The \texttt{amsopn} package

Michael Downes

Version 2.01, 1999/12/14

1 Introduction

The \texttt{amsopn} package provides a command \texttt{\textbackslash DeclareMathOperator} for defining new `math operator names' similar to the standard function names \texttt{\textbackslash sin}, \texttt{\textbackslash lim}, \texttt{\textbackslash max}, etc.

Standard file identification.

\texttt{\textbackslash NeedsTeXFormat\{LaTeX2e\}}\% LaTeX 2.09 can't be used (nor non-LaTeX) \% [1994/12/01]\% LaTeX date must December 1994 or later
\texttt{\textbackslash ProvidesPackage\{amsopn\}\%[1999/12/14 v2.01 operator names]}

What \texttt{\textbackslash nolimits@} does is keep a \texttt{\textbackslash limits} typed by the user from having any effect. This is used for operator names whose standard usage is never to have limits.

\texttt{\textbackslash def\textbackslash nolimits@\{\textbackslash ifnextchar\textbackslash limits\{\textbackslash nolimits\textbackslash gobble\}\{\textbackslash nolimits\}}

In operator names, it is sometimes desired to have text-mode punctuation characters such as \texttt{-/:'}. Because the body of an operator name is set in math mode, these few punctuation characters will not come out right (wrong symbol/and or wrong spacing). The purpose of \texttt{\textbackslash newcodes@} is to make them act like their normal text versions.

Where practical, we use decimal numbers to cut down main mem usage ("not needed).  

\texttt{\textbackslash begin\textbackslash group\ textbackslash catcode\textbackslash \\textbackslash ^=12\ \textbackslash gdef\textbackslash newcodes@\{\textbackslash mathcode\textbackslash \textbackslash '39\textbackslash mathcode\textbackslash \textbackslash '42\textbackslash mathcode\textbackslash \\textbackslash '613A\}}

Define \texttt{\textbackslash std@minus} for \texttt{\textbackslash relbar} use; otherwise there are problems with arrows constructed with \texttt{\textbackslash relbar}.

\texttt{\textbackslash ifnum\textbackslash mathcode\textbackslash \textbackslash '}-45\ \texttt{\textbackslash else}\ \texttt{\textbackslash mathchardef\textbackslash std@minus\textbackslash mathcode\textbackslash \textbackslash '='relax\ \textbackslash fi}\ \texttt{\textbackslash mathcode\textbackslash \textbackslash '='45\textbackslash mathcode\textbackslash \textbackslash '/47\textbackslash mathcode\textbackslash '\603A\textbackslash relax\}}

\texttt{\textbackslash end\textbackslash group}

The command \texttt{\textbackslash operatorname} prints its argument as a `math operator' like \texttt{\textbackslash sin} or \texttt{\textbackslash det}, with proper font and spacing.

\texttt{\textbackslash DeclareRobustCommand\{\textbackslash operatorname\}\{\%\}\textbackslash ifstar\{\textbf{\textbackslash qopname\textbackslash newcodes@ m}\}\{\textbf{\textbackslash qopname\textbackslash newcodes@ o}\}}}
In the interior of the \texttt{\textbackslash mathop} we need a null object (we choose a zero kern for minimum waste of main mem) in order to guard against the case where \texttt{\#3} is a single letter; \TeX will seize it and center it on the math axis if there is nothing else inside the \texttt{\textbackslash mathop} atom.

\texttt{\textbackslash DeclareRobustCommand(\textbackslash qopname)[3]\%}
\texttt{\textbackslash mathop\{	extbackslash kern\textbackslash z\textbackslash \operator\textbackslash font\textbackslash #3\textbackslash \%
\csname n\textbackslash #2\textbackslash limits\textbackslash \endcsname\%}  

\texttt{\textbackslash DeclareMathOperator}  
The command \texttt{\textbackslash DeclareMathOperator} defines the first argument to be an operator name whose text is the second argument. The star form means that the operator name should take limits (like \texttt{\textbackslash max} or \texttt{\textbackslash lim}).

\texttt{\textbackslash newcommand\{\textbackslash DeclareMathOperator\}\{\%
\textbackslash \@ifstar{\textbackslash declmathop\ m}{\textbackslash declmathop\ o}\}}  

In the basic set of operator names (below) we did not use \texttt{\textbackslash DeclareRobustCommand} because of the hash table cost. But we use it here to minimize the chances of trouble, since we are producing a user-defined command.

