The flexisym package

Authors: Michael J. Downes, Morten Høgholm
Maintained by Morten Høgholm, Will Robertson
Feedback: https://github.com/wspr/breqn/issues

2021/10/28 0.98l

User's guide

For now, the user's guide is in breqn.

Implementation

1 flexisym

```
⟨*package⟩
\RequirePackage{expl3}[2009/08/05]
\ProvidesExplPackage{flexisym}{2021/10/28}{0.98l}{Make math characters macros}

\edef\do{%
  \noexpand\AtEndOfPackage{%
    \catcode\number`\"=\number\catcode`\"
    \relax
  }%
}
\do \let\do\relax
\catcode`\"=12
\let\@sym\@gobble
\DeclareOption{robust}{%
  \def\@sym#1{%
    \ifx\protect\@typeset@protect \else\protect#1\exp_after:wN\use_none:nnnn\fi
  }%
}
```

The math groups (mg) here relate to \textfont$n$.

```
\def\mg@bin{2}% binary operators
\def\mg@rel{2}% relations
%%\def\mg@nre{B}% negated relations
\def\mg@del{3}% delimiters
```

```
%%\def\mg@arr{B}% arrows
\def\mg@acc{0}% accents
\def\mg@cop{3}% cumulative operators (sum, int)
\def\mg@latin{1}% (Latin) letters
\def\mg@greek{1}% (lowercase) Greek
\def\mg@Greek{0}% (capital) Greek
%%\def\mg@bflatin{4}% bold upright Latin letters ?
%%\def\mg@Bbb{B}% blackboard bold
\def\mg@cal{2}% script/calligraphic
%%\def\mg@frak{5}% Fraktur letters
\def\mg@digit{0}% decimal digits % 1 = oldstyle, 0 = capital
```

This is how we insert mathchars. The command has three arguments: class, fam and slot postion and so it is always given as hexadecimal. This way of separating things should make it easier to get this to work with XeTeX et al. which have many more slot positions

```
\cs_set_protected:Nn \math_char:NNn {
  \tex_mathchar:D \__int_eval:w " #1#2#3 \__int_eval_end:
}
```

Delimiters and radicals are similar except here we have both small and large variant. Radicals have no class.

```
\cs_set_protected:Nn \math_delimiter:NNnNn {
  \tex_delimiter:D \__int_eval:w " #1#2#3#4#5 \__int_eval_end:
}
\cs_set_protected:Nn \math_radical:NnNn {
  \tex_radical:D  \__int_eval:w " #1#2#3#4 \__int_eval_end:
}
\cs_set_protected:Nn \math_accent:NNnn {
  \tex_mathaccent:D \__int_eval:w " #1 #2 #3 \__int_eval_end: {#4}
}

\let\sumlimits\displaylimits
\let\intlimits\nolimits
\let\namelimits\displaylimits
```

TEX defines eight types of atoms.

0. Ordinary
1. Operators
2. Binary
3. Relation
4. Open
5. Close
6. Punctuation
7. Inner

TEX defines eight math classes.

0. Ordinary
1. Operators
2. Binary
3. Relation
4. Open
5. Close
6. Punctuation
7. Variable family

flexisym/breqn extends this to types of classes.

0. Ordinary: (Ord), Bidirectional delimiters (DeB), Radicals (Rad), Accented items (Acc)
1. Operators: Cumulative Operators sum-like (COs), Cumulative Operators integral-like (COi)
2. Binary: (Bin)
3. Relation: (Rel), Arrow delimiters (DeA)
4. Open: (DeL)
5. Close (DeR)
6. Punctuation: (Pun)
7. Variable family: (Var)

Here's an overview of what we are about to do. Math chars of each type as defined by us need a basic operation for inserting it. We will call that function `\math_bsym_`⟨*type*⟩`:Nn`. Next there are compund symbols for each type which we name `\math_bcsym_`⟨*type*⟩`:Nn`. Also, there is inline mode and display mode which are different. We will call them for `\math_isym_`⟨*type*⟩`:Nn` `\math_icsym_`⟨*type*⟩`:Nn` for inline mode and `\math_dsym_`⟨*type*⟩`:Nn` and `\math_dcsym_`⟨*type*⟩`:Nn`. The code uses the terms `\math_sym_`⟨*type*⟩`:Nn` and `\math_csym_`⟨*type*⟩`:Nn` for the current meaning of things. First up the basic definitions. `#1` is the math group it is from and `#2` is the slot position.

```
\cs_new:Npn \math_bsym_Ord:Nn {\math_char:NNn 0 }% \m@Ord
\cs_new:Npn \math_bsym_Var:Nn {\math_char:NNn 7 }% \m@Var
\cs_new:Npn \math_bsym_Bin:Nn {\math_char:NNn 2 }% \m@Bin
\cs_new:Npn \math_bsym_Rel:Nn {\math_char:NNn 3 }% \m@Bin
\cs_new:Npn \math_bsym_Pun:Nn {\math_char:NNn 6 }% \m@Pun
```

```
\cs_new:Nn  \math_bsym_COs:Nn { \math_char:NNn 1 #1 {#2} \sumlimits }% \m@COs
\cs_new:Nn  \math_bsym_COi:Nn { \math_char:NNn 1 #1 {#2} \intlimits }% \m@COi
\cs_new:Nn  \math_bsym_DeL:Nn { \math_sd_del_aux:Nnn 4 #1{#2} }% \m@DeL
\cs_new:Nn  \math_bsym_DeR:Nn { \math_sd_del_aux:Nnn 5 #1{#2} }% \m@DeR
\cs_new:Nn  \math_bsym_DeB:Nn { \math_sd_del_aux:Nnn 0 #1{#2} }% \m@DeB
\cs_new:Nn  \math_bsym_DeA:Nn { \math_sd_del_aux:Nnn 3 #1{#2} }% \m@DeA
\cs_new:Nn  \math_bsym_Rad:Nn { \math_sd_rad_aux:Nn #1{#2} }% \m@Rad
\cs_new:Npn \math_bsym_Acc:Nn #1#2#3#4 {\math_accent:NNnn #1#2{#3}{#4}}% name is wrong

```

Next is somewhat complicated internally. The way it is done is that delimiters and radicals need information about the smallest version of the symbol. If this smallest delimiter (SD) is defined, then use it. We have these functions to help us return the number. Extract the numbers to use and stick a function in front of it.

Code changed because now we require the smallest delimiter to be defined (it may be the same, no problem in that). So the two arguments present in `\math_bsym_DeL:Nn` are the location of extensible version (where the font will do the rest for us automatically). For each delimiter, a pointer is defined using the extensible characters family and slot as name and value equal to family and position of the smallest version. For `(` in standard LaTeX this is `{del}{00}` and `{OT1}{28}` respectively. Hence, `\math_bsym_DeL:Nn \mg@del {00}` must expand to `\math_delimiter:NNnNn 4 \mg@OT1 {28}\mg@del{00}`. So first expand away to get to the smallest version. Then call next function which shuffles the arguments around.

```
\cs_set:Npn \math_sd_del_aux:Nnn #1#2#3{
  \exp_args:Nf \math_sd_del_auxi:nN {\use:c{sd@#2#3}} #1 #2{#3}
}
\cs_set:Npn \math_sd_del_auxi:nN #1#2{ \math_delimiter:NNnNn #2 #1 }
```

Same for radicals.

```
\cs_set:Npn \math_sd_rad_aux:Nn #1#2{
  \exp_args:Nf \math_sd_rad_auxi:n {\use:c{sd@#1#2}} #1 {#2}
}
\cs_set:Npn \math_sd_rad_auxi:n #1{ \math_radical:NnNn #1 }


% \cs_set:Npn \math_sd_aux:nn  #1#2 {
%   %\exp_args:Nnf \use:nn { #1}  { \math_sd_auxi:Nn #2 }
%   \exp_args:Nnf \use:nn { #1}  { \use:c{sd@\use:nn#2} }
% }
% \cs_set:Npn \math_sd_auxi:Nn #1#2 {
%    \cs_if_free:cTF {sd@#1#2}
%   {   #1{#2}  }
%   {   \use:c{sd@#1#2}  }
% }
```

compound symbols here

```
\cs_set_protected:Npn \math_bcsym_Ord:Nn #1#2 { \@symtype \mathord   { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_Var:Nn #1#2 { \@symtype \mathord   { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_Bin:Nn #1#2 { \@symtype \mathbin   { \OrdSymbol {#2} } }%\
```

```
\cs_set_protected:Npn \math_bcsym_Rel:Nn #1#2 { \@symtype \mathrel   { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_Pun:Nn #1#2 { \@symtype \mathpunct { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_COi:Nn #1#2 { \@symtype \mathop    { \OrdSymbol {#2} \intl
\cs_set_protected:Npn \math_bcsym_COs:Nn #1#2 { \@symtype \mathop    { \OrdSymbol {#2} \suml
\cs_set_protected:Npn \math_bcsym_DeL:Nn #1#2 { \@symtype \mathopen  { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_DeR:Nn #1#2 { \@symtype \mathclose { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_DeB:Nn #1#2 { \@symtype \mathord   { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_DeA:Nn #1#2 { \@symtype \mathrel   { \OrdSymbol {#2} } }%\
\cs_set_protected:Npn \math_bcsym_Acc:Nn {\@sym}%\@symAcc FIX!
% These three?
\cs_set_protected:Npn \math_bcsym_Ope:Nn #1#2{\@symtype\mathopen{\OrdSymbol{#2}}}%\@symVar
\cs_set_protected:Npn \math_bcsym_Clo:Nn #1#2{\@symtype\mathclose{\OrdSymbol{#2}}}%\@symVar
\cs_set_protected:Npn \math_bcsym_Inn:Nn #1#2{\@symtype\mathinner{\OrdSymbol{#2}}}%\@symVar

\let\@symtype\@firstofone
\let\sym@global\global
```

The inline variants, using the basic operations. Currently we do not do anything to inline math.

```
\cs_new:Npn \math_isym_Ord:Nn { \math_bsym_Ord:Nn }% \m@Ord
\cs_new:Npn \math_isym_Var:Nn { \math_bsym_Var:Nn }% \m@Var
\cs_new:Npn \math_isym_Bin:Nn { \math_bsym_Bin:Nn }% \m@Bin
\cs_new:Npn \math_isym_Rel:Nn { \math_bsym_Rel:Nn }% \m@Bin
\cs_new:Npn \math_isym_Pun:Nn { \math_bsym_Pun:Nn }% \m@Pun
\cs_new:Npn \math_isym_COs:Nn { \math_bsym_COs:Nn }% \m@COs
\cs_new:Npn \math_isym_COi:Nn { \math_bsym_COi:Nn }% \m@COi
\cs_new:Npn \math_isym_DeL:Nn { \math_bsym_DeL:Nn }% \m@DeL
\cs_new:Npn \math_isym_DeR:Nn { \math_bsym_DeR:Nn }% \m@DeR
\cs_new:Npn \math_isym_DeB:Nn { \math_bsym_DeB:Nn }% \m@DeB
\cs_new:Npn \math_isym_DeA:Nn { \math_bsym_DeA:Nn }% \m@DeA
\cs_new:Npn \math_isym_Rad:Nn { \math_bsym_Rad:Nn }% \m@Rad
\cs_new:Npn \math_isym_Acc:Nn { \math_bsym_DeL:Nn }% name is wrong
% inline compound
\cs_set_protected:Npn \math_icsym_Ord:Nn { \math_bcsym_Ord:Nn }
\cs_set_protected:Npn \math_icsym_Var:Nn { \math_bcsym_Var:Nn }
\cs_set_protected:Npn \math_icsym_Bin:Nn { \math_bcsym_Bin:Nn }
\cs_set_protected:Npn \math_icsym_Rel:Nn { \math_bcsym_Rel:Nn }
\cs_set_protected:Npn \math_icsym_Pun:Nn { \math_bcsym_Pun:Nn }
\cs_set_protected:Npn \math_icsym_COi:Nn { \math_bcsym_COi:Nn }
\cs_set_protected:Npn \math_icsym_COs:Nn { \math_bcsym_COs:Nn }
\cs_set_protected:Npn \math_icsym_DeL:Nn { \math_bcsym_DeL:Nn }
\cs_set_protected:Npn \math_icsym_DeR:Nn { \math_bcsym_DeR:Nn }
\cs_set_protected:Npn \math_icsym_DeB:Nn { \math_bcsym_DeB:Nn }
\cs_set_protected:Npn \math_icsym_DeA:Nn { \math_bcsym_DeA:Nn }
\cs_set_protected:Npn \math_icsym_Acc:Nn { \math_bcsym_Acc:Nn }
\cs_set_protected:Npn \math_icsym_Ope:Nn { \math_bcsym_Ope:Nn }
\cs_set_protected:Npn \math_icsym_Clo:Nn { \math_bcsym_Clo:Nn }
\cs_set_protected:Npn \math_icsym_Inn:Nn { \math_bcsym_Inn:Nn }
```

