2}
```

This macro can be redefined using the option `headstyle` :

```
\define@key{thmbox}{headstyle}{\def\thmbox@headstyle##1##2{#1}}
```

`\thmbox@titlestyle` The macro `\thmbox@titlestyle` defines how the title of theorems is formatted, after the theorem number. Its argument is the text to be formatted.

```
\newcommand\thmbox@titlestyle[1]{ (\textit{#1\/})}
```

This macro can be redefined using the option `titlestyle` :

```
\define@key{thmbox}{titlestyle}{\def\thmbox@titlestyle##1{#1}}
```

`\thmbox@bodystyle` The macro `\thmbox@bodystyle` defines formatting options for the body of theorem-like environments. It is inserted before the text.

```
\newcommand\thmbox@bodystyle{\slshape\noindent}
```

This macro can be redefined using the option `bodystyle` :

```
\define@key{thmbox}{bodystyle}{\def\thmbox@bodystyle{#1}}
```

2.2 Package options

The only package option is `nothm`, which prevents from redefining the standard `\newtheorem` command.

```
\newif\ifthmbox@newtheorem
\thmbox@newtheoremtrue
\DeclareOption{nothm}{\thmbox@newtheoremfalse}
```

All other package options are considered to be formatting options and are parsed using the `keyval` package.

```
\DeclareOption*{\expandafter\thmboxoptions\expandafter{\CurrentOption}}
\ProcessOptions\relax
```

2.3 Formatting

We now define the code for formatting boxes. When a box is to be cut, the idea is the following : we assume the box `\thmbox@box` to contain the whole text, with an aribtrary height. First we compare the height of the box to the available space on the current page. If there is enough space, we simply place the box at once with the decorations around it. Otherwise, we cut the box at the available height, place the first part on the page, and proceed with the remaining text. The following steps are similar except that the available space is the height of the whole page.

The drawback with this method, apart from the fact that it cannot produce page breaks as good as TEX's page builder does, is that the construction cannot be nested. Putting a box of that kind into another works as long as there is no page break, but has unpredictable results when a page break occurs. If we wanted to allow nested boxes, we would have to completely rethink the system, and it would result in a much heavier code. Is this really necessary ?

`\thmbox@box` `\thmbox@box@` `\thmbox@dim`

We first introduce internal variables. We will use two boxes and a length register for computations during formatting.

```
\newbox\thmbox@box
\newbox\thmbox@box@
\newdimen\thmbox@dim
```

`\thmbox@put`

Placing a box after appropriate cutting is performed by the macro `\thmbox@put`. Its argument is the register that holds the box to be placed. The effect is simply to insert the appropriate spaces and rules, according to the value of `\thmbox@style`, and switch back to vertical mode.

```
\def\thmbox@put#1{
  \vskip\z@%
  \noindent%
  \hbox{%
    {\dimen0=\thmbox@leftmargin%
     \advance\dimen0-\thmbox@hskip%
     \advance\dimen0-\thmbox@thickness%
     \hskip\dimen0}%
    \vrule width \thmbox@thickness%
    \hskip\thmbox@hskip%
    \box#1%
    \ifx\thmbox@style L%
      \hskip\thmbox@hskip%
      \vrule width \thmbox@thickness%
    \fi}%
  \par\nobreak}
```

`\thmbox@start`

The first step as described above is always applied to the box `\thmbox@box`, and `\thmbox@box@` is reserved for the part that is cut out by a `\vsplit`. The macro `\thmbox@start` performs the first step in the process of formatting a complete box. A difficult point occurs here : if the box is to be inserted at the beginning of a page, it may happen that `\output` has not been called yet, so the height

of the current vertical material (the value of \pagetotal) may exceed the page height (contained in \vsize). In this case, the available space on the page must be computed as $2 \times$ \vsize $-$ \pagetotal instead of \vsize $-$ \pagetotal. This explains the conditional at the beginning of the macro.

```
\def\thmbox@start{%
  \ifthmbox@cut%
    \ifdim\pagetotal>\vsize%
      \thmbox@dim=2\vsize%
    \else%
      \thmbox@dim=\vsize%
    \fi%
    \advance\thmbox@dim -\pagetotal%
    \ifdim\thmbox@dim>\ht\thmbox@box%
      \thmbox@put\thmbox@box%
    \else%
      \setbox\thmbox@box@=\vsplit\thmbox@box to \thmbox@dim%
      \thmbox@put\thmbox@box@%
      \thmbox@page%
    \fi%
  \else%
    \thmbox@put\thmbox@box%
  \fi}
```

\thmbox@page The following steps are for the case when a box is placed at the beginning of an empty page. They are handled by the macro \thmbox@page, which systematically inserts a page break before the box (unless the box is empty, which may happen after cutting near the end of a box).

