functan*
Macros for functional analysis and related domains

Antoine Lejay
<Antoine.Lejay@iecn.u-nancy.fr>

July, 13 2004
Printed on July 22, 2004

1 Introduction

This package is designed especially for the people working in domains of
mathematics such as functional analysis and PDE theory. Of course, this
package may be used in other domains.

In functional analysis and PDE theory, one may face a lot of names of
spaces, sometimes depending on one or more parameters, such as $L^2(\Omega)$,
$H^1_0(\Omega)$, ... Besides, one has to deal with norms, convergence and scalar prod-
uct for each of these spaces. As a control sequence, i.e., a TEX command,
consists only of letters, using TEX's standard macros is not convenient for
dealing with such spaces. Yet composing each of these space names at each
their occurrence increases the risk of errors, and tends to be tedious.

Fortunately, TEX provides some way to create commands whose name
may contain arbitrary characters. We exploit this possibility to deal with
names of functional spaces and related objects (norms, convergence, ...).

This package allows for example to typeset:

We denote by $L^2(\Omega)$ the space of measurable functions on $\Omega$
which are square integrable. The associated norm $\|f\|_{L^2(\Omega)}$ is
defined by $\|f\|_{L^2(\Omega)} = (\int_{\Omega} f(x)^2 dx)^{1/2}$ and the correspond-
ing scalar product is $\langle f, g \rangle_{L^2(\Omega)} = \int_{\Omega} f(x)g(x)dx$. We write
$f_n \xrightarrow{L^2(\Omega)} f$ if the sequence $(f_n)_n$ converges to $f$ in $L^2(\Omega)$. If
for all $g \in L^2(\Omega)$, $\langle f_n, g \rangle_{L^2(\Omega)}$ converges to $\langle f, g \rangle_{L^2(\Omega)}$, then we
say that $(f_n)_n$ converges weakly to $f$ in $L^2(\Omega)$, and we write
$f_n \xrightarrow{L^2(\Omega)} f$.

*Package version v1.0 of 2004/07/03.
The code for this example is the following:

\Macro{L2Om}{\mathrm{L}^2(\Omega)}

We denote by $\m{L2Om}$ the space of measurable functions on $\Omega$ which are square integrable. The associated norm $\norm{L2Om}{\cdot}$ is defined by $\norm{L2Om}{f}=\left(\int_{\Omega} f(x)^2 \, dx\right)^{1/2}$ and the corresponding scalar product is $\scalprod{L2Om}{f}{g}=\int_{\Omega} f(x)g(x) \, dx$.

We write $f_n \conv{L2Om}{n\to\infty} f$ if the sequence $(f_n)_n$ converges to $f$ in $\m{L2Om}$.

If for all $g \in \m{L2Om}$, $\scalprod{L2Om}{f_n}{g}$ converges to $\scalprod{L2Om}{f}{g}$, then we say that $(f_n)_n$ converges weakly to $f$ in $\m{L2Om}$, and we write $f_n \wkconv{L2Om}{n\to\infty} f$.

This package uses of some commands of the package amsmath (some of these commands are available only for version 2.0 or higher), which is then automatically loaded. All the options of amsmath may be defined as options of functan.

Before using functan, be sure you have a recent version of amsmath installed in your computer.

2 Loading the package

This package is loaded as any package by the command \usepackage. This package has no options, except those of amsmath in order to avoid an option clash. Thus, the possible options are: leqno, intlimits, nointlimits, sumlimits, nosumlimits, namelimits, nonamelimits, reqno, centertags, tbtags, cmex10 and fleqn.

Incompatibilities : This package has no known incompatibilities with other packages.

3 The commands

3.1 An alternative system of macros

This system consists in associating a sequence of tokens to a name by the command \Macro, and call it later by the command \m.

\Macro The command \Macro{⟨name⟩}{⟨expansion⟩} associates the list of tokens ⟨expansion⟩ to the list of tokens ⟨name⟩, which may contain any arbitrary
sequence of characters (except \{, \} and active characters). This macro produces a warning message “The macro ‘⟨name⟩’ has already been defined” if ⟨name⟩ already exists.