\texttt{\long\textbackslash def\{\textbackslash declmathop\#1\#\#3\}%}
\texttt{\textbackslash \@ifdefinable(\textbackslash #2)(%}
\texttt{\textbackslash DeclareRobustCommand(\textbackslash #2)(\textbackslash qopname\textbackslash newmcodes\textbackslash \#1(\textbackslash #3)\textbackslash \%))%
}\%  
\texttt{\textbackslash onlypreamble\textbackslash DeclarnathOperator\textbackslash onlypreamble\textbackslash declmathop\}}  

\texttt{\def\arccos{\textbackslash qopname\textbackslash relax-o(arccos)}}  
\texttt{\def\arcsin{\textbackslash qopname\textbackslash relax-o(arcsin)}}  
\texttt{\def\arctan{\textbackslash qopname\textbackslash relax-o(arctan)}}  
\texttt{\def\arg{\textbackslash qopname\textbackslash relax-o(arg)}}  
\texttt{\def\cos{\textbackslash qopname\textbackslash relax-o(cos)}}  
\texttt{\def\cosh{\textbackslash qopname\textbackslash relax-o(cosh)}}  
\texttt{\def\cot{\textbackslash qopname\textbackslash relax-o(cot)}}  
\texttt{\def\coth{\textbackslash qopname\textbackslash relax-o(coth)}}  
\texttt{\def\csc{\textbackslash qopname\textbackslash relax-o(csc)}}  
\texttt{\def\deg{\textbackslash qopname\textbackslash relax-o(deg)}}  
\texttt{\def\det{\textbackslash qopname\textbackslash relax-o(det)}}  
\texttt{\def\dim{\textbackslash qopname\textbackslash relax-o(dim)}}  
\texttt{\def\exp{\textbackslash qopname\textbackslash relax-o(exp)}}  
\texttt{\def\gcd{\textbackslash qopname\textbackslash relax-o(gcd)}}  
\texttt{\def\hom{\textbackslash qopname\textbackslash relax-o(hom)}}  
\texttt{\def\inf{\textbackslash qopname\textbackslash relax-o(inf)}}  
\texttt{\def\injlim{\textbackslash qopname\textbackslash relax-o(inj\textbackslash lim)}}  
\texttt{\def\ker{\textbackslash qopname\textbackslash relax-o(ker)}}  
\texttt{\def\lg{\textbackslash qopname\textbackslash relax-o(lg)}}  
\texttt{\def\lim{\textbackslash qopname\textbackslash relax-o(lim)}}  
\texttt{\def\liminf{\textbackslash qopname\textbackslash relax-o(lim\textbackslash inf)}}  
\texttt{\def\limsup{\textbackslash qopname\textbackslash relax-o(lim\textbackslash sup)}}  
\texttt{\def\ln{\textbackslash qopname\textbackslash relax-o(ln)}}  
\texttt{\def\log{\textbackslash qopname\textbackslash relax-o(log)}}  
\texttt{\def\max{\textbackslash qopname\textbackslash relax-o(max)}}
This command is provided to allow the document styles to decide in which way math operators like 'max' or 'log' are typeset. The default is to set them in ⟨math group⟩ zero of the current math version. The original name was \operatorfont, retained for compatibility; the second name was added to make it more accessible so that users can call this font for use in special constructs that are not ordinary operator names but conceptually related.

\def\operator@font\mathgroup\symoperators
\def\operatorfont\operator@font

For backwards compatibility we keep this old command name for the time being:

\def\operatornamewithlimits\operatorname*

These macros use \mathpalettes so that they will change size in script and scriptscript styles, though it’s hard to imagine they will ever be used there (the arrows, particularly, look bad in subscript sizes). Notice that the use of \ex@ means that the vertical spacing may not be optimal in script and scriptscript sizes. Unfortunately \TeX provides no easy way to do math mode vertical spacing that varies with current math style like mu units.

\def\varlim@#1#2{% 
  vtop\{m@th\ialign{##\cr
    \hfil$#1\operator@font lim$\hfil\cr
    \noalign{\nointerlineskip\kern1.5\ex@}#2\cr
    \noalign{\nointerlineskip\kern-\ex@}\cr}}%
\def\varinjlim{%
  \mathop{\mathpalette\varlim@{\rightarrowfill@\textstyle}}\nmlimits@
}\def\varprojlim{%
  \mathop{\mathpalette\varlim@{\leftarrowfill@\textstyle}}\nmlimits@
}\def\varliminf{%
  \mathop{\mathpalette\varliminf@{}}\nmlimits@
}\def\varlimsup{%
  \mathop{\mathpalette\varlimsup@{}}\nmlimits@
}\let\nmlimits@\displaylimits
\DeclareOption{namelimits}{\let\nmlimits@\displaylimits}
\DeclareOption{nonamelimits}{\let\nmlimits@\nolimits}
\ProcessOptions\relax

If we don’t load the \texttt{amsgen} package then the use of \texttt{ex} in \texttt{varlim} will lead to trouble.

\RequirePackage{amsGen}\relax

The usual \texttt{endinput} to ensure that random garbage at the end of the file doesn’t get copied by docstrip.

\endinput