The display variants, using the basic operations. Currently we do not do

anything to inline math.

```
\cs_new:Npn \math_dsym_Ord:Nn { \math_bsym_Ord:Nn }
\cs_new:Npn \math_dsym_Var:Nn { \math_bsym_Var:Nn }
\cs_new:Npn \math_dsym_Bin:Nn { \math_bsym_Bin:Nn }
\cs_new:Npn \math_dsym_Rel:Nn { \math_bsym_Rel:Nn }
\cs_new:Npn \math_dsym_Pun:Nn { \math_bsym_Pun:Nn }
\cs_new:Npn \math_dsym_COs:Nn { \math_bsym_COs:Nn }
\cs_new:Npn \math_dsym_COi:Nn { \math_bsym_COi:Nn }
\cs_new:Npn \math_dsym_DeL:Nn { \math_bsym_DeL:Nn }
\cs_new:Npn \math_dsym_DeR:Nn { \math_bsym_DeR:Nn }
\cs_new:Npn \math_dsym_DeB:Nn { \math_bsym_DeB:Nn }
\cs_new:Npn \math_dsym_DeA:Nn { \math_bsym_DeA:Nn }
\cs_new:Npn \math_dsym_Rad:Nn { \math_bsym_Rad:Nn }
\cs_new:Npn \math_dsym_Acc:Nn { \math_bsym_DeL:Nn }
% inline compound
\cs_set_protected:Npn \math_dcsym_Ord:Nn { \math_bcsym_Ord:Nn }
\cs_set_protected:Npn \math_dcsym_Var:Nn { \math_bcsym_Var:Nn }
\cs_set_protected:Npn \math_dcsym_Bin:Nn { \math_bcsym_Bin:Nn }
\cs_set_protected:Npn \math_dcsym_Rel:Nn { \math_bcsym_Rel:Nn }
\cs_set_protected:Npn \math_dcsym_Pun:Nn { \math_bcsym_Pun:Nn }
\cs_set_protected:Npn \math_dcsym_COi:Nn { \math_bcsym_COi:Nn }
\cs_set_protected:Npn \math_dcsym_COs:Nn { \math_bcsym_COs:Nn }
\cs_set_protected:Npn \math_dcsym_DeL:Nn { \math_bcsym_DeL:Nn }
\cs_set_protected:Npn \math_dcsym_DeR:Nn { \math_bcsym_DeR:Nn }
\cs_set_protected:Npn \math_dcsym_DeB:Nn { \math_bcsym_DeB:Nn }
\cs_set_protected:Npn \math_dcsym_DeA:Nn { \math_bcsym_DeA:Nn }
\cs_set_protected:Npn \math_dcsym_Acc:Nn { \math_bcsym_Acc:Nn }
\cs_set_protected:Npn \math_dcsym_Ope:Nn { \math_bcsym_Ope:Nn }
\cs_set_protected:Npn \math_dcsym_Clo:Nn { \math_bcsym_Clo:Nn }
\cs_set_protected:Npn \math_dcsym_Inn:Nn { \math_bcsym_Inn:Nn }
```

Almost ready now! Now just need two commands to initialize these settings.

```
\cs_set:Npn \math_setup_inline_symbols:
 {
  \cs_set_eq:NN \math_sym_Ord:Nn  \math_isym_Ord:Nn
  \cs_set_eq:NN \math_sym_Var:Nn  \math_isym_Var:Nn
  \cs_set_eq:NN \math_sym_Bin:Nn  \math_isym_Bin:Nn
  \cs_set_eq:NN \math_sym_Rel:Nn  \math_isym_Rel:Nn
  \cs_set_eq:NN \math_sym_Pun:Nn  \math_isym_Pun:Nn
  \cs_set_eq:NN \math_sym_COs:Nn  \math_isym_COs:Nn
  \cs_set_eq:NN \math_sym_COi:Nn  \math_isym_COi:Nn
  \cs_set_eq:NN \math_sym_DeL:Nn  \math_isym_DeL:Nn
  \cs_set_eq:NN \math_sym_DeR:Nn  \math_isym_DeR:Nn
  \cs_set_eq:NN \math_sym_DeB:Nn  \math_isym_DeL:Nn
  \cs_set_eq:NN \math_sym_DeA:Nn  \math_isym_DeA:Nn
  \cs_set_eq:NN \math_sym_Rad:Nn  \math_isym_Rad:Nn
  \cs_set_eq:NN \math_sym_Acc:Nn  \math_isym_DeL:Nn
  \cs_set_eq:NN \math_csym_Ord:Nn \math_icsym_Ord:Nn
  \cs_set_eq:NN \math_csym_Var:Nn \math_icsym_Var:Nn
  \cs_set_eq:NN \math_csym_Bin:Nn \math_icsym_Bin:Nn
```

```
  \cs_set_eq:NN \math_csym_Rel:Nn \math_icsym_Rel:Nn
  \cs_set_eq:NN \math_csym_Pun:Nn \math_icsym_Pun:Nn
  \cs_set_eq:NN \math_csym_COi:Nn \math_icsym_COi:Nn
  \cs_set_eq:NN \math_csym_COs:Nn \math_icsym_COs:Nn
  \cs_set_eq:NN \math_csym_DeL:Nn \math_icsym_DeL:Nn
  \cs_set_eq:NN \math_csym_DeR:Nn \math_icsym_DeR:Nn
  \cs_set_eq:NN \math_csym_DeB:Nn \math_icsym_DeB:Nn
  \cs_set_eq:NN \math_csym_DeA:Nn \math_icsym_DeA:Nn
  \cs_set_eq:NN \math_csym_Acc:Nn \math_icsym_Acc:Nn
  \cs_set_eq:NN \math_csym_Ope:Nn \math_icsym_Ope:Nn
  \cs_set_eq:NN \math_csym_Clo:Nn \math_icsym_Clo:Nn
  \cs_set_eq:NN \math_csym_Inn:Nn \math_icsym_Inn:Nn
 }

\cs_set:Npn \math_setup_display_symbols:
 {
  \cs_set_eq:NN \math_sym_Ord:Nn  \math_dsym_Ord:Nn
  \cs_set_eq:NN \math_sym_Var:Nn  \math_dsym_Var:Nn
  \cs_set_eq:NN \math_sym_Bin:Nn  \math_dsym_Bin:Nn
  \cs_set_eq:NN \math_sym_Rel:Nn  \math_dsym_Rel:Nn
  \cs_set_eq:NN \math_sym_Pun:Nn  \math_dsym_Pun:Nn
  \cs_set_eq:NN \math_sym_COs:Nn  \math_dsym_COs:Nn
  \cs_set_eq:NN \math_sym_COi:Nn  \math_dsym_COi:Nn
  \cs_set_eq:NN \math_sym_DeL:Nn  \math_dsym_DeL:Nn
  \cs_set_eq:NN \math_sym_DeR:Nn  \math_dsym_DeR:Nn
  \cs_set_eq:NN \math_sym_DeB:Nn  \math_dsym_DeL:Nn
  \cs_set_eq:NN \math_sym_DeA:Nn  \math_dsym_DeA:Nn
  \cs_set_eq:NN \math_sym_Rad:Nn  \math_dsym_Rad:Nn
  \cs_set_eq:NN \math_sym_Acc:Nn  \math_dsym_DeL:Nn
  \cs_set_eq:NN \math_csym_Ord:Nn \math_dcsym_Ord:Nn
  \cs_set_eq:NN \math_csym_Var:Nn \math_dcsym_Var:Nn
  \cs_set_eq:NN \math_csym_Bin:Nn \math_dcsym_Bin:Nn
  \cs_set_eq:NN \math_csym_Rel:Nn \math_dcsym_Rel:Nn
  \cs_set_eq:NN \math_csym_Pun:Nn \math_dcsym_Pun:Nn
  \cs_set_eq:NN \math_csym_COi:Nn \math_dcsym_COi:Nn
  \cs_set_eq:NN \math_csym_COs:Nn \math_dcsym_COs:Nn
  \cs_set_eq:NN \math_csym_DeL:Nn \math_dcsym_DeL:Nn
  \cs_set_eq:NN \math_csym_DeR:Nn \math_dcsym_DeR:Nn
  \cs_set_eq:NN \math_csym_DeB:Nn \math_dcsym_DeB:Nn
  \cs_set_eq:NN \math_csym_DeA:Nn \math_dcsym_DeA:Nn
  \cs_set_eq:NN \math_csym_Acc:Nn \math_dcsym_Acc:Nn
  \cs_set_eq:NN \math_csym_Ope:Nn \math_dcsym_Ope:Nn
  \cs_set_eq:NN \math_csym_Clo:Nn \math_dcsym_Clo:Nn
  \cs_set_eq:NN \math_csym_Inn:Nn \math_dcsym_Inn:Nn
 }
```

Phew, that was it.

Well, almost. We need to set them up for use properly. Should they be added to `\everymath`? Probably, for math within displays. However, this is a lot of extra processing which we could tackle in the display setup.

```
\math_setup_inline_symbols:
```

Need an active character for a second. Don't rely on ~ being active!

```
\edef\tmp{\catcode\z@=\the\catcode\z@}
\catcode\z@=\active
\def\DeclareFlexSymbol#1#2#3#4{%
  \begingroup
  \cs_set_protected:Npx\@tempb{
    \exp_not:N\@sym\exp_not:N#1\exp_not:c{math_sym_#2:Nn}
        \exp_not:c{mg@#3}{#4}
  }
  \ifcat\exp_not:N#1\relax
    \sym@global\let#1\@tempb
  \else
    \sym@global\mathcode`#1="8000\relax
    \lccode\z@=`#1\relax
    \lowercase{\sym@global\let^^@\@tempb}% zero char
  \fi
  \endgroup
}
\tmp % restore catcode
\cs_set:Npn \DeclareFlexDelimiter #1#2#3#4#5#6{
  \DeclareFlexSymbol{#1}{#2}{#3}{#4}
  \cs_gset:cpx{sd@\use:c{mg@#3}#4}{\exp_not:c{mg@#5}{#6}}
}

\DeclareFlexCompoundSymbol{\cdots}{Inn}{\cdotp\cdotp\cdotp} \def\@symInn#1#2{\@symtyp
\@symtype \mathinner{\OrdSymbol{\cdtop\cdotp\cdotp}}
\def\DeclareFlexCompoundSymbol#1#2#3{%
  \exp_args:NNo \DeclareRobustCommand#1{\csname math_csym_#2:Nn\endcsname#1{#3}}%
  \sym@global\let#1#1\relax
}
\DeclareRobustCommand\textchar{\text@char\textfont}
\DeclareRobustCommand\scriptchar{\text@char\scriptfont}%
```

Simplified the next bit because now the slot is read as one argument so no afterassignment and what have you. Just drop the char directly.

```
\def\text@char@sym#1#2#3#4{% #3=fam, #4=slot
  \begingroup
    \cs_set_eq:NN \@sym \prg_do_nothing: % defense against infinite loops
```

the next line will result in `\scriptfont`⟨*num*⟩, where `#3` provides the ⟨*num*⟩.

```
   \the\text@script@char#3%
   \char"#4\endgroup
}
\edef\tmp{\catcode\z@=\the\catcode\z@}
\catcode\z@=\active
\def\text@char#1#2{\begingroup
  \check@mathfonts
  \cs_set_eq:NN \text@script@char #1
```

```
  \cs_set_eq:NN \@sym \text@char@sym
  \cs_set_eq:NN \@symtype \use_ii:nn
  \cs_set_eq:NN \OrdSymbol \use:n
  \cs_set_eq:NN \ifmmode \iftrue
  \everymath{$\use_none:n}%$
  \def\mkern{\muskip\z@}
  \cs_set_eq:NN\mskip\mkern
  \ifcat\relax\noexpand#2% true if #2 is a cs.
    #2%
  \else
    \lccode\z@=\expandafter`\string#2\relax
    \lowercase{^^@}%
  \fi
  \endgroup
}
\tmp % restore catcode
\providecommand\textprime{}
\DeclareRobustCommand\textprime{\leavevmode
  \raise.8ex\hbox{\text@char\scriptfont\prime}%
}
\@ifundefined{resetMathstrut@}{}{%
  \def\resetMathstrut@{%
    \setbox\z@\hbox{\textchar\vert}%
    \ht\Mathstrutbox@\ht\z@ \dp\Mathstrutbox@\dp\z@
  }%
}
```