Nota: The sequence ⟨name⟩ may correspond to the control sequence of a \TeX macro without creating interferences. Internally, the system stores the expansion of ⟨name⟩ as the \TeX control sequence functan@macro@⟨name⟩. The command \m{⟨name⟩} gives the expansion corresponding to ⟨name⟩ if a couple ⟨name⟩ and ⟨expansion⟩ has already been defined by \Macro. Otherwise, an error message “The macro ‘⟨name⟩’ is not defined” is produced. The error message “No macro’s name given” is also produced if the argument is given, that is if \m{} is encountered.

\Macro{L2Om}{\mathrm{L}^2(\Omega)}
Denote by $\m{L2Om}$ the space of functions such that... Denote by $L^2(\Omega)$ the space of functions such that...

Nota: A command \mempty is also defined whose action is similar to \m, except that it does not produce an error message if its argument is empty.

3.2 Norms

The command \norm allows to typeset norms easily.

\norm The command \norm[⟨size⟩]{⟨name⟩}{⟨arg⟩} is roughly an equivalent to \left\|⟨arg⟩\right\| when one call \norm[]{}{⟨arg⟩}, then the delimiter || is used with its normal size. If ⟨size⟩ is equal to \Big, Big, bigg or Bigg, then ⟨size⟩ is inserted before the delimiters ||.

Nota: If ⟨size⟩ is not one of the elements of Table 1, the the normal size is used.

\begin{align*}
\norm{f} & \|f\| \\
\norm[\text{T2Om}]{f} & \|f\|_{L^2(\Omega)} \\
\norm[\Big]{f} & \|f\|_{L^2(\Omega)} \\
\norm[\text{T2Om}]{\frac{1}{f}} & \|\frac{1}{f}\|_{L^2(\Omega)} \\
\norm[]{}{\frac{1}{f}} & \|\frac{1}{f}\|_{L^2(\Omega)}
\end{align*}
The command \newnorm\{(name)\}{\langle\expansion\rangle}\{\langle\arg 1\rangle\}{\langle\arg 2\rangle}\{\langle\name\rangle\} allows to override the behavior of the \norm macro. Instead of using a |...| structure, the user may for example use other delimiters. Here \langle\name\rangle is any sequence of tokens (except {, } and active characters), already defined by \Macro or not. The arguments #1 (for the size) et #2 (for the argument) may be used in \langle\expansion\rangle. The macros \casesize and \delonearg (see Section 3.5 below) may be used to create some new kind of norms.

\newnorm\{L2Om\}{\delonearg[#1]{#2}{|}{|}}
\norm\{L2Om\}\{f\}
\norm\{Big\}\{L2Om\}\{f\}

Nota: It seems that some side effects appear when \newnorm is used inside a group. Hence, it is recommended to use this command in the preamble.

### 3.3 Scalar products

Macros for scalar products are similar to macros for the norms, except that two argument are needed.

The command \scalprod\{(size)\}{\langle\name\rangle}\{\langle\arg 1\rangle\}{\langle\arg 2\rangle}\{\langle\name\rangle\} creates roughly \langle\langle\arg 1\rangle,\langle\arg 2\rangle\rangle_{\m\langle\name\rangle}. The optional \langle\size\rangle argument shall be taken in Table 1, and acts as a modifier for the size of the brackets.

\scalprod\\{f\}\{g\}
\scalprod\{L2Om\}\{f\}\{g\}
\scalprod\{L2Om\}\{f\}\{\frac{1}{g}\}
\scalprod\{big\}\{L2Om\}\{f\}\{\frac{1}{g}\}

The command \newscalprod\{(size)\}{\langle\subscr\rangle}\{\langle\arg 1\rangle\}{\langle\arg 2\rangle}\{\langle\subscr\rangle\} acts like \scalprod, except that \langle\subscr\rangle is not the name already defined by \Macro, but a sequence of tokens put in subscript after the closing bracket. The rules are the same as for \norm.

