

The `amscd`x package

Martin Vermeer*,

based on `amscd` by

Frank Mittelbach, Rainer Schöpf and Michael Downes

Version v2.2x, 2019/07/02

This file is maintained by Martin Vermeer.

1 Introduction

The `amscd` package provides a CD environment that emulates the commutative diagram capabilities of $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ 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`, `xymatrix`, `kuvio`, or `tikz-cd`.

Example:

$$\begin{array}{ccc} S^{\mathcal{W}\lambda} \otimes T & \xrightarrow{j} & T \\ \downarrow & & \downarrow_{\text{End } P} \\ (S \otimes T)/I & \xlongequal{\quad} & (Z \otimes T)/J \end{array} \quad (1)$$

```
\begin{equation}\begin{CD} S^{\{\mathcal{W}\}_\Lambda} \otimes T @>j>> T \\ @VVV @VV{\text{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]:

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

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

*firstname.lastname@aalto.fi

```

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

```

2 Enhancements in amscdx

We implemented double (“fat”) and dashed arrows in both directions, and color attributes for arrows and their annotations. Naming scheme:

- The left-right double arrow @ZZZ
- The up-down double arrow @XXX.

The color, a defined `xcolor` color name, can be added to an arrow by inserting `\CDlor{<color>}` into the diagram before, i.e., to the immediate left of, the arrow that you wish to color. The command affects all arrows following it, so cancel it by the command `\CDlor{black}`. If you want to change the color of all the arrows, you can place the color command outside and before the `CD` environment.

Double arrows are obtained by inserting `\CDFattrue` into the diagram before, i.e., to the immediate left of, the arrow that you wish to be doubled. The command only affects the arrow immediately following it. To change the whole diagram to double arrows, place the `\CDFattrue` command outside and before the `CD` environment. If you do that, you can exempt individual arrows, i.e., make them single again, by the command `\CDFatfalse`.

A similar technique applies for obtaining dashed arrows, but with the boolean (flag) `\CDash`, i.e., the command `\CDashtrue` will give a dashed arrow, while `\CDashfalse` will turn it off. `\CDash` will only work if the state of the `\CDFat` flag be false. If both `\CDFat` and `\CDash` are true, the `CDFat` flag takes precedence.

Dependencies

- the `xcolor` package, for the color;
- the `graphics` package, if you want vertical dashed arrows, as this uses the `\rotatebox` command.

These are now required within the package.

Option

The package takes one option: `lyx`, to be issued if you use it in LyX. Place, in the preamble, the command `\usepackage[lyx]{amscdx}`. This forces a `\renewenvironment{CD}` to be called instead of `\newenvironment{CD}`, as `CD` is already defined.

A demonstration of some of the new capabilities (yes, this looks tacky):

$$\begin{array}{ccc}
 \text{Left header} & & \text{Right header} \\
 TL & \xleftarrow{\text{Atop}} & TR \\
 \begin{array}{c} \uparrow \\ \text{Left} \\ \downarrow \\ N_x \end{array} & & \begin{array}{c} \uparrow \\ \text{Right} \\ \text{column} \\ \downarrow \\ N_x \end{array} \\
 ML & \xleftarrow[\text{pqr}]{xyzxyzxyz} & MR \\
 \begin{array}{c} \uparrow \\ M^2 \\ \uparrow \\ \text{Left} \\ \text{column} \end{array} & & \begin{array}{c} \downarrow \\ + \\ \downarrow \\ N_y \end{array} \\
 BL & \xleftrightarrow[\text{below}]{\text{above}} & BR
 \end{array} \tag{3}$$

```

\begin{equation}
\CDfattrue
\CDlor{green}
\begin{CD}
\text{Left header} @. \text{Right header} \\
TL \CDfatfalse\CDashtrue @Z\text{Atop}Z{\hspace{4cm}}Z TR \\
\CDfatfalse @X\text{Left}X{N_x}X \\
\CDfatfalse \CDashtrue
\CDlor{brown} @A{\scriptsize\begin{array}{r}\text{Right}}\ \\
\text{column}\end{array}}A{N^x}A \\
ML \CDfatfalse \CDashtrue \CDlor{lime} @<{xyzxyzxyz}<<{pqr}< MR \\
\CDlor{red} @A{M^2}A\scriptsize\begin{array}{l}\text{Left}\ \\
\text{column}\end{array}A @V+V{N_y}V \\
BL \CDlor{purple} @Z\text{above}Z\text{below}Z BR \\
\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{amscdx}[2019/07/02 v2.2x AMS Commutative Diagrams]

```

```

\newif\iflyx\lyxfalse
\DeclareOption{lyx}{\lyxtrue}

```

```

\DeclareOption*{}
\ProcessOptions*\relax

```

```

\RequirePackage{amsgen}
\RequirePackage{xcolor}
\RequirePackage{graphicx}

```

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 `amscdx` 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.