Arrow fills. changed to 7mu as in amsmath

```
\@ifundefined{rightarrowfill@}{}{%
  \def\rightarrowfill@#1{\m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\copy\z@\mkern-7mu\cleaders
    \hbox{$#1\mkern-2mu\box\z@\mkern-2mu$}\hfill
    \mkern-6mu\OrdSymbol{\rightarrow}$}
  \def\leftarrowfill@#1{\m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\OrdSymbol{\leftarrow}\mkern-6mu\cleaders
    \hbox{$#1\mkern-2mu\copy\z@\mkern-2mu$}\hfill
    \mkern-7mu\box\z@$}
  \def\leftrightarrowfill@#1{\m@th\setboxz@h{$#1\relbar$}\ht\z@\z@
    $#1\OrdSymbol{\leftarrow}\mkern-6mu\cleaders
    \hbox{$#1\mkern-2mu\box\z@\mkern-2mu$}\hfill
    \mkern-6mu\OrdSymbol{\rightarrow}$}
}
```

hey, this looks like a simple case switch...

```
\def\binrel@sym#1#2#3#4{%
  \xdef\binrel@@##1{%
    \ifx\math_sym_Ord:Nn #2 \math_csym_Ord:Nn
    \else\ifx\math_sym_Var:Nn#2 \math_csym_Var:Nn
    \else\ifx\math_sym_COs:Nn#2 \math_csym_COs:Nn
    \else\ifx\math_sym_COi:Nn#2 \math_csym_COi:Nn
    \else\ifx\math_sym_Bin:Nn#2 \math_csym_Bin:Nn
```

```
    \else\ifx\math_sym_Rel:Nn#2 \math_csym_Rel:Nn
    \else\ifx\math_sym_Pun:Nn#2 \math_csym_Pun:Nn
    \else\exp_not:N\@symErr \fi\fi\fi\fi\fi\fi\fi
  ?{\exp_not:N\OrdSymbol{##1}}}%
}

\def\binrel@a{%
  \def\math_sym_Ord:Nn##1##2{\gdef\binrel@@####1{\math_sym_Ord:Nn##1{\OrdSymbol{####1}}}}%
  \def\math_sym_Var:Nn##1##2{\gdef\binrel@@####1{\math_sym_Var:Nn##1{\OrdSymbol{####1}}}}%
  \def\math_sym_COs:Nn##1##2{\gdef\binrel@@####1{\math_sym_COs:Nn##1{\OrdSymbol{####1}}}}%
  \def\math_sym_COi:Nn##1##2{\gdef\binrel@@####1{\math_sym_COi:Nn##1{\OrdSymbol{####1}}}}%
  \def\math_sym_Bin:Nn##1##2{\gdef\binrel@@####1{\math_sym_Bin:Nn##1{\OrdSymbol{####1}}}}%
  \def\math_sym_Rel:Nn##1##2{\gdef\binrel@@####1{\math_sym_Rel:Nn##1{\OrdSymbol{####1}}}}%
  \def\math_sym_Pun:Nn##1##2{\gdef\binrel@@####1{\math_sym_Pun:Nn##1{\OrdSymbol{####1}}}}%
}
\def\binrel@#1{%
  \setbox\z@\hbox{$%
    \let\mathchoice\@gobblethree
    \let\@sym\binrel@sym \binrel@a
    #1$}%
}
\def\@symextension{sym}
\newcommand\usesymbols[1]{%
  \clist_map_variable:nNn{#1}\@tempb{%
    \exp_args:No\@onefilewithoptions{\@tempb}[][]\@symextension
  }%
}
% Need to introduce \ProvidesExplFile somehow
\newcommand\ProvidesSymbols[1]{\ProvidesFile{#1.sym}}
\DeclareRobustCommand{\not}[1]{\math_csym_Rel:Nn\not{\OrdSymbol{\notRel#1}}}
\DeclareRobustCommand{\OrdSymbol}[1]{%
  \begingroup\mathchars@reset#1\endgroup
}
\def\mathchars@reset{\let\@sym\@sym@ord \let\@symtype\@symtype@ord
  \let\OrdSymbol\relax}
\def\@symtype@ord#1#{}% a strange sort of \@gobble
\def\@sym@ord#1#2{\exp_after:wN\@sym@ord@a\string#2\@nil}%
```

Read delimited argument here. We want to find first character of DeA, Bin, etc. and the control sequence checked agains is `\m@DeL`, `\m@Pun`, etc. The lccode trick makes the . into an @ with catcode 12. This is what results when the code is called with `\string`. Beware of this when we change internal names for math groups! If a Delimiter is found, insert it with class 0 but use the smallest version available. Otherwise just insert math char of class 0. The code here is not pretty and it indicates it should be tackled differently!

```
\begingroup
\lccode`\.=`\_ \lowercase{\endgroup
\def\@sym@ord@a#1.#2.}#3#4\@nil#5#6{%
  \if D#3
     %\math_ord_delim_aux:Nn #5{#6}
```

```
    \math_sd_del_aux:Nnn 0 #5{#6}% check if this works!
  \else
    \math_char:NNn 0 #5{#6}
  \fi
}
\cs_set:Nn \math_ord_delim_aux:Nn {
  \math_sd_aux:nn { \math_char:NNn 0 } {#1{#2}}
}
```

Before declaring any math characters active, we have to take care of a small problem with amsmath v2.x, if it is loaded before flexisym. `\std@minus` and `\std@equal` are defined as

```
\mathchardef\std@minus\mathcode`\-\relax
\mathchardef\std@equal\mathcode`\=\relax
```

in `amsmath.sty` and again `\AtBeginDocument`. The latter is because

> In case some alternative math fonts are loaded later. [`amsmath.dtx`]

The problem arises because flexisym sets the mathcode of all symbols to 32768 which is illegal for a `\mathchardef`.

We have to remove the assignments from the `\AtBeginDocument` hook as they will cause an error there.

```
\@ifpackageloaded{amsmath}{%
```

Split the contents of `\@begindocumenthook` by reading what we search for as a delimited argument and ensure these two assignments do not take place. It is questionable if anything reasonable can be done to them. In the case of a package such as mathpazo which defines

```
\DeclareMathSymbol{=}{\mathrel}{upright}{"3D}
```

the `\Relbar` will look wrong if we don't use the correct symbol. The way to solve this is define additional `.sym` files which contain the definition of `\relbar` and `\Relbar` needed. We need those additional files anyway for things like `\joinord`.

```
}{}
```

There is problem when using `\DeclareMathOperator` as the operators defined call a command `\newmcodes@` which relies on the mathcode of `-` being less than 32768. We delay the definition `\AtBeginDocument` in case amssymb hasn't been loaded yet.

```
\AtBeginDocument{%
\ifx\Umathcode\@undefined
\gdef\newmcodes@{\mathcode`\'39\mathcode`\*42\mathcode`\."613A%
  \ifnum\mathcode`\-=45 \else
```

The extra check. Don't do anything if `-` is math active.

```
    \ifnum\mathcode`\-=32768\space
    \else
```

```
      \mathchardef\std@minus\mathcode`\-\relax
    \fi
  \fi
  \mathcode`\-45\mathcode`\/47\mathcode`\:"603A\relax}
\else
\gdef\newmcodes@{\mathcode`\'39\mathcode`\*42\mathcode`\."613A%
  \ifnum\Umathcodenum`\-=45 \else
```

The extra check. Don't do anything if - is math active.

```
    \ifnum\Umathcodenum`\-=16777216\space
    \else
      \Umathcharnumdef\std@minus\Umathcodenum`\-\relax
    \fi
  \fi
  \mathcode`\-45\mathcode`\/47\mathcode`\:"603A\relax}
\fi
}
```

And we then continue with the options.

```
\DeclareOption{mathstyleoff}{%
  \PassOptionsToPackage{noactivechars}{mathstyle}}
\DeclareOption{cmbase}{\usesymbols{cmbase}}
\DeclareOption{mathpazo}{\usesymbols{mathpazo}}
\DeclareOption{mathptmx}{\usesymbols{mathptmx}}
\ExecuteOptions{cmbase}
\ProcessOptions\relax
\renewcommand{\lnot}{\neg}
\renewcommand{\land}{\wedge}
\renewcommand{\lor}{\vee}
\renewcommand{\le}{\leq}
\renewcommand{\ge}{\geq}
\renewcommand{\ne}{\neq}
\renewcommand{\owns}{\ni}
\renewcommand{\gets}{\leftarrow}
\renewcommand{\to}{\rightarrow}
\renewcommand{\|}{\Vert}
\RequirePackage{mathstyle}
⟨/package⟩\endinput
```

2 cmbase, mathpazo, mathptmx

For each math font package we define a corresponding symbol file with extension `sym`. The Computer Modern base is called `cmbase` and `mathpazo` and `mathptmx` corresponds to the packages. The definitions are almost identical as they mostly concern the positions in the math font encodings. Look for differences in `\joinord`, `\relbar` and `\Relbar`. If you inspect the source code, you'll see that the support for mathptmx didn't require any work but I thought it better to create a `sym` file to maintain a uniform interface.

Open question on ! and ?: maybe they should have type 'Pun' instead of 'DeR'. Need to search for uses in math in AMS archives. Or, maybe add a special 'Clo' type for them: non-extensible closing delimiter.

Default mathgroup setup.

```
⟨*cmbase | mathpazo | mathptmx⟩
⟨cmbase⟩\ProvidesSymbols{cmbase}[2007/12/19 v0.92]
⟨mathpazo⟩\ProvidesSymbols{mathpazo}[2010/07/11 v0.3]
⟨mathptmx⟩\ProvidesSymbols{mathptmx}[2010/07/11 v0.3]
\ExplSyntaxOn
\cs_gset:cpx {mg@OT1}  {\hexnumber@\symoperators}
\cs_gset:cpx {mg@OML}  {\hexnumber@\symletters}
\cs_gset:cpx {mg@OMS}  {\hexnumber@\symsymbols}
\cs_gset:cpx {mg@OMX}  {\hexnumber@\symlargesymbols}
\cs_gset:Npx \mg@bin   {\mg@OMS}
\cs_gset:Npx \mg@del   {\mg@OMX}
\cs_gset:Npx \mg@digit {\exp_not:c{mg@OT1}}
\cs_gset:Npn \mg@latin {\mg@OML}
\cs_gset_eq:NN \mg@Latin \mg@latin
\cs_gset_eq:NN \mg@greek \mg@latin
⟨cmbase | mathptmx⟩\cs_gset_eq:NN\mg@Greek\mg@digit
```

Mathpazo takes the upper case greeks from the letter font if slantedGreek is in effect, but from *upright* if not. Mathptmx also takes the slanted greek from the letter font.

```
⟨mathpazo⟩\@ifpackagewith{mathpazo}{slantedGreek}{%
⟨mathpazo⟩   \cs_gset_eq:NN\mg@Greek\mg@latin
⟨mathpazo⟩}{%
⟨mathpazo⟩   \cs_gset:cpx{mg@Greek}{\hexnumber@\symupright}
⟨mathpazo⟩}
⟨mathptmx⟩\@ifpackagewith{mathptmx}{slantedGreek}{%
⟨mathptmx⟩    \cs_gset_eq:NN\mg@Greek\mg@latin
⟨mathptmx⟩}{}
\cs_gset_eq:NN \mg@rel \mg@bin
\cs_gset_eq:NN \mg@ord \mg@bin
\cs_gset_eq:NN \mg@cop \mg@del
```

Symbols from the 128-character `cmr` encoding. Paren and square bracket delimiters from this encoding are covered by the definitions in the `cmex` section, however.

```
\DeclareFlexSymbol{!}     {Pun}{OT1}{21}
\DeclareFlexSymbol{+}     {Bin}{OT1}{2B}
\DeclareFlexSymbol{:}     {Rel}{OT1}{3A}
\DeclareFlexSymbol{\colon}{Pun}{OT1}{3A}
\DeclareFlexSymbol{;}     {Pun}{OT1}{3B}
\AtBeginDocument{\DeclareFlexSymbol{=}     {Rel}{OT1}{3D}}
\DeclareFlexSymbol{?}     {Pun}{OT1}{3F}
```

$\mathcal{AMS}$TEX, and therefore the amsmath package, make the uppercase Greek letters class 0 (nonvariable) instead of 7 (variable), to eliminate the glaring inconsistency

with lowercase Greek. (In plain TEX, `{\bf\Delta}` works, while `{\bf\delta}` doesn't.) Let us try to make them both variable (fonts permitting) instead of nonvariable.

```
\DeclareFlexSymbol{\Gamma}  {Var}{Greek}{00}
\DeclareFlexSymbol{\Delta}  {Var}{Greek}{01}
\DeclareFlexSymbol{\Theta}  {Var}{Greek}{02}
\DeclareFlexSymbol{\Lambda} {Var}{Greek}{03}
\DeclareFlexSymbol{\Xi}     {Var}{Greek}{04}
\DeclareFlexSymbol{\Pi}     {Var}{Greek}{05}
\DeclareFlexSymbol{\Sigma}  {Var}{Greek}{06}
\DeclareFlexSymbol{\Upsilon}{Var}{Greek}{07}
\DeclareFlexSymbol{\Phi}    {Var}{Greek}{08}
\DeclareFlexSymbol{\Psi}    {Var}{Greek}{09}
\DeclareFlexSymbol{\Omega}  {Var}{Greek}{0A}
```