\newscalprod\{X,Y\}\{Ax\}\{y\}
\newscalprod\{X,Y\}\{B^{-1}x\}\{y\}

The command \newscalprod\{(name)\}{\langle\expansion\rangle} allows to override the
behavior of \texttt{\scaprod}, as for \texttt{\newnorm}. Here, \langle expansion \rangle may use the three arguments \#1 (for the optional size), \#2 (\langle arg 1 \rangle) and \#3 (\langle arg 2 \rangle). The command \texttt{\towarg} (see Section 3.5 below) may be used to help the user to deal with new types of scalar products.

\begin{verbatim}
\newscalprod{L2Om}{\deltwoarg[#1]{#2}{#3}{(}{)}{|}}
\scaprod{L2}{f}{g} \quad (f | g)
\scaprod{[Big]}{L2}{f}{g} \quad \left< f \middle| g \right>
\scaprod{}{L2}{f^2}{g^2} \quad (f^2 | g^2)
\end{verbatim}

\textit{Nota:} It seems that some side effects appear when \texttt{\newnorm} is used inside a group. Hence, it is recommended to used this command in the preamble.

### 3.4 Convergence

The package \texttt{amsmath} provides a way to create extensible arrows, whose size depends on the material put above and below the arrow.

The command \texttt{\conv\{\langle name\rangle\}\{\langle below\rangle\}} creates an extensible arrow by putting the expansion of \langle name \rangle above the arrow, and \langle below \rangle under the arrow. This command uses the commands \texttt{\xrightarrow} of \texttt{amsmath}. However, the \texttt{\xrightarrow} command takes as an optional argument the material to be put below. Here, it is a mandatory second argument.

\begin{verbatim}
x_n\conv{}{} x \quad x_n \xrightarrow{} x
f_n\conv\{\text{uniformly}\}\{n\to\infty\} f \quad f_n \xrightarrow{n\to\infty} f
f_n\conv*\{\text{uniformly}\}\{\text{uniformly}\}\{n\to\infty\}\{n\to\infty\} f \quad f_n \xrightarrow{n\to\infty} f
\end{verbatim}

The command \texttt{\newconv\{\langle name\rangle\}\{\langle expansion\rangle\}} allows to override the behavior of \texttt{\conv\{\langle name\rangle\}}, whether \langle name \rangle corresponds to a name defined by \texttt{\Macro} or not. The token list \langle expansion \rangle may use one argument \#1.

\begin{verbatim}
f_n\newconv\*\{L2Om\}\{\star\text{\text{-}L}\}\{L2\}\{\star\text{\text{-}L}\}\{\Omega\}\{n\to\infty\}\{n\to\infty\} f \quad f_n \xrightarrow{n\to\infty} f
\end{verbatim}

The commands \texttt{\leftconv\{\langle name\rangle\}\{\langle below\rangle\}} and \texttt{\leftconv\*\{\langle above\rangle\}\{\langle below\rangle\}} act like \texttt{\conv} and \texttt{\conv*}, except that an arrow pointing to the left (\texttt{\leftarrow}) is used instead of an arrow pointing to the right (\texttt{\rightarrow}).

The commands \texttt{\leftrightconv\{\langle name\rangle\}\{\langle below\rangle\}} and \texttt{\leftrightconv\*\{\langle above\rangle\}\{\langle below\rangle\}} act like \texttt{\conv} and \texttt{\conv*}, except that an arrow pointing to the left (\texttt{\leftarrow}) is used instead of an arrow pointing to the right (\texttt{\rightarrow}).
and \leftrightconv*(\langle above\rangle)\{\langle below\rangle\} act like \conv and \conv*, except that an arrow pointing both to the left and to the right (\leftarrow\rightarrow) is used instead of an arrow pointing to the right (\rightarrow).

There are other commands similar to \conv, \conv*, \leftconv, \leftconv*, \leftrightconv and \leftrightconv* that is with the same arguments, but with different types of arrows.