```
\def\thmbox@page{%
  \ifvoid\thmbox@box%
  \else%
    \eject%
    \ifdim\vsize<\ht\thmbox@box%
      \setbox\thmbox@box@=\vsplit\thmbox@box to \vsize%
      \thmbox@put\thmbox@box@%
      \thmbox@page%
    \else%
      \thmbox@put\thmbox@box%
    \fi%
  \fi}
```

After the formatting of text boxes, we have to handle the header of the box as well as the part below the last box, where there may be a horizontal rule.

\thmbox@head The macro \thmbox@head produces a header with the text in its argument. A horizontal rule is possibly placed below, after a space of \thmbox@vskip. This space is produced by adding an invisible vertical rule in the title so that the size of the space does not depend on the text. Finally we add an invisible horizontal rule to switch back to vertical mode without making a new paragraph.

```
\def\thmbox@head#1{%
  \par\noindent\vbox{%
    \setbox\thmbox@box@=\hbox{%
      \vrule width 0mm height 0mm depth \thmbox@vskip%
      #1}%
    \copy\thmbox@box@%
    \ifthmbox@underline%
      \hrule width \wd\thmbox@box@ height \thmbox@thickness%
    \fi}%
  \hrule height 0mm\relax}
```

\thmbox@tail The terminal par of the box is produced by the macro `\thmbox@tail`. As opposed to the previous one, this one is very dependent on the style of the box. However we assume is is called only with a style equal to `M` or `L`, and not for the style `S`.

```
\def\thmbox@tail{%
  \hrule height 0mm%
  \ifx\thmbox@style M%
    \thmbox@dim=1cm%
  \else\ifx\thmbox@style L%
    \thmbox@dim=\hsize%
    \advance\thmbox@dim-\thmbox@leftmargin%
    \advance\thmbox@dim-\thmbox@rightmargin%
    \advance\thmbox@dim2\thmbox@hskip%
    \advance\thmbox@dim2\thmbox@thickness%
  \fi\fi%
  \noindent%
  {\dimen0=\thmbox@leftmargin%
   \advance\dimen0-\thmbox@hskip%
   \advance\dimen0-\thmbox@thickness%
   \hskip\dimen0}%
  \vrule width \thmbox@dim height \thmbox@thickness%
  \par}
```

2.4 LaTeX interface

2.4.1 Environnements

thmbox Now that the previous commands are defined, we can actually define the `thmbox` environment. The prefix part is used to set the formatting options and produce the header, then start a vertical box with appropriate width for the contents of the text. The postfix part closes this box, formats it using the previously defined commands, and possibly inserts material at the end.

The code contains groupings that may appear unnecessary, around `#2` in the header and around the environment's contents. They are here to obtain a satisfactory behaviour when using colours.

```
\newenvironment{thmbox}[2][]{%
  \parskip\z@%
  \setkeys{thmbox}{#1}%
  \ifx\thmbox@style S\else\ifx\thmbox@style M\else\ifx\thmbox@style L\else%
```

```
    \PackageWarning{thmbox}{\thmbox@style\ is not a valid style for
      \string\thmbox, using M}%
    \let\thmbox@style=M%
  \fi\fi\fi%
  \thmbox@head{{#2}}\nobreak\relax%
  \thmbox@dim=\hsize%
  \advance\thmbox@dim-\leftskip%
  \advance\thmbox@dim-\rightskip%
  \setbox\thmbox@box=\vbox\bgroup%
    \hsize=\thmbox@dim%
    \advance\hsize -\thmbox@leftmargin%
    \advance\hsize -\thmbox@rightmargin%
    \textwidth=\hsize%
    \linewidth=\hsize%
    \vskip\thmbox@vskip%
    \begingroup}{\endgroup%
    \vskip\thmbox@vskip%
  \egroup%
  \thmbox@start%
  \ifx\thmbox@style S\else\thmbox@tail\fi%
  \@endparenv}
```

`leftbar` The definitions used for wrapping boxes allow for easily defining an evironment that places a vertical bar on the side of the text. The code of this enivonment is basically that of `thmbox` without the header and tail material :

```
\newenvironment{leftbar}[1][]{%
  \setkeys{thmbox}{#1}%
  \par\vskip\thmbox@vskip%
  \setbox\thmbox@box=\vbox\bgroup%
    \hsize=\textwidth%
    \advance\hsize -\thmbox@leftmargin%
    \advance\hsize -\thmbox@rightmargin%
    \begingroup}{\endgroup%
    \vskip\thmbox@vskip%
  \egroup%
  \thmbox@start%
  \vskip\thmbox@vskip\par}
```

2.4.2 LaTeX theorems

`\newboxtheorem` In order to use this package transparently in a LaTeX text, we define the command `\newboxtheorem` on the model of the standard `\newtheorem` command. The code of this version respects the semantics of the original one, with two possible syntaxes :

- `\newtheorem{`*environment*`}[`*counter*`]{`*title*`}` to use an already existing counter,
- `\newtheorem{`*environment*`}{`*title*`}[`*coupter*`]` to create a new counter that possibly depends on an existsing one.

We add an optional first argument that contains formatting options for the boxes used in a particular kind of theorem. The definition of this macro is in a very LaTeXish style...