Decimal digits.

```
\DeclareFlexSymbol{0}{Var}{digit}{30}
\DeclareFlexSymbol{1}{Var}{digit}{31}
\DeclareFlexSymbol{2}{Var}{digit}{32}
\DeclareFlexSymbol{3}{Var}{digit}{33}
\DeclareFlexSymbol{4}{Var}{digit}{34}
\DeclareFlexSymbol{5}{Var}{digit}{35}
\DeclareFlexSymbol{6}{Var}{digit}{36}
\DeclareFlexSymbol{7}{Var}{digit}{37}
\DeclareFlexSymbol{8}{Var}{digit}{38}
\DeclareFlexSymbol{9}{Var}{digit}{39}
```

Symbols from the 128-character `cmmi` encoding.

```
\DeclareFlexSymbol{,}{Pun}{OML}{3B}
\DeclareFlexSymbol{.}{Ord}{OML}{3A}
\DeclareFlexSymbol{/}{Ord}{OML}{3D}
\DeclareFlexSymbol{<}{Rel}{OML}{3C}
\DeclareFlexSymbol{>}{Rel}{OML}{3E}
```

To do: make the Var property of lc Greek work properly.

```
\DeclareFlexSymbol{\alpha}     {Var}{greek}{0B}
\DeclareFlexSymbol{\beta}      {Var}{greek}{0C}
\DeclareFlexSymbol{\gamma}     {Var}{greek}{0D}
\DeclareFlexSymbol{\delta}     {Var}{greek}{0E}
\DeclareFlexSymbol{\epsilon}   {Var}{greek}{0F}
\DeclareFlexSymbol{\zeta}      {Var}{greek}{10}
\DeclareFlexSymbol{\eta}       {Var}{greek}{11}
\DeclareFlexSymbol{\theta}     {Var}{greek}{12}
\DeclareFlexSymbol{\iota}      {Var}{greek}{13}
\DeclareFlexSymbol{\kappa}     {Var}{greek}{14}
\DeclareFlexSymbol{\lambda}    {Var}{greek}{15}
\DeclareFlexSymbol{\mu}        {Var}{greek}{16}
\DeclareFlexSymbol{\nu}        {Var}{greek}{17}
\DeclareFlexSymbol{\xi}        {Var}{greek}{18}
\DeclareFlexSymbol{\pi}        {Var}{greek}{19}
\DeclareFlexSymbol{\rho}       {Var}{greek}{1A}
```

```
\DeclareFlexSymbol{\sigma}     {Var}{greek}{1B}
\DeclareFlexSymbol{\tau}       {Var}{greek}{1C}
\DeclareFlexSymbol{\upsilon}   {Var}{greek}{1D}
\DeclareFlexSymbol{\phi}       {Var}{greek}{1E}
\DeclareFlexSymbol{\chi}       {Var}{greek}{1F}
\DeclareFlexSymbol{\psi}       {Var}{greek}{20}
\DeclareFlexSymbol{\omega}     {Var}{greek}{21}
\DeclareFlexSymbol{\varepsilon}{Var}{greek}{22}
\DeclareFlexSymbol{\vartheta}  {Var}{greek}{23}
\DeclareFlexSymbol{\varpi}     {Var}{greek}{24}
\DeclareFlexSymbol{\varrho}    {Var}{greek}{25}
\DeclareFlexSymbol{\varsigma}  {Var}{greek}{26}
\DeclareFlexSymbol{\varphi}    {Var}{greek}{27}
```

Note that in plain TEX `\imath` and `\jmath` are not variable-font. But if a `j` changes font to, let's say, sans serif or calligraphic, a dotless `j` in the same context should change font in the same way.

```
\DeclareFlexSymbol{\imath}           {Var}{OML}{7B}
\DeclareFlexSymbol{\jmath}           {Var}{OML}{7C}
\DeclareFlexSymbol{\ell}             {Ord}{OML}{60}
\DeclareFlexSymbol{\wp}              {Ord}{OML}{7D}
\DeclareFlexSymbol{\partial}         {Ord}{OML}{40}
\DeclareFlexSymbol{\flat}            {Ord}{OML}{5B}
\DeclareFlexSymbol{\natural}         {Ord}{OML}{5C}
\DeclareFlexSymbol{\sharp}           {Ord}{OML}{5D}
\DeclareFlexSymbol{\triangleleft}    {Bin}{OML}{2F}
\DeclareFlexSymbol{\triangleright}   {Bin}{OML}{2E}
\DeclareFlexSymbol{\star}            {Bin}{OML}{3F}
\DeclareFlexSymbol{\smile}           {Rel}{OML}{5E}
\DeclareFlexSymbol{\frown}           {Rel}{OML}{5F}
\DeclareFlexSymbol{\leftharpoonup}   {Rel}{OML}{28}
\DeclareFlexSymbol{\leftharpoondown} {Rel}{OML}{29}
\DeclareFlexSymbol{\rightharpoonup}  {Rel}{OML}{2A}
\DeclareFlexSymbol{\rightharpoondown}{Rel}{OML}{2B}
```

Latin

```
\DeclareFlexSymbol{a}{Var}{latin}{61}
\DeclareFlexSymbol{b}{Var}{latin}{62}
\DeclareFlexSymbol{c}{Var}{latin}{63}
\DeclareFlexSymbol{d}{Var}{latin}{64}
\DeclareFlexSymbol{e}{Var}{latin}{65}
\DeclareFlexSymbol{f}{Var}{latin}{66}
\DeclareFlexSymbol{g}{Var}{latin}{67}
\DeclareFlexSymbol{h}{Var}{latin}{68}
\DeclareFlexSymbol{i}{Var}{latin}{69}
\DeclareFlexSymbol{j}{Var}{latin}{6A}
\DeclareFlexSymbol{k}{Var}{latin}{6B}
\DeclareFlexSymbol{l}{Var}{latin}{6C}
\DeclareFlexSymbol{m}{Var}{latin}{6D}
\DeclareFlexSymbol{n}{Var}{latin}{6E}
```

```
\DeclareFlexSymbol{o}{Var}{latin}{6F}
\DeclareFlexSymbol{p}{Var}{latin}{70}
\DeclareFlexSymbol{q}{Var}{latin}{71}
\DeclareFlexSymbol{r}{Var}{latin}{72}
\DeclareFlexSymbol{s}{Var}{latin}{73}
\DeclareFlexSymbol{t}{Var}{latin}{74}
\DeclareFlexSymbol{u}{Var}{latin}{75}
\DeclareFlexSymbol{v}{Var}{latin}{76}
\DeclareFlexSymbol{w}{Var}{latin}{77}
\DeclareFlexSymbol{x}{Var}{latin}{78}
\DeclareFlexSymbol{y}{Var}{latin}{79}
\DeclareFlexSymbol{z}{Var}{latin}{7A}
\DeclareFlexSymbol{A}{Var}{Latin}{41}
\DeclareFlexSymbol{B}{Var}{Latin}{42}
\DeclareFlexSymbol{C}{Var}{Latin}{43}
\DeclareFlexSymbol{D}{Var}{Latin}{44}
\DeclareFlexSymbol{E}{Var}{Latin}{45}
\DeclareFlexSymbol{F}{Var}{Latin}{46}
\DeclareFlexSymbol{G}{Var}{Latin}{47}
\DeclareFlexSymbol{H}{Var}{Latin}{48}
\DeclareFlexSymbol{I}{Var}{Latin}{49}
\DeclareFlexSymbol{J}{Var}{Latin}{4A}
\DeclareFlexSymbol{K}{Var}{Latin}{4B}
\DeclareFlexSymbol{L}{Var}{Latin}{4C}
\DeclareFlexSymbol{M}{Var}{Latin}{4D}
\DeclareFlexSymbol{N}{Var}{Latin}{4E}
\DeclareFlexSymbol{O}{Var}{Latin}{4F}
\DeclareFlexSymbol{P}{Var}{Latin}{50}
\DeclareFlexSymbol{Q}{Var}{Latin}{51}
\DeclareFlexSymbol{R}{Var}{Latin}{52}
\DeclareFlexSymbol{S}{Var}{Latin}{53}
\DeclareFlexSymbol{T}{Var}{Latin}{54}
\DeclareFlexSymbol{U}{Var}{Latin}{55}
\DeclareFlexSymbol{V}{Var}{Latin}{56}
\DeclareFlexSymbol{W}{Var}{Latin}{57}
\DeclareFlexSymbol{X}{Var}{Latin}{58}
\DeclareFlexSymbol{Y}{Var}{Latin}{59}
\DeclareFlexSymbol{Z}{Var}{Latin}{5A}
```

The `\ldotPun` glyph is used in constructing the `\ldots` symbol. It is just a period with a different math symbol class. `\lhookRel` and `\rhookRel` are used in a similar way for building hooked arrow symbols.

```
\DeclareFlexSymbol{\ldotPun}{Pun}{OML}{3A}
\def\ldotp{\ldotPun}
\DeclareFlexSymbol{\lhookRel}{Rel}{OML}{2C}
\DeclareFlexSymbol{\rhookRel}{Rel}{OML}{2D}
```

Symbols from the 128-character `cmsy` encoding.

```
\DeclareFlexSymbol{*}           {Bin}{bin}{03} % \ast
\AtBeginDocument{\DeclareFlexSymbol{-}          {Bin}{bin}{00}}
```

```
\DeclareFlexSymbol{|}            {Ord}{OMS}{6A}
\DeclareFlexSymbol{\aleph}       {Ord}{ord}{40}
\DeclareFlexSymbol{\Re}          {Ord}{ord}{3C}
\DeclareFlexSymbol{\Im}          {Ord}{ord}{3D}
\DeclareFlexSymbol{\infty}       {Ord}{ord}{31}
\DeclareFlexSymbol{\prime}       {Ord}{ord}{30}
\DeclareFlexSymbol{\emptyset}    {Ord}{ord}{3B}
\DeclareFlexSymbol{\nabla}       {Ord}{ord}{72}
\DeclareFlexSymbol{\top}         {Ord}{ord}{3E}
\DeclareFlexSymbol{\bot}         {Ord}{ord}{3F}
\DeclareFlexSymbol{\triangle}    {Ord}{ord}{34}
\DeclareFlexSymbol{\forall}      {Ord}{ord}{38}
\DeclareFlexSymbol{\exists}      {Ord}{ord}{39}
\DeclareFlexSymbol{\neg}         {Ord}{ord}{3A}
\DeclareFlexSymbol{\clubsuit}    {Ord}{ord}{7C}
\DeclareFlexSymbol{\diamondsuit} {Ord}{ord}{7D}
\DeclareFlexSymbol{\heartsuit}   {Ord}{ord}{7E}
\DeclareFlexSymbol{\spadesuit}   {Ord}{ord}{7F}
\DeclareFlexSymbol{\smallint}    {COs}{OMS}{73}
```

Binary operators.

```
\DeclareFlexSymbol{\bigtriangleup}  {Bin}{bin}{34}
\DeclareFlexSymbol{\bigtriangledown}{Bin}{bin}{35}
\DeclareFlexSymbol{\wedge}          {Bin}{bin}{5E}
\DeclareFlexSymbol{\vee}            {Bin}{bin}{5F}
\DeclareFlexSymbol{\cap}            {Bin}{bin}{5C}
\DeclareFlexSymbol{\cup}            {Bin}{bin}{5B}
\DeclareFlexSymbol{\ddagger}        {Bin}{bin}{7A}
\DeclareFlexSymbol{\dagger}         {Bin}{bin}{79}
\DeclareFlexSymbol{\sqcap}          {Bin}{bin}{75}
\DeclareFlexSymbol{\sqcup}          {Bin}{bin}{74}
\DeclareFlexSymbol{\uplus}          {Bin}{bin}{5D}
\DeclareFlexSymbol{\amalg}          {Bin}{bin}{71}
\DeclareFlexSymbol{\diamond}        {Bin}{bin}{05}
\DeclareFlexSymbol{\bullet}         {Bin}{bin}{0F}
\DeclareFlexSymbol{\wr}             {Bin}{bin}{6F}
\DeclareFlexSymbol{\div}            {Bin}{bin}{04}
\DeclareFlexSymbol{\odot}           {Bin}{bin}{0C}
\DeclareFlexSymbol{\oslash}         {Bin}{bin}{0B}
\DeclareFlexSymbol{\otimes}         {Bin}{bin}{0A}
\DeclareFlexSymbol{\ominus}         {Bin}{bin}{09}
\DeclareFlexSymbol{\oplus}          {Bin}{bin}{08}
\DeclareFlexSymbol{\mp}             {Bin}{bin}{07}
\DeclareFlexSymbol{\pm}             {Bin}{bin}{06}
\DeclareFlexSymbol{\circ}           {Bin}{bin}{0E}
\DeclareFlexSymbol{\bigcirc}        {Bin}{bin}{0D}
\DeclareFlexSymbol{\setminus}       {Bin}{bin}{6E}
\DeclareFlexSymbol{\cdot}           {Bin}{bin}{01}
\DeclareFlexSymbol{\ast}            {Bin}{bin}{03}
\DeclareFlexSymbol{\times}          {Bin}{bin}{02}
```