The commands \wkconv, \wkconv*, \leftwkconv, \leftwkconv*, \leftrightwkconv and \leftrightwkconv* use \rightarrow\leftarrow instead of \rightarrow\rightarrow. Such an arrow is generally used to denote weak convergence (hence the \text{wk}).

f_n\wkconv\{n\to\infty\} f

f_n\wkconv[H10]{n\to\infty} f

\µ_n\Conv\{n\to\infty\} \µ

x<0\Leftrightconv*\{\text{Lemma 1.1}\}\{\langle above\rangle\}\{\langle below\rangle\} f(x)\geq 0

\H_{010}^{m+n}

\Conv\{n\to\infty\} \Conv^*\{n\to\infty\}

\Leftconv\{n\to\infty\} \Leftconv^*\{n\to\infty\}

\Leftrightconv\{n\to\infty\} \Leftrightconv^*\{n\to\infty\}

\text{3.5 Defining macros}

The package functan provides a few macros to help the user to define its own norms, scalar products, ...

The command \delonearg\{size\}\{\langle arg\rangle\}\{\langle left del\rangle\}\{\langle right del\rangle\} surrounds \langle arg\rangle the left delimiter \langle left del\rangle and the right delimiter \langle right del\rangle. The possible values of the optional argument \langle size\rangle (which is by default set to auto) are given in Table 1.

\newcommand{\event}[2][\text{auto}]{\delonearg[#1]{#2}{\{}{}{}}
\newcommand{\ket}[2][\text{auto}]{\delonearg[#1]{#2}{\langle}{\rangle}}
\newcommand{\bra}[2][\text{auto}]{\delonearg[#1]{#2}{\langle}{\rangle}}

\deltwoarg\langle size\rangle\{\langle arg 1\rangle\}\{\langle arg 2\rangle\}...

\left\langle\text{left del}\rangle\{\langle right del\rangle\}\{\langle sep\rangle\} is similar to \delonearg, except that it separates \langle arg 1\rangle and \langle arg 2\rangle by a separator \langle sep\rangle. If \langle sep\rangle is set to 1, then a vertical bar is used, with an automatic adjustment to the size of the delimiters.
\newcommand{\set}[3][auto]{\deltwoarg[#1]{#2}{#3}{\{}{\}}{,\,}}
\set{x\geq 0}{P(x)=0}

\newcommand{\braket}[3][auto]{\deltwoarg[#1]{#2}{#3}{\langle}{\rangle}{|}}
\braket{Ax}{y} \langle Ax \mid y \rangle

The command \texttt{\casesize{⟨size⟩}}\{⟨auto⟩\}\{⟨empty⟩\}...
...\{⟨big⟩\}\{⟨Big⟩\}\{⟨bigg⟩\}\{⟨Bigg⟩\} takes \texttt{⟨size⟩} in Table 1 and executes
\{⟨empty⟩\} if \texttt{⟨size⟩} is an empty list of tokens, \{⟨auto⟩\} if \texttt{⟨size⟩} is \texttt{auto}, \{⟨big⟩\}
if \texttt{⟨size⟩} is \texttt{big}, ... The error message “Size argument ‘⟨size⟩’ of ‘casesize’
not valid” is produced if \texttt{⟨size⟩} is not in Table 1

\newcommand{\delcasesize}[3][auto]{\deltwoarg[#1]{#2}{#3}{\langle}{\rangle}{|}}
\delcasesize{⟨size⟩}{⟨auto⟩}{⟨empty⟩}{⟨other⟩} is simi-
lar to \texttt{\casesize}, except that \texttt{⟨other⟩} is executed if \texttt{⟨size⟩} is different from
\texttt{auto} or from an empty list of tokens.

4 Examples

4.1 Macros with arguments

It is possible to define a macro \texttt{⟨name⟩} such that \texttt{\m{⟨name⟩}} accepts some
arguments. For that, it is sufficient that it expands into a \TeX command
that accepts arguments.

