The `amscd` package

Frank Mittelbach Rainer Schöpf Michael Downes

Version v2.1, 2017/04/14

This file is maintained by the LaTeX Project team.
Bug reports can be opened (category `amslatex`) at
https://latex-project.org/bugs/.

1 Introduction

The `amscd` package provides a `CD` environment that emulates the commutative diagram capabilities of $\mathcal{AMS}$-TEX version 2.x. This means that only simple rectangular diagrams are supported, with no diagonal arrows or more exotic features. Many users will be better served by one of the more powerful diagram packages such as `diagram`, `xypic`, or `kuvio`.

Example:

$$\begin{array}{ccc} S^{\mathcal{W}_\Lambda}\otimes T & \xrightarrow{j} & T \\ \big\downarrow & & \big\downarrow{\scriptstyle \operatorname{End} P} \\ (S\otimes T)/I & = & (Z\otimes T)/J \end{array} \tag{1}$$

```
\begin{equation}\begin{CD}
S^{{\mathcal{W}}_\Lambda}\otimes T   @>j>>   T\\
@VVV                                    @VV{\End P}V\\
(S\otimes T)/I                  @=      (Z\otimes T)/J
\end{CD}\end{equation}
```

(assuming `\End` is defined as an 'operator name'.

Another example:

We will make liberal use of Cichon's Diagram [C]:

$$\begin{array}{ccccccc} \operatorname{cov}(\mathcal{L}) & \longrightarrow & \operatorname{non}(\mathcal{K}) & \longrightarrow & \operatorname{cf}(\mathcal{K}) & \longrightarrow & \operatorname{cf}(\mathcal{L}) \\ \big\downarrow & & \big\uparrow & & \big\uparrow & & \big\downarrow \\ \operatorname{add}(\mathcal{L}) & \longrightarrow & \operatorname{add}(\mathcal{K}) & \longrightarrow & \operatorname{cov}(\mathcal{K}) & \longrightarrow & \operatorname{non}(\mathcal{L}) \end{array} \tag{2}$$

```
\begin{equation}\begin{CD}
\cov(\mathcal{L}) @>>> \non(\mathcal{K}) @>>> \cf(\mathcal{K}) @>>>
\cf(\mathcal{L})\\
@VVV @AAA @AAA @VVV\\
```

```
\add(\mathcal{L}) @>>> \add(\mathcal{K}) @>>> \cov(\mathcal{K}) @>>>
\non(\mathcal{L})
\end{CD}\end{equation}
```

Standard package info.

```
\NeedsTeXFormat{LaTeX2e}% LaTeX 2.09 can't be used (nor non-LaTeX)
[1994/12/01]% LaTeX date must December 1994 or later
\ProvidesPackage{amscd}[2017/04/14 v2.1 AMS Commutative Diagrams]

\RequirePackage{amsgen}
```

Better not to redefine `\math@cr` if it is already defined, because for CD use only we want to omit the part of the code related to `\dspbrk@lvl` (see `amsmath.sty`).

[mjd,1999/11/04] These definitions have gone somewhat obsolete; but we had probably better leave them as they are for backward compatibility.

```
\@ifundefined{math@cr}{%
  \def\math@cr{{\ifnum0=`}\fi
    \@ifstar{\global\@eqpen\@M\math@cr@}%
          {\global\@eqpen\interdisplaylinepenalty \math@cr@}}
```

The following section merely duplicates some code from the `amsmath` package, in case the `amscd` package is used by itself. For documentation of the code refer to `amsmath.dtx`.

```
  \def\math@cr@{\new@ifnextchar[\math@cr@@{\math@cr@@[\z@]}}
  \def\math@cr@@[#1]{\ifnum0=`{\fi}\math@cr@@@
    \noalign{\vskip#1\relax}}
  \def\restore@math@cr{\def\math@cr@@@{\cr}}
}{}
\restore@math@cr
```

[mjd,1999/11/04] These definitions too are somewhat obsolete; but we had probably better leave them as they are for backward compatibility.

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

\def\atdef@#1{\expandafter\def\csname\space @\string#1\endcsname}
\@ifundefined{Iat}{%
  \DeclareRobustCommand{\Iat}{\FN@\at@}
}{}
```

```
\begingroup \catcode`\@=\active
```

Define math @ to replicate its mathcode-dictated behavior. This is in case @ occurs outside of CD.

```
\csname if\string @compatibility\endcsname
\else \fam=\mathcode`\@
  \xdef @{\mathchar\number\fam\space }
\fi
\gdef\CDat{\let @=\Iat}
\endgroup
\mathcode`\@="8000 % make @ pseudo-active in math
\def\at@{\let\next@\at@@
 \ifcat\noexpand\next a\else
 \ifcat\noexpand\next0\else
 \ifcat\noexpand\next\relax\else
 \let\next@\at@@@\fi\fi\fi\next@}
\def\at@@#1{\expandafter
  \ifx\csname\space @\string#1\endcsname\relax
    \DN@{\at@@@#1}%
  \else
    \DN@{\csname\space @\string#1\endcsname}%
  \fi\next@}%
```

The following items should be defined only if they are not already defined, either to leave the package name untouched (in the case of \PackageError) or to avoid redundant allocation of token or dimen registers.