Relation symbols.

```
\DeclareFlexSymbol{\propto}        {Rel}{rel}{2F}
\DeclareFlexSymbol{\sqsubseteq}    {Rel}{rel}{76}
\DeclareFlexSymbol{\sqsupseteq}    {Rel}{rel}{77}
\DeclareFlexSymbol{\parallel}      {Rel}{rel}{6B}
\DeclareFlexSymbol{\mid}           {Rel}{rel}{6A}
\DeclareFlexSymbol{\dashv}         {Rel}{rel}{61}
\DeclareFlexSymbol{\vdash}         {Rel}{rel}{60}
\DeclareFlexSymbol{\nearrow}       {Rel}{rel}{25}
\DeclareFlexSymbol{\searrow}       {Rel}{rel}{26}
\DeclareFlexSymbol{\nwarrow}       {Rel}{rel}{2D}
\DeclareFlexSymbol{\swarrow}       {Rel}{rel}{2E}
\DeclareFlexSymbol{\Leftrightarrow}{Rel}{rel}{2C}
\DeclareFlexSymbol{\Leftarrow}     {Rel}{rel}{28}
\DeclareFlexSymbol{\Rightarrow}    {Rel}{rel}{29}
\DeclareFlexSymbol{\leq}           {Rel}{rel}{14}
\DeclareFlexSymbol{\geq}           {Rel}{rel}{15}
\DeclareFlexSymbol{\succ}          {Rel}{rel}{1F}
\DeclareFlexSymbol{\prec}          {Rel}{rel}{1E}
\DeclareFlexSymbol{\approx}        {Rel}{rel}{19}
\DeclareFlexSymbol{\succeq}        {Rel}{rel}{17}
\DeclareFlexSymbol{\preceq}        {Rel}{rel}{16}
\DeclareFlexSymbol{\supset}        {Rel}{rel}{1B}
\DeclareFlexSymbol{\subset}        {Rel}{rel}{1A}
\DeclareFlexSymbol{\supseteq}      {Rel}{rel}{13}
\DeclareFlexSymbol{\subseteq}      {Rel}{rel}{12}
\DeclareFlexSymbol{\in}            {Rel}{rel}{32}
\DeclareFlexSymbol{\ni}            {Rel}{rel}{33}
\DeclareFlexSymbol{\gg}            {Rel}{rel}{1D}
\DeclareFlexSymbol{\ll}            {Rel}{rel}{1C}
\DeclareFlexSymbol{\leftrightarrow}{Rel}{rel}{24}
\DeclareFlexSymbol{\leftarrow}     {Rel}{rel}{20}
\DeclareFlexSymbol{\rightarrow}    {Rel}{rel}{21}
\DeclareFlexSymbol{\sim}           {Rel}{rel}{18}
\DeclareFlexSymbol{\simeq}         {Rel}{rel}{27}
\DeclareFlexSymbol{\perp}          {Rel}{rel}{3F}
\DeclareFlexSymbol{\equiv}         {Rel}{rel}{11}
\DeclareFlexSymbol{\asymp}         {Rel}{rel}{10}
```

The `\notRel` glyph is a special zero-width glyph intended only for use in constructing negated symbols. `\mapstoRel` and `\cdotPun` have similar but more restricted applications.

```
\DeclareFlexSymbol{\notRel}   {Rel}{rel}{36}
\DeclareFlexSymbol{\mapstoOrd}{Ord}{OMS}{37}
\DeclareFlexSymbol{\cdotOrd}  {Ord}{OMS}{01}
\cs_set:Npn\cdotp{\mathpunct{\cdotOrd}}
```

Symbols from the 128-character `cmex` encoding. `COs` stands for 'cumulative operator (sum-like)'. `COi` stands for 'cumulative operator (integral-like)'. These typically differ only in the default placement of limits. `cop` stands for 'cumulative

operator math group'.

```
\DeclareFlexSymbol{\coprod}   {COs}{cop}{60}
\DeclareFlexSymbol{\bigvee}   {COs}{cop}{57}
\DeclareFlexSymbol{\bigwedge} {COs}{cop}{56}
\DeclareFlexSymbol{\biguplus} {COs}{cop}{55}
\DeclareFlexSymbol{\bigcap}   {COs}{cop}{54}
\DeclareFlexSymbol{\bigcup}   {COs}{cop}{53}
\DeclareFlexSymbol{\int}      {COi}{cop}{52}
\DeclareFlexSymbol{\prod}     {COs}{cop}{51}
\DeclareFlexSymbol{\sum}      {COs}{cop}{50}
\DeclareFlexSymbol{\bigotimes}{COs}{cop}{4E}
\DeclareFlexSymbol{\bigoplus} {COs}{cop}{4C}
\DeclareFlexSymbol{\bigodot}  {COs}{cop}{4A}
\DeclareFlexSymbol{\oint}     {COi}{cop}{48}
\DeclareFlexSymbol{\bigsqcup} {COs}{cop}{46}
```

Delimiter symbols. `DeL` stands for 'delimiter (left)'. `DeR` stands for 'delimiter (right)'. `DeB` stands for 'delimiter (bidirectional)'. The principal encoding point for an extensible delimiter is the first link in the list of linked sizes as specified in the font metric information. For a math encoding such as OT1/OML/OMS/OMX where not all sizes of a given delimiter reside in a given font, the extra encoding point for the smallest delimiter must be supplied by defining

```
\sd@GXX
```

where G is the mathgroup and XX is the hexadecimal glyph position. `\DeclareFlexDelimiter` does that for us.

```
\DeclareFlexDelimiter{\rangle}{DeR}{del}{0B}{OMS}{69}
\DeclareFlexDelimiter{\langle}{DeL}{del}{0A}{OMS}{68}
\DeclareFlexDelimiter{\rbrace}{DeR}{del}{09}{OMS}{67}
\DeclareFlexDelimiter{\lbrace}{DeL}{del}{08}{OMS}{66}
\DeclareFlexDelimiter{\rceil} {DeR}{del}{07}{OMS}{65}
\DeclareFlexDelimiter{\lceil} {DeL}{del}{06}{OMS}{64}
\DeclareFlexDelimiter{\rfloor}{DeR}{del}{05}{OMS}{63}
\DeclareFlexDelimiter{\lfloor}{DeL}{del}{04}{OMS}{62}
\DeclareFlexDelimiter{(}      {DeL}{del}{00}{OT1}{28}
\DeclareFlexDelimiter{)}      {DeR}{del}{01}{OT1}{29}
\DeclareFlexDelimiter{[}      {DeL}{del}{02}{OT1}{5B}
\DeclareFlexDelimiter{]}      {DeR}{del}{03}{OT1}{5D}
\DeclareFlexDelimiter{\lVert} {DeL}{del}{0D}{OMS}{6B}
\DeclareFlexDelimiter{\rVert} {DeR}{del}{0D}{OMS}{6B}
\DeclareFlexDelimiter{\lvert} {DeL}{del}{0C}{OMS}{6A}
\DeclareFlexDelimiter{\rvert} {DeR}{del}{0C}{OMS}{6A}
\DeclareFlexDelimiter{\Vert}  {DeB}{del}{0D}{OMS}{6B}
\DeclareFlexDelimiter{\vert}  {DeB}{del}{0C}{OMS}{6A}
```

Maybe make the vert bars mathord instead of delimiter, to discourage poor usage.

```
\DeclareFlexDelimiter{|}{DeB}{del}{0C}{OMS}{6A}
\DeclareFlexDelimiter{/}{DeB}{del}{0E}{OML}{3D}
```

These wacky delimiters need to be supported I guess for compabitility reasons. The DeA delimiter type is a special case used only for these arrows.

```
\DeclareFlexDelimiter{\lmoustache} {DeL}{del}{40}{del}{7A}
\DeclareFlexDelimiter{\rmoustache} {DeR}{del}{41}{del}{7B}
\DeclareFlexDelimiter{\lgroup}     {DeL}{del}{3A}{del}{3A}
\DeclareFlexDelimiter{\rgroup}     {DeR}{del}{3B}{del}{3B}
\DeclareFlexDelimiter{\bracevert}  {DeB}{del}{3E}{del}{3E}
\DeclareFlexDelimiter{\arrowvert}  {DeB}{del}{3C}{OMS}{6A}
\DeclareFlexDelimiter{\Arrowvert}  {DeB}{del}{3D}{OMS}{6B}
\DeclareFlexDelimiter{\uparrow}    {DeA}{del}{78}{OMS}{22}
\DeclareFlexDelimiter{\downarrow}  {DeA}{del}{79}{OMS}{23}
\DeclareFlexDelimiter{\updownarrow}{DeA}{del}{3F}{OMS}{6C}
\DeclareFlexDelimiter{\Uparrow}    {DeA}{del}{7E}{OMS}{2A}
\DeclareFlexDelimiter{\Downarrow}  {DeA}{del}{7F}{OMS}{2B}
\DeclareFlexDelimiter{\Updownarrow}{DeA}{del}{77}{OMS}{6D}
\DeclareFlexDelimiter{\backslash}  {DeB}{del}{0F}{OMS}{6E}
```

3 Some compound symbols

The following symbols are not robust in standard LaTeX because they use `#` or `\mathpalette` (which is not robust and contains a `#` in its expansion): `\angle`, `\cong`, `\notin`, `\rightleftharpoons`.

In this definition of `\hbar`, the symbol is cobbled together from a math italic h and the cmr overbar accent glyph.

```
\DeclareFlexSymbol{\hbarOrd}{Ord}{OT1}{16}
\DeclareFlexCompoundSymbol{\hbar}{Ord}{\hbarOrd\mkern-9mu h}
```

For `\surd`, the interior symbol gets math class 1 (cumulative operator) to make the glyph vertically centered on the math axis, but the desired horizontal spacing is the spacing for a mathord. (Couldn't it just be class mathopen, though?)

```
\DeclareFlexSymbol{\surdOrd}{Ord}{OMS}{70}
\DeclareFlexCompoundSymbol{\surd}{Ord}{\mathop{\surdOrd}}
```

As shown in this definition of `\angle`, rule dimens are not allowed to use math-units, unfortunately.

```
\DeclareFlexCompoundSymbol{\angle}{Ord}{%
  \vbox{\ialign{%
      $\m@th\scriptstyle##$\crcr
      \notRel\mathrel{\mkern14mu}\crcr
      \noalign{\nointerlineskip}%
      \mkern2.5mu\leaders\hrule \@height.34pt\hfill\mkern2.5mu\crcr
  }}%
}
```

The `\not` function, which is defined in the flexisym package, requires a suitably defined `\notRel` symbol.

```
\DeclareFlexCompoundSymbol{\neq}{Rel}{\not{=}}
```

.

```
\DeclareFlexCompoundSymbol{\mapsto}{Rel}{\mapstoOrd\rightarrow}
```

The `\@vereq` function ends by centering the whole construction on the math axis, unlike `\buildrel` where the base symbol remains at its normal altitude. Furthermore, `\@vereq` leaves the math style of the top symbol as given instead of downsizing to scriptstyle.

```
\DeclareFlexCompoundSymbol{\cong}{Rel}{\mathpalette\@vereq\sim}
```

The `\m@th` in the `fontmath.ltx` definition of `\notin` is superfluous unless `\c@ncel` doesn't include it (which was perhaps true in an older version of `plain.tex`?).

```
\providecommand*\joinord{}
⟨cmbase | mathptmx⟩ \renewcommand*\joinord{\mkern-3mu }
⟨mathpazo⟩ \renewcommand*\joinord{\mkern-3.45mu }
\DeclareFlexCompoundSymbol{\notin}{Rel}{\mathpalette\c@ncel\in}
\DeclareFlexCompoundSymbol{\rightleftharpoons}{Rel}{\mathpalette\rlh@{}}
\DeclareFlexCompoundSymbol{\doteq}{Rel}{\buildrel\textstyle.\over=}
\DeclareFlexCompoundSymbol{\hookrightarrow}{Rel}{\lhookRel\joinord\rightarrow}
\DeclareFlexCompoundSymbol{\hookleftarrow}{Rel}{\leftarrow\joinord\rhookRel}
\DeclareFlexCompoundSymbol{\bowtie}{Rel}{\triangleright\joinord\triangleleft}
\DeclareFlexCompoundSymbol{\models}{Rel}{\vert\joinord=}
\DeclareFlexCompoundSymbol{\Longrightarrow}{Rel}{\Relbar\joinord\Rightarrow}
\DeclareFlexCompoundSymbol{\longrightarrow}{Rel}{\relbar\joinord\rightarrow}
\DeclareFlexCompoundSymbol{\Longleftarrow}{Rel}{\Leftarrow\joinord\Relbar}
\DeclareFlexCompoundSymbol{\longleftarrow}{Rel}{\leftarrow\joinord\relbar}
\DeclareFlexCompoundSymbol{\longmapsto}{Rel}{\mapstochar\longrightarrow}
\DeclareFlexCompoundSymbol{\longleftrightarrow}{Rel}{\leftarrow\joinord\rightarrow}
\DeclareFlexCompoundSymbol{\Longleftrightarrow}{Rel}{\Leftarrow\joinord\Rightarrow}
```