\newcommand{\evolution}[2]\{\m{#1}(0,T;\m{#2})\}
\Macro{L20T}{\evolution{L2}}
\m{L20T}{H10Om}\cap\m{L20T}{L2Om} \quad L^2(0,T;H^1_0(\Omega)) \cap L^2(0,T;L^2(\Omega))

4.2 Matrix norms

A matrix norm with three bars may be defined by

\newnrm{matrix}{\delcasesize{#1}%\left|\left|\left| #2\right|\right|\right|}{\|#2\|}{\|\#1\#1\#1\#1\#2 #1\#1\#1\|}}
so that
\nrm{matrix}{A^{-2}}\leq \nrm{matrix}{A}^{-2} \quad |||A^2||| \leq |||A|||^2
\nrm[]{matrix}{A^{-2}}\leq \nrm{matrix}{A}^{-2} \quad |||A^2||| \leq |||A|||^2

4.3 Sets

An example of code for defining sets was given with the documentation of
\texttt{\deltwoarg}. Here is another possibility
4.4 Duality products

The command \texttt{\textbackslash casesize} may be used for defining a duality product, which is a bit tricky since we want the position of name of the dual space to be adjusted to the size of the delimiters. The command \texttt{\textbackslash dual} defined by

\begin{verbatim}
\newcommand{\dual}[5][auto]{\casesize{#1}{%
\left\langle #4;#5\right\rangle #3
\right.}_{\m{#2}}}
\end{verbatim}

may be used to produce

\begin{verbatim}
\Macro{H1}{\texttt{\textbackslash mathrm{H}^{-1}}}
\Macro{H-1}{\texttt{\textbackslash mathrm{H}^{-1}}}
\dual{H-1}{H1}{f}{\frac{1}{g}}{H^1}
\dual[]{H-1}{H1}{f}{\frac{1}{g}}{H^1}
\dual[bigg]{H-1}{H1}{f}{\frac{1}{g}}{H^1}
\end{verbatim}

5 The code

5.1 Package heading

This package uses some code of the \texttt{amsmath} package. All the options of \texttt{amsmath} may be set as options of \texttt{functan} in order to avoid an option clash.

\begin{verbatim}
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{functan}[2004/07/03 v1.0, Macros for functional analysis]
\DeclareOption{leqno}{\PassOptionsToPackage{leqno}{amsmath}}
\DeclareOption{intlimits}{\PassOptionsToPackage{intlimits}{amsmath}}
\DeclareOption{nointlimits}{\PassOptionsToPackage{leqno}{amsmath}}
\end{verbatim}
5.2 Defining macros

This commands defines a new macro with name \textit{name}, by creating a \LaTeX command \texttt{functan@macro@\{name\}}. A warning message is sent if \textit{name} has already been defined.

\begin{verbatim}
\newcommand{\Macro}[2][]{\@ifundefined{functan@macro@#1}{}{\PackageWarning{functan}{The macro '#1' has already been defined}}}\@namedef{functan@macro@#1}{#2}}
\end{verbatim}

This command allows to call the macro defined by \textit{name}. An error message is sent if \textit{name} has not been defined, or if no argument is given. The macro \texttt{mempty} provides no error message is the argument is empty. It is used to deal with \texttt{conv} macros.

\begin{verbatim}
\newcommand{\m}[1][1]{\@ifempty{#1}{\PackageError{functan}{No macro's name given}{}\@ifundefined{functan@macro@#1}{}{\PackageError{functan}{The macro ‘#1’ is not defined}{}\@nameuse{functan@macro@#1}}}{\PackageError{functan}{The macro ‘#1’ is not defined}{}\@ifundefined{functan@macro@#1}{}{\PackageError{functan}{The macro ‘#1’ is not defined}{}\@nameuse{functan@macro@#1}}}}
\end{verbatim}
5.3 Dealing with size

The macro \texttt{\delcasesize} is an alias for \texttt{\functan@casesize}. These macros require four arguments. This first one is the size among $\emptyset$ (empty argument for normal size) \texttt{auto} and one of the delimiters possible size (\texttt{big}, \texttt{Big}, \texttt{bigg}, \texttt{Bigg}). It \texttt{auto} is detected, then the second argument is executed. It no argument is given, then the third argument is executed. Otherwise, the fourth argument is executed. Another macro, \texttt{\casesize} (see below), is intended to execute a different code the the six distinct possibilities ($\emptyset$, \texttt{auto}, \texttt{big}, \texttt{Big}, \texttt{bigg}, \texttt{Bigg}).