```
\@ifundefined{default@tag}{%
  \def\default@tag{%
    \def\tag{\PackageError{amscd}{\protect\tag\space not allowed
      here}\@eha}}%
}{}%
\@ifundefined{at@@@}{%
  \def\at@@@{\PackageError{amscd}{\Invalid@@ @}{\the\athelp@}\char64\relax}
}{}
\@ifundefined{athelp@}{\csname newhelp\endcsname\athelp@
{Only certain combinations beginning with @ make sense to me.^^J%
I'll assume you wanted @@ for a printed @.}}{}
\@ifundefined{minaw@}{\newdimen\minaw@}{}
\@ifundefined{bigaw@}{\newdimen\bigaw@}{}
```

Assignment of a couple of dimensions, and initialization of \ampersand@. We check to see if we need to define \minaw@ and \bigaw@.

```
\minaw@11.111pt
\newdimen\minCDarrowwidth
\minCDarrowwidth2.5pc
\newif\ifCD@
\let\ampersand@\relax
```

Added \restore@math@cr\default@tag to fix line numbering problems, 7-JUN-1991. Suggested by I. Zakharevich.

```
\newenvironment{CD}{%
```

```
  \CDat
  \bgroup\relax\iffalse{\fi\let\ampersand@&\iffalse}\fi
  \CD@true\vcenter\bgroup\let\\\math@cr\restore@math@cr\default@tag
  \tabskip\z@skip\baselineskip20\ex@
  \lineskip3\ex@\lineskiplimit3\ex@\halign\bgroup
  &\hfill$\m@th##$\hfill\crcr
}{%
  \crcr\egroup\egroup\egroup
}
```

`\CD@check` This check is used by all macros that must not appear outside the `CD` environment. The first argument is the symbol to be used after `@`, the second one the action to be performed.

Once again we use the trick of defining a temporary control sequence `\next@` and calling it after the final `\fi`. This is not absolutely necessary, but it ensures that the conditional text is processed in one and the same column of the enclosing alignment.

```
\def\CD@check#1#2{\ifCD@\DN@{#2}\else
  \DN@{\PackageError{amscd}{@\protect#1 not
    allowed outside of the CD environment}\@eha}%
  \fi\next@}

\atdef@>#1>#2>{\ampersand@
  \ifCD@ \global\bigaw@\minCDarrowwidth \else \global\bigaw@\minaw@ \fi
  \setboxz@h{$\m@th\scriptstyle\;{#1}\;\;$}%
  \ifdim\wdz@>\bigaw@\global\bigaw@\wdz@\fi
```

If `#2` is empty we can save some work.

```
  \@ifnotempty{#2}{\setbox\@ne\hbox{$\m@th\scriptstyle\;{#2}\;\;$}%
    \ifdim\wd\@ne>\bigaw@\global\bigaw@\wd\@ne\fi}%
 \ifCD@\enskip\fi
   \mathrel{\mathop{\hbox to\bigaw@{\rightarrowfill@\displaystyle}}%
     \limits^{#1}\@ifnotempty{#2}{_{#2}}}%
 \ifCD@\enskip\fi \ampersand@}
%
\atdef@<#1<#2<{\ampersand@
  \ifCD@ \global\bigaw@\minCDarrowwidth \else \global\bigaw@\minaw@ \fi
  \setboxz@h{$\m@th\scriptstyle\;\;{#1}\;$}%
  \ifdim\wdz@>\bigaw@ \global\bigaw@\wdz@ \fi
  \@ifnotempty{#2}{\setbox\@ne\hbox{$\m@th\scriptstyle\;\;{#2}\;$}%
    \ifdim\wd\@ne>\bigaw@ \global\bigaw@\wd\@ne \fi}%
  \ifCD@\enskip\fi
    \mathrel{\mathop{\hbox to\bigaw@{\leftarrowfill@\displaystyle}}%
      \limits^{#1}\@ifnotempty{#2}{_{#2}}}%
  \ifCD@\enskip\fi \ampersand@}
```

Variants of the above two arrows, using `(` and `)` characters instead of `<` and `>` characters, are provided for those whose keyboards don't have the latter.

```
\begingroup \catcode`\~=\active \lccode`\~=`\@
\lowercase{%
```

```
   \global\atdef@)#1)#2){~>#1>#2>}
   \global\atdef@(#1(#2({~<#1<#2<}
 }% end lowercase
 \endgroup

 \atdef@ A#1A#2A{\CD@check{A..A..A}{\llap{$\m@th\vcenter{\hbox
   {$\scriptstyle#1$}}$}\Big\uparrow
   \rlap{$\m@th\vcenter{\hbox{$\scriptstyle#2$}}$}&&}}
 %
 \atdef@ V#1V#2V{\CD@check{V..V..V}{\llap{$\m@th\vcenter{\hbox
   {$\scriptstyle#1$}}$}\Big\downarrow
   \rlap{$\m@th\vcenter{\hbox{$\scriptstyle#2$}}$}&&}}
 %
 \atdef@={\CD@check={&\enskip\mathrel
   {\vbox{\hrule\@width\minCDarrowwidth\vskip2\ex@\hrule\@width
   \minCDarrowwidth}}\enskip&}}
 %
 \atdef@|{\CD@check|{\Big\Vert&&}}
 %
 \atdef@\vert{\CD@check\vert{\Big\Vert&&}}
 %
 \atdef@.{\CD@check.{&&}}
```

The `\pretend` command has weird syntax that doesn't fit well with standard LaTeX syntax so we leave it undone, at least for now. [mjd,1994/10/27]

```
 %\def\pretend#1\haswidth#2{\setboxz@h{$\m@th\scriptstyle{#2}$}\hbox
 % to\wdz@{\hfill$\m@th\scriptstyle{#1}$\hfill}}
```

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
 \endinput
```