Here is what you get from the old definition of `\iff`.

```
\glue 2.77771 plus 2.77771
\glue(\thickmuskip) 2.77771 plus 2.77771
\OMS/cmsy/m/n/10 (
\hbox(0.0+0.0)x-1.66663
.\kern -1.66663
\OMS/cmsy/m/n/10 )
\penalty 500
\glue 2.77771 plus 2.77771
\glue(\thickmuskip) 2.77771 plus 2.77771
```

Looks like it could be simplified slightly. But it's not so easy as it looks to do it without screwing up the line breaking possibilities.

```
\renewcommand*\iff{%
  \mskip\thickmuskip\Longleftrightarrow\mskip\thickmuskip
}
```

Some dotly symbols.

```
\DeclareFlexCompoundSymbol{\cdots}{Inn}{\cdotp\cdotp\cdotp}%
\DeclareFlexCompoundSymbol{\vdots}{Ord}{%
```

```
  \vbox{\baselineskip4\p@ \lineskiplimit\z@
    \kern6\p@\hbox{.}\hbox{.}\hbox{.}}}
\DeclareFlexCompoundSymbol{\ddots}{Inn}{%
  \mkern1mu\raise7\p@
  \vbox{\kern7\p@\hbox{.}}\mkern2mu%
  \raise4\p@\hbox{.}\mkern2mu\raise\p@\hbox{.}\mkern1mu%
}
```

.

```
\def\relbar{\begingroup \def\smash@{tb}% in case amsmath is loaded
    \mathpalette\mathsm@sh{\mathchar"200 }\endgroup}
```

For `\Relbar` we take an equal sign of class 0 (Ord) from the operator family. For `cmr` and mathptmx we know this is family 0.

```
⟨cmbase | mathptmx⟩\def\Relbar{\mathchar"3D }
```

For the mathpazo setup we need to use the equal sign from `cmr` and so must insert class 0 and use the symbol from the upright symbols.

```
⟨mathpazo⟩\edef\Relbar{\mathchar\string"\hexnumber@\symupright3D }
```

Done.

```
\ExplSyntaxOff
⟨/cmbase | mathpazo | mathptmx⟩
```

Various synonyms such as `\le` for `\leq` and `\to` for `\rightarrow` are defined in flexisym with `\def` instead of `\let`, for slower execution speed but smaller chance of synchronization problems.

```
⟨*msabm⟩
\ProvidesSymbols{msabm}[2001/09/08 v0.91]
\ExplSyntaxOn

\RequirePackage{amsfonts}\relax

\cs_gset:cpx{mg@MSA}{\hexnumber@\symAMSa}%
\cs_gset:cpx{mg@MSB}{\hexnumber@\symAMSb}%

\DeclareFlexSymbol{\boxdot}             {Bin}{MSA}{00}
\DeclareFlexSymbol{\boxplus}            {Bin}{MSA}{01}
\DeclareFlexSymbol{\boxtimes}           {Bin}{MSA}{02}
\DeclareFlexSymbol{\square}             {Ord}{MSA}{03}
\DeclareFlexSymbol{\blacksquare}        {Ord}{MSA}{04}
\DeclareFlexSymbol{\centerdot}          {Bin}{MSA}{05}
\DeclareFlexSymbol{\lozenge}            {Ord}{MSA}{06}
\DeclareFlexSymbol{\blacklozenge}       {Ord}{MSA}{07}
\DeclareFlexSymbol{\circlearrowright}   {Rel}{MSA}{08}
\DeclareFlexSymbol{\circlearrowleft}    {Rel}{MSA}{09}
```

In amsfonts.sty:

```
%%\DeclareFlexSymbol{\rightleftharpoons}{Rel}{MSA}{0A}
\DeclareFlexSymbol{\leftrightharpoons}  {Rel}{MSA}{0B}
\DeclareFlexSymbol{\boxminus}           {Bin}{MSA}{0C}
\DeclareFlexSymbol{\Vdash}              {Rel}{MSA}{0D}
\DeclareFlexSymbol{\Vvdash}             {Rel}{MSA}{0E}
```

```
\DeclareFlexSymbol{\vDash}              {Rel}{MSA}{0F}
\DeclareFlexSymbol{\twoheadrightarrow}  {Rel}{MSA}{10}
\DeclareFlexSymbol{\twoheadleftarrow}   {Rel}{MSA}{11}
\DeclareFlexSymbol{\leftleftarrows}     {Rel}{MSA}{12}
\DeclareFlexSymbol{\rightrightarrows}   {Rel}{MSA}{13}
\DeclareFlexSymbol{\upuparrows}         {Rel}{MSA}{14}
\DeclareFlexSymbol{\downdownarrows}     {Rel}{MSA}{15}
\DeclareFlexSymbol{\upharpoonright}     {Rel}{MSA}{16}
 \let\restriction\upharpoonright
\DeclareFlexSymbol{\downharpoonright}   {Rel}{MSA}{17}
\DeclareFlexSymbol{\upharpoonleft}      {Rel}{MSA}{18}
\DeclareFlexSymbol{\downharpoonleft}    {Rel}{MSA}{19}
\DeclareFlexSymbol{\rightarrowtail}     {Rel}{MSA}{1A}
\DeclareFlexSymbol{\leftarrowtail}      {Rel}{MSA}{1B}
\DeclareFlexSymbol{\leftrightarrows}    {Rel}{MSA}{1C}
\DeclareFlexSymbol{\rightleftarrows}    {Rel}{MSA}{1D}
\DeclareFlexSymbol{\Lsh}                {Rel}{MSA}{1E}
\DeclareFlexSymbol{\Rsh}                {Rel}{MSA}{1F}
\DeclareFlexSymbol{\rightsquigarrow}    {Rel}{MSA}{20}
\DeclareFlexSymbol{\leftrightsquigarrow}{Rel}{MSA}{21}
\DeclareFlexSymbol{\looparrowleft}      {Rel}{MSA}{22}
\DeclareFlexSymbol{\looparrowright}     {Rel}{MSA}{23}
\DeclareFlexSymbol{\circeq}             {Rel}{MSA}{24}
\DeclareFlexSymbol{\succsim}            {Rel}{MSA}{25}
\DeclareFlexSymbol{\gtrsim}             {Rel}{MSA}{26}
\DeclareFlexSymbol{\gtrapprox}          {Rel}{MSA}{27}
\DeclareFlexSymbol{\multimap}           {Rel}{MSA}{28}
\DeclareFlexSymbol{\therefore}          {Rel}{MSA}{29}
\DeclareFlexSymbol{\because}            {Rel}{MSA}{2A}
\DeclareFlexSymbol{\doteqdot}           {Rel}{MSA}{2B}
 \let\Doteq\doteqdot
\DeclareFlexSymbol{\triangleq}          {Rel}{MSA}{2C}
\DeclareFlexSymbol{\precsim}            {Rel}{MSA}{2D}
\DeclareFlexSymbol{\lesssim}            {Rel}{MSA}{2E}
\DeclareFlexSymbol{\lessapprox}         {Rel}{MSA}{2F}
\DeclareFlexSymbol{\eqslantless}        {Rel}{MSA}{30}
\DeclareFlexSymbol{\eqslantgtr}         {Rel}{MSA}{31}
\DeclareFlexSymbol{\curlyeqprec}        {Rel}{MSA}{32}
\DeclareFlexSymbol{\curlyeqsucc}        {Rel}{MSA}{33}
\DeclareFlexSymbol{\preccurlyeq}        {Rel}{MSA}{34}
\DeclareFlexSymbol{\leqq}               {Rel}{MSA}{35}
\DeclareFlexSymbol{\leqslant}           {Rel}{MSA}{36}
\DeclareFlexSymbol{\lessgtr}            {Rel}{MSA}{37}
\DeclareFlexSymbol{\backprime}          {Ord}{MSA}{38}
\DeclareFlexSymbol{\risingdotseq}       {Rel}{MSA}{3A}
\DeclareFlexSymbol{\fallingdotseq}      {Rel}{MSA}{3B}
\DeclareFlexSymbol{\succcurlyeq}        {Rel}{MSA}{3C}
\DeclareFlexSymbol{\geqq}               {Rel}{MSA}{3D}
\DeclareFlexSymbol{\geqslant}           {Rel}{MSA}{3E}
\DeclareFlexSymbol{\gtrless}            {Rel}{MSA}{3F}
```

in amsfonts.sty

```
%% \DeclareFlexSymbol{\sqsubset}     {Rel}{MSA}{40}
%% \DeclareFlexSymbol{\sqsupset}     {Rel}{MSA}{41}
\DeclareFlexSymbol{\vartriangleright}   {Rel}{MSA}{42}
\DeclareFlexSymbol{\vartriangleleft}    {Rel}{MSA}{43}
\DeclareFlexSymbol{\trianglerighteq}    {Rel}{MSA}{44}
\DeclareFlexSymbol{\trianglelefteq}     {Rel}{MSA}{45}
\DeclareFlexSymbol{\bigstar}            {Ord}{MSA}{46}
\DeclareFlexSymbol{\between}            {Rel}{MSA}{47}
\DeclareFlexSymbol{\blacktriangledown}  {Ord}{MSA}{48}
\DeclareFlexSymbol{\blacktriangleright} {Rel}{MSA}{49}
\DeclareFlexSymbol{\blacktriangleleft}  {Rel}{MSA}{4A}
\DeclareFlexSymbol{\vartriangle}        {Rel}{MSA}{4D}
\DeclareFlexSymbol{\blacktriangle}      {Ord}{MSA}{4E}
\DeclareFlexSymbol{\triangledown}       {Ord}{MSA}{4F}
\DeclareFlexSymbol{\eqcirc}             {Rel}{MSA}{50}
\DeclareFlexSymbol{\lesseqgtr}          {Rel}{MSA}{51}
\DeclareFlexSymbol{\gtreqless}          {Rel}{MSA}{52}
\DeclareFlexSymbol{\lesseqqgtr}         {Rel}{MSA}{53}
\DeclareFlexSymbol{\gtreqqless}         {Rel}{MSA}{54}
\DeclareFlexSymbol{\Rrightarrow}        {Rel}{MSA}{56}
\DeclareFlexSymbol{\Lleftarrow}         {Rel}{MSA}{57}
\DeclareFlexSymbol{\veebar}             {Bin}{MSA}{59}
\DeclareFlexSymbol{\barwedge}           {Bin}{MSA}{5A}
\DeclareFlexSymbol{\doublebarwedge}     {Bin}{MSA}{5B}
```

In amsfonts.sty

```
%%\DeclareFlexSymbol{\angle}         {Ord}{MSA}{5C}
\DeclareFlexSymbol{\measuredangle}   {Ord}{MSA}{5D}
\DeclareFlexSymbol{\sphericalangle}  {Ord}{MSA}{5E}
\DeclareFlexSymbol{\varpropto}       {Rel}{MSA}{5F}
\DeclareFlexSymbol{\smallsmile}      {Rel}{MSA}{60}
\DeclareFlexSymbol{\smallfrown}      {Rel}{MSA}{61}
\DeclareFlexSymbol{\Subset}          {Rel}{MSA}{62}
\DeclareFlexSymbol{\Supset}          {Rel}{MSA}{63}
\DeclareFlexSymbol{\Cup}             {Bin}{MSA}{64}
 \let\doublecup\Cup
\DeclareFlexSymbol{\Cap}             {Bin}{MSA}{65}
 \let\doublecap\Cap
\DeclareFlexSymbol{\curlywedge}      {Bin}{MSA}{66}
\DeclareFlexSymbol{\curlyvee}        {Bin}{MSA}{67}
\DeclareFlexSymbol{\leftthreetimes}  {Bin}{MSA}{68}
\DeclareFlexSymbol{\rightthreetimes} {Bin}{MSA}{69}
\DeclareFlexSymbol{\subseteqq}       {Rel}{MSA}{6A}
\DeclareFlexSymbol{\supseteqq}       {Rel}{MSA}{6B}
\DeclareFlexSymbol{\bumpeq}          {Rel}{MSA}{6C}
\DeclareFlexSymbol{\Bumpeq}          {Rel}{MSA}{6D}
\DeclareFlexSymbol{\lll}             {Rel}{MSA}{6E}
 \let\llless\lll
\DeclareFlexSymbol{\ggg}             {Rel}{MSA}{6F}
```

```
 \let\gggtr\ggg
\DeclareFlexSymbol{\circledS}        {Ord}{MSA}{73}
\DeclareFlexSymbol{\pitchfork}       {Rel}{MSA}{74}
\DeclareFlexSymbol{\dotplus}         {Bin}{MSA}{75}
\DeclareFlexSymbol{\backsim}         {Rel}{MSA}{76}
\DeclareFlexSymbol{\backsimeq}       {Rel}{MSA}{77}
\DeclareFlexSymbol{\complement}      {Ord}{MSA}{7B}
\DeclareFlexSymbol{\intercal}        {Bin}{MSA}{7C}
\DeclareFlexSymbol{\circledcirc}     {Bin}{MSA}{7D}
\DeclareFlexSymbol{\circledast}      {Bin}{MSA}{7E}
\DeclareFlexSymbol{\circleddash}     {Bin}{MSA}{7F}
```