\begin{verbatim}
\global\def\functan@size@auto{auto}
\newcommand{\functan@casesize}[4]{\def\functan@size{#1}\
    \ifx\functan@size\functan@size@auto #2\else\@ifempty{#1}{#3}{#4}\fi}
\let\delcasesize\functan@casesize
\end{verbatim}

The first argument is one of the delimiter’s size specification (\texttt{auto}, \texttt{∅}, \texttt{big}, \texttt{Big}, \texttt{bigg}, \texttt{Bigg}), and the six others are the different codes for each of the possible size in the order given just above.

\begin{verbatim}
\global\def\functan@size@big{big}
\global\def\functan@size@Big{Big}
\global\def\functan@size@bigg{bigg}
\global\def\functan@size@Bigg{Bigg}
\newcommand{\casesize}[7]{\def\functan@size{#1}\
    \@ifempty{#1}{#3}{\ifx\functan@size\functan@size@auto{#2}\else\
        \ifx\functan@size\functan@size@big{#4}\else\
            \ifx\functan@size\functan@size@Big{#5}\else\
                \ifx\functan@size\functan@size@bigg{#6}\else\
                    \ifx\functan@size\functan@size@Bigg{#7}\else\
                        \PackageError{functan}{Size argument ‘#1’ of ‘casesize’ not valid}\%\fi\fi\fi\fi\fi}}
\end{verbatim}

These two macros may be used for defining a pair of delimiters with automatic or manual size adjustment.

\begin{verbatim}
\newcommand{\delonearg}[4][auto]{\functan@casesize{#1}{\left#3 #2 \right#4}{#3 #2 #4}{\@nameuse{#1l}#3 #2 \@nameuse{#1r}#4}}
\newcommand{\deltwoarg}[6][auto]{\def\functan@vert@bar{|}\
    \edef\functan@arg@bar{#6}\
    \ifx\functan@arg@bar\functan@vert@bar\PackageError{functan}{Size argument ‘#1’ of ‘casesize’ not valid}\%\fi\fi\fi\fi\fi}
\end{verbatim}

\textit{Nota:} A \texttt{big-g-g} argument is transformed into \TeX control sequence with the command \texttt{@nameuse}. If the argument of \texttt{nameuse} is not a control sequence for an already defined control sequence, then \texttt{@nameuse{(...)}} is transformed into an empty list.
5.4 Norms

\functan@norm
This is the generic macro for the norms, which uses double bars $\|\|$ as delimiters.

\functan@starred@norm
\functan@nonstarred@norm
This two macros call the command \functan@norm either with (non-starred form) or without (starred form) expending the second argument $\langle name \rangle$ as the name of a macro defined by \Macro. Note that if the second argument is empty in \functan@nonstarred@norm, then no error message is produced. This allows to have a subscript just after the norm such as in $\|f\|_1$ without producing an error message of type “double subscript error”. Note that the previous example may also be written $\|1\|f\|$. For the non-starred version, if a TeX macro \functan@named@norm@\langle name \rangle exists (which is defined by \newnorm), then this macro is called instead of \functan@norm.

\newnorm
This macro allows to override the behavior of the \norm macro by defining a TeX command \functan@named@norm@\langle name \rangle.

\norm
This macro is the one which is finally used. The presence of a star * is checked and either \functan@starred@norm or \functan@nonstarred@norm is called in consequence.
5.5 Convergence

Nota: It is for this set of macros that the compatibility with version 2.0 of amsmath or higher is important.

\newconv

This macro allows to override the behavior of the \conv macro, by defining a \TeX control sequence \functan@named@conv@⟨name⟩.