```
\def\newboxtheorem{%
  \@ifnextchar[{\thmbox@newA}{\thmbox@newA[]}}%
```

The sub-macros eventually call `\thmbox@new`, which performs the actual theorem definition according to four arguments, respectively the formatting options, the environment name, the title used and the name of LaTeX counter.

The macro `\thmbox@newA` is used when options are specified. We store the options in the macro `\thmbox@temp` for stability. This is needed because it may contain an option of the form `titlestyle={...#1...}` which could cause problems in a `\def`.

```
\def\thmbox@newA[#1]#2{%
  \def\thmbox@temp##1{#1}%
  \@ifnextchar[{\thmbox@newC{#2}}{\thmbox@newD{#2}}}
```

The macro `\thmbox@newC` corresponds to the case where an existing counter is used for the new environment.

```
\def\thmbox@newC#1[#2]#3{%
  \expandafter\thmbox@new\expandafter{\thmbox@temp{####1}}{#1}{#3}{#2}}
```

The macro `\thmbox@newD` corresponds to the case where a new counter is created.

```
\def\thmbox@newD#1#2{%
  \@ifnextchar[{\thmbox@newE{#1}{#2}}{%
    \newcounter{#1}%
    \expandafter\thmbox@new\expandafter{\thmbox@temp{####1}}{#1}{#2}{#1}}}
```

The macro `\thmbox@newE` corresponds to the case where the new counter depends on another counter.

```
\def\thmbox@newE#1#2[#3]{%
  \newcounter{#1}[#3]%
  \expandafter\def\csname the#1\endcsname{%
    \csname the#3\endcsname.\arabic{#1}}
  \expandafter\thmbox@new\expandafter{\thmbox@temp{####1}}{#1}{#2}{#1}}
```

The macro `\thmbox@new` performs the actual environment definition. Because of the way we handle the environment's optional argument, we don't define the new environment using LaTeX macros. The drawback is that there is no error checking.

```
\def\thmbox@new#1#2#3#4{%
  \expandafter\def\csname#2\endcsname{%
    \setkeys{thmbox}{#1}%
    \@ifnextchar[{\thmbox@beginA{#3}{#4}}{%
      \thmbox@begin{#3}{#4}{}}}%
  \expandafter\def\csname end#2\endcsname{%
    \endthmbox\smallbreak}}
```

Like previously, the macro `\thmbox@beginA` is used when the optional argument is present.

```
\def\thmbox@beginA#1#2[#3]{%
  \thmbox@begin{#1}{#2}{\thmbox@titlestyle{#3}}}
```

The macro `\thmbox@begin` is responsible for opening a theorem environment as defined by the new version of `\newtheorem`. Its arguments contain respectively options for the box, the name of the element (e.g. "Theorem"), the counter used and the text to insert after the number.

```
\def\thmbox@begin#1#2#3{%
  \medbreak%
  \refstepcounter{#2}%
  \thmbox{\thmbox@headstyle{#1}{\csname the#2\endcsname}#3}%
  \thmbox@bodystyle\ignorespaces}
```

Optionally (see the package option `nothm`), this macro replaces the original `\newtheorem` so that the package can be used without modifying an already prepared text.

```
\ifthmbox@newtheorem
\let\newtheorem\newboxtheorem
\fi
```

2.5 Examples and proofs

`example` For examples, we write the text in a smaller face, with an extra left margin and a mark at the beginning.

```
\def\example{}
\@ifundefined{examplename}{\def\examplename{Example}}{}
\renewenvironment{example}[1][\examplename]{%
  \par\smallbreak\small%
  \list{\hspace\labelsep\textbf{#1\,:}}{%
    \leftmargin=\parindent%
    \labelwidth=\parindent}%
  \item\relax}{%
  \endlist}
```

`proof` The format of proofs is mostly that of examples, with two differences : the right margin is also extended, and a black square is placed in this extra margin at the end of the last line.

```
\def\proof{}
\@ifundefined{proofname}{\def\proofname{Proof}}{}
\renewenvironment{proof}[1][]{%
  \small%
  \list{\hspace\labelsep\textbf{\proofname\ #1\unskip\,:}}{%
    \topsep=\smallskipamount%
    \partopsep=0pt%
    \leftmargin=\parindent%
    \rightmargin=\parindent%
    \listparindent=\parindent%
    \labelwidth=\parindent}%
  \item\relax\ignorespaces}%
```

```
{\parskip\z@%
 \par\noindent%
 \setbox\thmbox@box=\hbox{%
   \kern .5em\vbox{%
     \hrule width .7em height .7em
     \vskip\baselineskip}}%
 \wd\thmbox@box=0mm%
 \ht\thmbox@box=0mm%
 \hfill\box\thmbox@box%
 \endlist\par}
```