Begin AMSb declarations

```
\DeclareFlexSymbol{\lvertneqq}       {Rel}{MSB}{00}
\DeclareFlexSymbol{\gvertneqq}       {Rel}{MSB}{01}
\DeclareFlexSymbol{\nleq}            {Rel}{MSB}{02}
\DeclareFlexSymbol{\ngeq}            {Rel}{MSB}{03}
\DeclareFlexSymbol{\nless}           {Rel}{MSB}{04}
\DeclareFlexSymbol{\ngtr}            {Rel}{MSB}{05}
\DeclareFlexSymbol{\nprec}           {Rel}{MSB}{06}
\DeclareFlexSymbol{\nsucc}           {Rel}{MSB}{07}
\DeclareFlexSymbol{\lneqq}           {Rel}{MSB}{08}
\DeclareFlexSymbol{\gneqq}           {Rel}{MSB}{09}
\DeclareFlexSymbol{\nleqslant}       {Rel}{MSB}{0A}
\DeclareFlexSymbol{\ngeqslant}       {Rel}{MSB}{0B}
\DeclareFlexSymbol{\lneq}            {Rel}{MSB}{0C}
\DeclareFlexSymbol{\gneq}            {Rel}{MSB}{0D}
\DeclareFlexSymbol{\npreceq}         {Rel}{MSB}{0E}
\DeclareFlexSymbol{\nsucceq}         {Rel}{MSB}{0F}
\DeclareFlexSymbol{\precnsim}        {Rel}{MSB}{10}
\DeclareFlexSymbol{\succnsim}        {Rel}{MSB}{11}
\DeclareFlexSymbol{\lnsim}           {Rel}{MSB}{12}
\DeclareFlexSymbol{\gnsim}           {Rel}{MSB}{13}
\DeclareFlexSymbol{\nleqq}           {Rel}{MSB}{14}
\DeclareFlexSymbol{\ngeqq}           {Rel}{MSB}{15}
\DeclareFlexSymbol{\precneqq}        {Rel}{MSB}{16}
\DeclareFlexSymbol{\succneqq}        {Rel}{MSB}{17}
\DeclareFlexSymbol{\precnapprox}     {Rel}{MSB}{18}
\DeclareFlexSymbol{\succnapprox}     {Rel}{MSB}{19}
\DeclareFlexSymbol{\lnapprox}        {Rel}{MSB}{1A}
\DeclareFlexSymbol{\gnapprox}        {Rel}{MSB}{1B}
\DeclareFlexSymbol{\nsim}            {Rel}{MSB}{1C}
\DeclareFlexSymbol{\ncong}           {Rel}{MSB}{1D}
\DeclareFlexSymbol{\diagup}          {Ord}{MSB}{1E}
\DeclareFlexSymbol{\diagdown}        {Ord}{MSB}{1F}
\DeclareFlexSymbol{\varsubsetneq}    {Rel}{MSB}{20}
\DeclareFlexSymbol{\varsupsetneq}    {Rel}{MSB}{21}
\DeclareFlexSymbol{\nsubseteqq}      {Rel}{MSB}{22}
\DeclareFlexSymbol{\nsupseteqq}      {Rel}{MSB}{23}
\DeclareFlexSymbol{\subsetneqq}      {Rel}{MSB}{24}
```

```
\DeclareFlexSymbol{\supsetneqq}      {Rel}{MSB}{25}
\DeclareFlexSymbol{\varsubsetneqq}   {Rel}{MSB}{26}
\DeclareFlexSymbol{\varsupsetneqq}   {Rel}{MSB}{27}
\DeclareFlexSymbol{\subsetneq}       {Rel}{MSB}{28}
\DeclareFlexSymbol{\supsetneq}       {Rel}{MSB}{29}
\DeclareFlexSymbol{\nsubseteq}       {Rel}{MSB}{2A}
\DeclareFlexSymbol{\nsupseteq}       {Rel}{MSB}{2B}
\DeclareFlexSymbol{\nparallel}       {Rel}{MSB}{2C}
\DeclareFlexSymbol{\nmid}            {Rel}{MSB}{2D}
\DeclareFlexSymbol{\nshortmid}       {Rel}{MSB}{2E}
\DeclareFlexSymbol{\nshortparallel}  {Rel}{MSB}{2F}
\DeclareFlexSymbol{\nvdash}          {Rel}{MSB}{30}
\DeclareFlexSymbol{\nVdash}          {Rel}{MSB}{31}
\DeclareFlexSymbol{\nvDash}          {Rel}{MSB}{32}
\DeclareFlexSymbol{\nVDash}          {Rel}{MSB}{33}
\DeclareFlexSymbol{\ntrianglerighteq}{Rel}{MSB}{34}
\DeclareFlexSymbol{\ntrianglelefteq} {Rel}{MSB}{35}
\DeclareFlexSymbol{\ntriangleleft}   {Rel}{MSB}{36}
\DeclareFlexSymbol{\ntriangleright}  {Rel}{MSB}{37}
\DeclareFlexSymbol{\nleftarrow}      {Rel}{MSB}{38}
\DeclareFlexSymbol{\nrightarrow}     {Rel}{MSB}{39}
\DeclareFlexSymbol{\nLeftarrow}      {Rel}{MSB}{3A}
\DeclareFlexSymbol{\nRightarrow}     {Rel}{MSB}{3B}
\DeclareFlexSymbol{\nLeftrightarrow} {Rel}{MSB}{3C}
\DeclareFlexSymbol{\nleftrightarrow} {Rel}{MSB}{3D}
\DeclareFlexSymbol{\divideontimes}   {Bin}{MSB}{3E}
\DeclareFlexSymbol{\varnothing}      {Ord}{MSB}{3F}
\DeclareFlexSymbol{\nexists}         {Ord}{MSB}{40}
\DeclareFlexSymbol{\Finv}            {Ord}{MSB}{60}
\DeclareFlexSymbol{\Game}            {Ord}{MSB}{61}
```

In amsfonts.sty:

```
%%\DeclareFlexSymbol{\mho}           {Ord}{MSB}{66}
\DeclareFlexSymbol{\eth}             {Ord}{MSB}{67}
\DeclareFlexSymbol{\eqsim}           {Rel}{MSB}{68}
\DeclareFlexSymbol{\beth}            {Ord}{MSB}{69}
\DeclareFlexSymbol{\gimel}           {Ord}{MSB}{6A}
\DeclareFlexSymbol{\daleth}          {Ord}{MSB}{6B}
\DeclareFlexSymbol{\lessdot}         {Bin}{MSB}{6C}
\DeclareFlexSymbol{\gtrdot}          {Bin}{MSB}{6D}
\DeclareFlexSymbol{\ltimes}          {Bin}{MSB}{6E}
\DeclareFlexSymbol{\rtimes}          {Bin}{MSB}{6F}
\DeclareFlexSymbol{\shortmid}        {Rel}{MSB}{70}
\DeclareFlexSymbol{\shortparallel}   {Rel}{MSB}{71}
\DeclareFlexSymbol{\smallsetminus}   {Bin}{MSB}{72}
\DeclareFlexSymbol{\thicksim}        {Rel}{MSB}{73}
\DeclareFlexSymbol{\thickapprox}     {Rel}{MSB}{74}
\DeclareFlexSymbol{\approxeq}        {Rel}{MSB}{75}
\DeclareFlexSymbol{\succapprox}      {Rel}{MSB}{76}
\DeclareFlexSymbol{\precapprox}      {Rel}{MSB}{77}
```

```
\DeclareFlexSymbol{\curvearrowleft}  {Rel}{MSB}{78}
\DeclareFlexSymbol{\curvearrowright} {Rel}{MSB}{79}
\DeclareFlexSymbol{\digamma}         {Ord}{MSB}{7A}
\DeclareFlexSymbol{\varkappa}        {Ord}{MSB}{7B}
\DeclareFlexSymbol{\Bbbk}            {Ord}{MSB}{7C}
\DeclareFlexSymbol{\hslash}          {Ord}{MSB}{7D}
```

In amsfonts.sty:

```
%%\DeclareFlexSymbol{\hbar}          {Ord}{MSB}{7E}
\DeclareFlexSymbol{\backepsilon}     {Rel}{MSB}{7F}
\ExplSyntaxOff
⟨/msabm⟩
```

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

\" 7, 12
\’ 365, 374
* 365, 374
\- 366, 367, 369, 372, 375, 376, 378, 381
\. 349, 365, 374
\/ 372, 381
\: 372, 381
\@firstofone 100
\@gobble 13, 346
\@gobblethree 328
\@height 713
\@ifpackageloaded 361
\@ifpackagewith 419, 424
\@ifundefined 284, 290
\@nil 347, 350
\@onefilewithoptions 335
\@sym 13, 15, 94, 229, 255, 264, 329, 344
\@sym@ord 344, 347
\@sym@ord@a 347, 350
\@symAcc 94
\@symBin 85
\@symCOi 88
\@symCOs 89
\@symDeA 93
\@symDeB 92
\@symDeL 90
\@symDeR 91
\@symErr 313
\@symOrd 83
\@symPun 87
\@symRel 86
\@symVar 84, 96–98
\@symextension 332, 335
\@symtype . 83–93, 96–98, 100, 265, 344
\@symtype@ord 344, 346
\@tempb 228, 233, 237, 334, 335
\@typeset@protect 16
\@undefined 364
\@vereq 718
_ 35, 38, 41, 44, 349
\| 400

A

\active 225, 260
\aleph 568
\alpha 463
\amalg 597
\angle 708, 849
\approx 633
\approxeq 965
\Arrowvert 696
\arrowvert 695
\ast 565, 613
\asymp 651
\AtBeginDocument 363, 435, 566
\AtEndOfPackage 6

B

\backepsilon 975

\backprime 818
\backsim 876
\backsimeq 877
\backslash 703
\barwedge 847
\baselineskip 741
\Bbbk 972
\because 803
\begingroup 227, 254, 261, 342, 348, 748
\beta 464
\beth 953
\between 832
\bigcap 660
\bigcirc 610
\bigcup 661
\bigodot 667
\bigoplus 666
\bigotimes 665
\bigsqcup 669
\bigstar 831
\bigtriangledown 587
\bigtriangleup 586
\biguplus 659
\bigvee 657
\bigwedge 658
\binrel@ 326
\binrel@@ 305, 318–324
\binrel@a 317, 329
\binrel@sym 304, 329
\blacklozenge 767
\blacksquare 764
\blacktriangle 837
\blacktriangledown 833
\blacktriangleleft 835
\blacktriangleright 834
\bot 576
\bowtie 727
\box 293, 298, 301
\boxdot 760
\boxminus 772
\boxplus 761
\boxtimes 762
\bracevert 694
\buildrel 724
\bullet 599
\Bumpeq 868
\bumpeq 867

C

\c@ncel 722
\Cap 859, 860
\cap 590
\catcode 7, 12, 224, 225, 259, 260
\cdot 612
\cdotOrd 654, 655
\cdotp 655, 739
\cdots 739
\centerdot 765
\char 257
\check@mathfonts 262
\chi 483
\circ 609
\circeq 797
\circlearrowleft 769
\circlearrowright 768
\circledast 881
\circledcirc 880
\circleddash 882
\circledS 873
\cleaders 292, 296, 300
\clist 334
\clubsuit 581
\colon 433
\complement 878
\cong 718
\coprod 656
\copy 292, 297
\crcr 710, 711, 713
\cs 34, 37, 40, 43, 50–62, 64, 67, 68, 71, 74, 78, 79, 83–94, 96–98, 102–114, 116–143, 145–160, 162–189, 192, 194–221, 228, 242, 244, 255, 263–267, 270, 358, 408–418, 420, 422, 425, 427–429, 655, 758, 759
\csname 248
\Cup 857, 858
\cup 591
\curlyeqprec 812
\curlyeqsucc 813
\curlyvee 862
\curlywedge 861
\curvearrowleft 968
\curvearrowright 969

D

\dagger 593
\daleth 955
\dashv 620
\ddagger 592

\ddots 743
\DeclareFlexCompoundSymbol 247, 705, 707, 708, 716–718, 722–735, 739, 740, 743
\DeclareFlexDelimiter . . 242, 670–703
\DeclareFlexSymbol . 226, 243, 430–561, 563–654, 656–669, 704, 706, 760–782, 784–804, 806–857, 859, 861–869, 871, 873–975
\DeclareOption 14, 384, 386–388
\DeclareRobustCommand 248, 251, 252, 281, 340, 341
\def 15, 19–33, 226, 247, 253, 261, 269, 285, 291, 295, 299, 304, 317–324, 326, 332, 344, 346, 347, 350, 562, 748, 750
\Delta 438
\delta 466
\diagdown 914
\diagup 913
\diamond 598
\diamondsuit 582
\digamma 970
\displaylimits 47, 49
\div 601
\divideontimes 945
\do 5, 11
\Doteq 805
\doteq 724
\doteqdot 804, 805
\dotplus 875
\doublebarwedge 848
\doublecap 860
\doublecup 858
\Downarrow 701
\downarrow 698
\downdownarrows 781
\downharpoonleft 786
\downharpoonright 784
\dp 287