\newcommand{\newconv}[1]{%
\@namedef{functan@named@conv@#1}##1}

\conv

This macros stands for the usual convergence symbol (→). The presence of a star * is checked, and either \functan@nonstarred@conv or \functan@starred@conv is called in consequence. If a \TeX command called \functan@named@conv@⟨name⟩ exists, where ⟨name⟩ is the first argument of \conv, then this command is called instead of the others.

The command \xrightarrow is defined in the amsmath package.

\newcommand{\xrightarrow}{%
\@ifstar{\functan@starred@conv}{\functan@nonstarred@conv}}
\newcommand{\functan@nonstarred@conv}[2]{%
\@ifundefined{functan@named@conv@#1}{}
{\@nameuse{functan@named@conv@#1}{#2}}}
\newcommand{\functan@starred@conv}[2]{\xrightarrow{#2}{#1}}

\xleftarrowglobalarrow \Xleftarrowglobalarrow \Xrightarrowglobalarrow \Xleftglobalarrow

These commands are defined in a way similar to the one in \xrightarrow in the package amsmath. The first four arguments are lengths added on each side of the argument.

\newcommand{\xleftarrowglobalarrow}[2][%]
\ext@arrow 9999\leftarrowfill@{#1}{#2}}
\newcommand{\Xleftarrowglobalarrow}[2][%]
\ext@arrow 0359\Leftarrowfill@{#1}{#2}}
\newcommand{\Xrightarrowglobalarrow}[2][%]
\ext@arrow 3095\Rightarrowfill@{#1}{#2}}
\newcommand{\Xleftglobalarrow}[2][%]
\ext@arrow 9999\Leftarrowfill@{#1}{#2}}

\leftconv \leftrightconv

The commands \leftconv and \leftrightconv are defined as the command \conv, except that it is not possible to override them (however, any kind of arrow may be used with the command \newconv).

\newcommand{\leftconv}{%
\@ifstar{\functan@starred@leftconv}{}
{\functan@nonstarred@leftconv}}
\newcommand{\functan@nonstarred@leftconv}[2]{%
\xleftarrow{#2}{\mempty{#1}}}
\newcommand{\functan@starred@leftconv}[2]{\xleftarrow{#2}{#1}}
\newcommand{\leftrightconv}{%
\@ifstar{\functan@starred@leftrightconv}{}
{\functan@nonstarred@leftrightconv}}
\newcommand{\functan@nonstarred@leftrightconv}[2]{%
\xleftarrow{#2}{\mempty{#1}}}
\newcommand{\functan@starred@leftrightconv}[2]{\xleftarrow{#2}{#1}}
These macros are defined like \conv. These macro may be used for the weak convergence. Its definition is similar to the one of \conv.
5.6 Scalar products

The definition of the scalar product is similar to the definition of norms.

\newcommand{\scalprod}{\@ifstar{\functan@starred@scalprod}{\functan@nonstarred@scalprod}}
\newcommand{\functan@starred@scalprod}{[4]\[auto\]{\deltwoarg[#1]{#3}{#4}{\langle}{}{\rangle}{,}\@ifempty{#2}{}{_{\m{#2}}}}}
\newcommand{\functan@nonstarred@scalprod}{[4]\[auto\]{\@ifundefined{functan@scalprod@named@#2}{\deltwoarg[#1]{#3}{#4}{\langle}{}{\rangle}{,}\@ifempty{#2}{}{_{\m{#2}}}}{\nameuse{functan@scalprod@named@#2}{#1}{#3}{#4}}}}
\newcommand{\newscalprod}{[1]}
\newcommand{\package}{\arrowfill@}