E

\edef 5, 224, 259, 751
\ell 494
\else 16, 234, 273, 307–313, 354, 366, 368, 373, 375, 377
\emptyset 573
\endcsname 248
\endgroup . 239, 257, 277, 342, 349, 749
\endinput 402
\epsilon 467
\eqcirc 839
\eqsim 952
\eqslantgtr 811
\eqslantless 810
\equiv 650
\eta 469
\eth 951
\everymath 268
\ExecuteOptions 389
\exists 579
\exp 16, 65, 69, 75, 76, 229, 230, 232, 244, 248, 313, 314, 335, 347, 414
\expandafter 274
\ExplSyntaxOff 752, 976
\ExplSyntaxOn 407, 756

F

\fallingdotseq 820
\fi 16, 238, 276, 313, 356, 370, 371, 379, 380, 382
\Finv 948
\flat 497
\forall 578
\frown 504

G

\Game 949
\Gamma 437
\gamma 465
\gdef 318–324, 365, 374
\ge 395
\geq 395, 630
\geqq 822
\geqslant 823
\gets 398
\gg 642
\ggg 871, 872
\gggtr 872
\gimel 954
\global 101
\gnapprox 910
\gneq 896
\gneqq 892
\gnsim 902
\gtrapprox 800
\gtrdot 957
\gtreqless 841
\gtreqqless 843
\gtrless 824

\gtrsim 799
\gvertneqq 884

H

\hbar 705, 974
\hbarOrd 704, 705
\hbox 282, 286, 293, 297, 301, 327, 742, 745, 746
\heartsuit 583
\hexnumber@ 408–411, 422, 751, 758, 759
\hfill 293, 297, 301, 713
\hookleftarrow 726
\hookrightarrow 725
\hrule 713
\hslash 973
\ht 287, 291, 295, 299

I

\ialign 709
\if 351
\ifcat 232, 271
\iff 736
\ifmmode 267
\ifnum 366, 367, 375, 376
\iftrue 267
\ifx 16, 306–312, 364
\Im 570
\imath 492
\in 640, 722
\infty 571
\int 662
\intercal 879
\intlimits 48, 56, 88
\iota 471

J

\jmath 493
\joinord . . 719–721, 725–732, 734, 735

K

\kappa 472
\kern 742, 745

L

\Lambda 440
\lambda 473
\land 392
\langle 671
\lbrace 673
\lccode 236, 274, 349
\lceil 675
\ldotp 562
\ldotPun 561, 562
\le 394
\leaders 713
\leavevmode 281
\Leftarrow 627, 731, 735
\leftarrow 296, 300, 398, 645, 726, 732, 734
\leftarrowfill@ 295
\leftarrowtail 788
\leftharpoondown 506
\leftharpoonup 505
\leftleftarrows 778
\Leftrightarrow 626
\leftrightarrow 644
\leftrightarrowfill@ 299
\leftrightarrows 789
\leftrightharpoons 771
\leftrightsquigarrow 794
\leftthreetimes 863
\leq 394, 629
\leqq 815
\leqslant 816
\lessapprox 809
\lessdot 956
\lesseqgtr 840
\lesseqqgtr 842
\lessgtr 817
\lesssim 808
\let 11, 13, 47–49, 100, 101, 233, 237, 249, 328, 329, 344, 345, 783, 805, 858, 860, 870, 872
\lfloor 677
\lgroup 692
\lhookRel 563, 725
\lineskiplimit 741
\ll 643
\Lleftarrow 845
\lll 869, 870
\llless 870
\lmoustache 690
\lnapprox 909
\lneq 895
\lneqq 891
\lnot 391
\lnsim 901
\Longleftarrow 731
\longleftarrow 732
\Longleftrightarrow 735, 737

\longleftrightarrow 734
\longmapsto 733
\Longrightarrow 729
\longrightarrow 730, 733
\looparrowleft 795
\looparrowright 796
\lor 393
\lowercase 237, 275, 349
\lozenge 766
\Lsh 791
\ltimes 958
\lVert 682
\lvert 684
\lvertneqq 883

M

\m@Bin 52, 53, 104, 105
\m@COi 56, 108
\m@COs 55, 107
\m@DeA 60, 112
\m@DeB 59, 111
\m@DeL 57, 109
\m@DeR 58, 110
\m@Ord 50, 102
\m@Pun 54, 106
\m@Rad 61, 113
\m@th 291, 295, 299, 710
\m@Var 51, 103
\mapsto 717
\mapstochar 733
\mapstoOrd 653, 717
\math 34, 37, 40, 43, 50–62, 64, 65, 67–69, 71, 74, 75, 78, 83–94, 96–98, 102–114, 116–143, 145–160, 162–189, 192, 194–221, 223, 306–312, 318–324, 340, 352, 353, 355, 358, 359
\mathbin 85
\mathchar 749–751
\mathchardef 369
\mathchars@reset 342, 344
\mathchoice 328
\mathclose 91, 97
\mathcode 235, 365–367, 369, 372, 374, 381
\mathinner 98
\mathop 88, 89, 707
\mathopen 90, 96
\mathord 83, 84, 92
\mathpalette 718, 722, 723, 749
\mathpunct 87, 655
\mathrel 86, 93, 711
\mathsm@sh 749
\Mathstrutbox@ 287
\measuredangle 850
\mg@acc 24
\mg@arr 23
\mg@Bbb 30
\mg@bflatin 29
\mg@bin 19, 412, 427, 428
\mg@cal 31
\mg@cop 25, 429
\mg@del 22, 413, 429
\mg@digit 33, 414, 418
\mg@frak 32
\mg@Greek 28, 418, 420, 425
\mg@greek 27, 417
\mg@Latin 416
\mg@latin 26, 415–417, 420, 425
\mg@nre 21
\mg@OML 415
\mg@OMS 412
\mg@OMX 413
\mg@ord 428
\mg@rel 20, 427
\mho 950
\mid 619
\mkern 269, 270, 292–294, 296–298, 300–302, 705, 711, 713, 720, 721, 744–746
\models 728
\mp 607
\mskip 270, 737
\mu 474
\multimap 801
\muskip 269

N

\nabla 574
\namelimits 49
\natural 498
\ncong 912
\ne 396
\nearrow 622
\neg 391, 580
\neq 396, 716
\newcommand 333, 339
\newmcodes@ 365, 374
\nexists 947
\ngeq 886

\ngeqq 904
\ngeqslant 894
\ngtr 888
\ni 397, 641
\nLeftarrow 941
\nleftarrow 939
\nLeftrightarrow 943
\nleftrightarrow 944
\nleq 885
\nleqq 903
\nleqslant 893
\nless 887
\nmid 928
\noalign 712
\noexpand 6, 271
\nointerlineskip 712
\nolimits 48
\not 340, 716
\notin 722
\notRel 340, 652, 711
\nparallel 927
\nprec 889
\npreceq 897
\nRightarrow 942
\nrightarrow 940
\nshortmid 929
\nshortparallel 930
\nsim 911
\nsubseteq 925
\nsubseteqq 917
\nsucc 890
\nsucceq 898
\nsupseteq 926
\nsupseteqq 918
\ntriangleleft 937
\ntrianglelefteq 936
\ntriangleright 938
\ntrianglerighteq 935
\nu 475
\number 7
\nVDash 934
\nVdash 932
\nvDash 933
\nvdash 931
\nwarrow 624

O

\odot 602
\oint 668
\Omega 447
\omega 485
\ominus 605
\oplus 606
\OrdSymbol 83–93, 96–98, 266, 294, 296, 300, 302, 314, 318–324, 340, 341, 345
\oslash 603
\otimes 604
\over 724
\owns 397

P

\p@ 741, 742, 744–746
\parallel 618
\partial 496
\PassOptionsToPackage 385
\perp 649
\Phi 445
\phi 482
\Pi 442
\pi 477
\pitchfork 874
\pm 608
\prec 632
\precapprox 967
\preccurlyeq 814
\preceq 635
\precnapprox 907
\precneqq 905
\precnsim 899
\precsim 807
\prg 255
\prime 282, 572
\ProcessOptions 390
\prod 663
\propto 615
\protect 16
\providecommand 280, 719
\ProvidesExplFile 338
\ProvidesExplPackage 3
\ProvidesFile 339
\ProvidesSymbols . . 339, 404–406, 755
\Psi 446
\psi 484

R

\raise 282, 744, 746
\rangle 670
\rbrace 672
\rceil 674

\Re 569
\relax 8, 11, 232, 235, 236, 249, 271, 274, 345, 369, 372, 378, 381, 390, 757
\Relbar 729, 731, 750, 751
\relbar . . . 291, 295, 299, 730, 732, 748
\renewcommand . . 391–400, 720, 721, 736
\RequirePackage 2, 401, 757
\resetMathstrut@ 285
\restriction 783
\rfloor 676
\rgroup 693
\rho 478
\rhookRel 564, 726
\Rightarrow 628, 729, 735
\rightarrow 294, 302, 399, 646, 717, 725, 730, 734
\rightarrowfill@ 291
\rightarrowtail 787
\rightharpoondown 508
\rightharpoonup 507
\rightleftarrows 790
\rightleftharpoons 723, 770
\rightrightarrows 779
\rightsquigarrow 793
\rightthreetimes 864
\risingdotseq 819
\rlh@ 723
\rmoustache 691
\Rrightarrow 844
\Rsh 792
\rtimes 959
\rVert 683
\rvert 685

S

\scriptchar 252
\scriptfont 252, 282
\scriptstyle 710
\searrow 623
\setbox 286, 327
\setboxz@h 291, 295, 299
\setminus 611
\sharp 499
\shortmid 960
\shortparallel 961
\Sigma 443
\sigma 479
\sim 647, 718
\simeq 648
\smallfrown 854
\smallint 585
\smallsetminus 962
\smallsmile 853
\smash@ 748
\smile 503
\space 367, 376
\spadesuit 584
\sphericalangle 851
\sqcap 594
\sqcup 595
\sqsubset 825
\sqsubseteq 616
\sqsupset 826
\sqsupseteq 617
\square 763
\star 502
\std@minus 369, 378
\string 274, 347, 751
\Subset 855
\subset 637
\subseteq 639
\subseteqq 865
\subsetneq 923
\subsetneqq 919
\succ 631
\succapprox 966
\succcurlyeq 821
\succeq 634
\succnapprox 908
\succneqq 906
\succnsim 900
\succsim 798
\sum 664
\sumlimits 47, 55, 89
\Supset 856
\supset 636
\supseteq 638
\supseteqq 866
\supsetneq 924
\supsetneqq 920
\surd 707
\surdOrd 706, 707
\swarrow 625
\sym@global . . . 101, 233, 235, 237, 249
\symAMSa 758
\symAMSb 759
\symlargesymbols 411
\symletters 409
\symoperators 408

\symsymbols 410
\symupright 422, 751

T

\tau 480
\tex 35, 38, 41, 44
\text@char 251, 252, 261, 282
\text@char@sym 253, 264
\text@script@char 256, 263
\textchar 251, 286
\textfont 251
\textprime 280, 281
\textstyle 724
\the 224, 256, 259
\therefore 802
\Theta 439
\theta 470
\thickapprox 964
\thickmuskip 737
\thicksim 963
\times 614
\tmp 224, 241, 259, 279
\to 399
\top 575
\triangle 577
\triangledown 838
\triangleleft 500, 727
\trianglelefteq 830
\triangleq 806
\triangleright 501, 727
\trianglerighteq 829
\twoheadleftarrow 777
\twoheadrightarrow 776

U

\Umathcharnumdef 378
\Umathcode 364
\Umathcodenum 375, 376, 378
\Uparrow 700
\uparrow 697
\Updownarrow 702
\updownarrow 699
\upharpoonleft 785
\upharpoonright 782, 783
\uplus 596
\Upsilon 444
\upsilon 481
\upuparrows 780
\use 16, 65, 69, 75, 76, 81, 244, 265, 266, 268
\usesymbols 333, 386–388

V

\varepsilon 486
\varkappa 971
\varnothing 946
\varphi 491
\varpi 488
\varpropto 852
\varrho 489
\varsigma 490
\varsubsetneq 915
\varsubsetneqq 921
\varsupsetneq 916
\varsupsetneqq 922
\vartheta 487
\vartriangle 836
\vartriangleleft 828
\vartriangleright 827
\vbox 709, 741, 745
\Vdash 773
\vDash 775
\vdash 621
\vdots 740
\vee 393, 589
\veebar 846
\Vert 400, 686
\vert 286, 687, 728
\Vvdash 774

W

\wedge 392, 588
\wp 495
\wr 600

X

\xdef 305
\Xi 441
\xi 476

Z

\z@ 224, 225, 236, 259, 260, 269, 274, 286, 287, 291–293, 295, 297–299, 301, 327, 741
\zeta 468