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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>\isempty</td>
<td>34, 40, 47, 55, 86, 178, 182</td>
</tr>
<tr>
<td>\isnotempty</td>
<td>78, 79, 81</td>
</tr>
<tr>
<td>\ifstar</td>
<td>91, 96, 110, 117, 124, 130, 137, 154, 161, 168, 174</td>
</tr>
<tr>
<td>\isundefined</td>
<td>31, 37, 41, 85, 98, 180</td>
</tr>
<tr>
<td>\nedef</td>
<td>33, 90, 94, 185</td>
</tr>
<tr>
<td>\nameuse</td>
<td>39, 43, 63, 70, 74, 80, 88, 99, 183</td>
</tr>
<tr>
<td>\arrowfill</td>
<td>143, 146, 150</td>
</tr>
<tr>
<td>\casesize</td>
<td>7, 49</td>
</tr>
<tr>
<td>\Conv</td>
<td>6, 123</td>
</tr>
<tr>
<td>\conv</td>
<td>5, 95</td>
</tr>
<tr>
<td>\Conv*</td>
<td>6</td>
</tr>
</tbody>
</table>
\conv* \hfill 5

D
\DeclareOption \hfill 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27
\delcasesize \hfill 44

E
\ext@arrow \hfill 102, 104, 106, 108, 145, 148, 152

F
\functan@arg@bar \hfill 66, 67
\functan@casesize \hfill 44, 62, 68, 72, 77
\functan@nonstarred@Conv \hfill 124, 125
\functan@nonstarred@conv \hfill 96, 97
\functan@nonstarred@Leftconv \hfill 131, 132
\functan@nonstarred@leftconv \hfill 111, 112
\functan@nonstarred@Leftrightconv \hfill 138, 139
\functan@nonstarred@leftrightconv \hfill 118, 119
\functan@nonstarred@leftrightwkconv \hfill 169, 170
\functan@nonstarred@leftwkconv \hfill 162, 163
\functan@nonstarred@norm \hfill 82, 92
\functan@nonstarred@scalprod \hfill 175, 179
\functan@nonstarred@wkconv \hfill 155, 156
\functan@norm \hfill 76, 83, 86, 87
\functan@size \hfill 45, 46, 54, 56–60
\functan@size@auto \hfill 44, 46, 56
\functan@size@Big \hfill 50, 59
\functan@size@Bigg \hfill 49, 57
\functan@size@big \hfill 52, 60
\functan@size@bigg \hfill 51, 58
\functan@starred@Conv \hfill 124, 127
\functan@starred@conv \hfill 96, 100
\functan@starred@Leftconv \hfill 130, 134
\functan@starred@leftconv \hfill 110, 114
\functan@starred@Leftrightconv \hfill 137, 141
\functan@starred@leftrightconv \hfill 117, 121
\functan@starred@leftrightwkconv \hfill 168, 172
\functan@starred@leftwkconv \hfill 161, 165
\functan@starred@norm \hfill 82, 91
\functan@starred@scalprod \hfill 174, 176
\functan@starred@wkconv \hfill 154, 158
\functan@vert@bar \hfill 82

L
\langle \hfill 177, 181
\left \hfill 62, 68, 72, 77
\Leftarrowfill@ \hfill 106
\Leftconv \hfill 6, 123
\leftconv \hfill 5, 109
\Leftconv* \hfill 6
\leftconv* \hfill 5
\leftharpoonfill@ \hfill 146, 148
\leftharpoonup \hfill 146, 150
\Leftrightarrowfill@ \hfill 108
\leftrightarrowfill@ \hfill 102
\Lefttrightconv \hfill 6, 123
\lefttrightconv \hfill 5, 109
\Lefttrightconv* \hfill 6
\lefttrightconv* \hfill 5
\leftrightharpoonfill@ \hfill 149, 152
\leftrightwkconv \hfill 5, 153
\leftrightwkconv* \hfill 5
\leftwkconv \hfill 5, 153
\leftwkconv* \hfill 5
\| \hfill 77, 79, 80

M
\m \hfill 2, 34, 87, 182
\Macro \hfill 2, 31
\mempty \hfill 34, 99, 113, 120, 126, 133, 140, 157, 164, 171

N
\newconv \hfill 5, 93
\newnorm \hfill 3, 89
\newscalprod \hfill 4, 174
\norm \hfill 3, 91
\norm* \hfill 3

O
\onearg \hfill 6, 61

P
\PackageError \hfill 35, 38, 42
\PackageWarning \hfill 32