```
%
\def\clap#1{\hbox to Opt{\hss#1\hss}}
%
\newif\ifCDfat
\newif\ifCDash
\def\dye@{black}
\def\CDlor#1{\gdef\dye@{#1}} % must be gdef
%
%
\@ifundefined{rightarrowfill@}{
\def\rightarrowfill@#1{%
\m@th\setboxz@h{#1\relbar$}\ht\z@\z@
#1\copy\z@\mkern-6mu\cleaders
\hbox{#1\mkern-2mu\box\z@\mkern-2mu$}\hfill
\mkern-6mu\mathord\rightarrow$}
\def\leftarrowfill@#1{%
\m@th\setboxz@h{#1\relbar$}\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\relbar$}\ht\z@\z@
#1\mathord\leftarrow\mkern-6mu\cleaders
\hbox{#1\mkern-2mu\box\z@\mkern-2mu$}\hfill
\mkern-6mu\mathord\rightarrow$}
}{}

```

Added double arrows

```
\def\Rightarrowfill@#1{%
\m@th\setboxz@h{#1\Relbar$}\ht\z@\z@
#1\copy\z@\mkern-6mu\cleaders
```

```

\hbox{#1\mkern-2mu\box\z@\mkern-2mu}\hfill
\mkern-6mu\mathord\Rightarrow$}
\def\Leftarrowfill@#1{%
\m@th\setboxz@h{#1\Relbar$}\ht\z@\z@
#1\mathord\Leftarrow\mkern-6mu\cleaders%
\hbox{#1\mkern-2mu\copy\z@\mkern-2mu}\hfill
\mkern-6mu\box\z@$}
\def\Leftrightarrowfill@#1{%
\m@th\setboxz@h{#1\Relbar$}\ht\z@\z@
#1\mathord\Leftarrow\mkern-6mu\cleaders
\hbox{#1\mkern-2mu\box\z@\mkern-2mu}\hfill
\mkern-6mu\mathord\Rightarrow$}
%
\def\rightdasharrowfill@#1{%
\m@th\setboxz@h{#1\:\relbar\:\}\ht\z@\z@
#1\copy\z@\mkern-6mu\cleaders
\hbox{#1\mkern-2mu\box\z@\mkern-2mu}\hfill
\mkern-2mu\mathord\rightarrow$}
\def\leftdasharrowfill@#1{%
\m@th\setboxz@h{#1\:\relbar\:\}\ht\z@\z@
#1\mathord\leftarrow\mkern-2mu\cleaders%
\hbox{#1\mkern-2mu\copy\z@\mkern-2mu}\hfill
\mkern-6mu\box\z@$}
\def\leftrightdasharrowfill@#1{%
\m@th\setboxz@h{#1\:\relbar\:\}\ht\z@\z@
#1\mathord\leftarrow\mkern-2mu\cleaders
\hbox{#1\mkern-2mu\box\z@\mkern-2mu}\hfill
\mkern-2mu\mathord\rightarrow$}

\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'\@
\edef @{\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\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{amscdx}{\protect\tag\space not allowed
here}\@eha}}%
}%
\@ifundefined{at@@@}{%
\def\at@@@{\PackageError{amscdx}{\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.

```

\iflyx\else\newenvironment{CD}{\}\fi
\renewenvironment{CD}{%
\CDat
\bgrouper\relax\iffalse{\fi\let\ampersand@&\iffalse}\fi
\CD@true\center\bgrouper\let\\math@cr\restore@math@cr\default@tag
\tabskip\z@skip\baselineskip20\ex@
\lineskip3\ex@\lineskiplimit3\ex@\halign\bgrouper
&\hfill$\m@th##$\hfill\crrcr
}%
\crrcr\egrouper\egrouper\egrouper
}

```

`\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{amscdx}{\protect#1 not
    allowed outside of the CD environment}\@eha}%
  \fi\next@}
```

The boolean `\ifCDFat` has scope only within a table cell:

```
\atdef@>#1>#2>{%
  \ifCDFat\gdef\ar@{\Rightarrowfill@}\else\ifCDash\gdef\ar@{\rightdasharrowfill@}\else\gdef\ar@{\righarrowfill@}
  \gdef\before@{\dye@}
  \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@one\hbox{$\m@th\scriptstyle\;{#2}\;};$}%
  \ifdim\wd@one>\bigaw@\global\bigaw@\wd@one\fi}%
\ifCD@\enskip\fi
\color{\before@}\mathrel{\mathop{\hbox to\bigaw@{\ar@displaystyle}}}%
  \limits^{#1}\@ifnotempty{#2}{_{#2}}}%
\ifCD@\enskip\fi \ampersand@}
%
\atdef@<#1<#2<{%
  \ifCDFat\gdef\ar@{\Leftarrowfill@}\else\ifCDash\gdef\ar@{\leftdasharrowfill@}\else\gdef\ar@{\leftarrowfill@}
  \gdef\before@{\dye@}
  \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@one\hbox{$\m@th\scriptstyle\;{#2}\;};$}%
  \ifdim\wd@one>\bigaw@\global\bigaw@\wd@one \fi}%
  \ifCD@\enskip\fi
  \color{\before@}\mathrel{\mathop{\hbox to\bigaw@{\ar@displaystyle}}}%
  \limits^{#1}\@ifnotempty{#2}{_{#2}}}%
  \ifCD@\enskip\fi \ampersand@}
\atdef@ Z#1Z#2Z{\CD@check{x..x..x}}{%
  \ifCDFat\gdef\ar@{\Leftrightarrowfill@}\else\ifCDash\gdef\ar@{\leftrightrightdasharrowfill@}\else\gdef\ar@{\Leftrightarrowfill@}
  \gdef\before@{\dye@}
  \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@one\hbox{$\m@th\scriptstyle\;{#2}\;};$}%
  \ifdim\wd@one>\bigaw@\global\bigaw@\wd@one \fi}%
  \ifCD@\enskip\fi
  \color{\before@}\mathrel{\mathop{\hbox to\bigaw@{\ar@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}{\color{\dye@}\llap{$\m@th\vcenter{\hbox
  {$\scriptstyle#1$}}$}
  \ifCDFat \Big\Uparrow \else
  \ifCDash \vphantom{\Bigg\Updownarrow}\text{\rotatebox{90}{\clap{$\mkern8mu\relbar\:\relbar\mkern-2mu}}
  \else \Big\uparrow \fi \fi
  \rlap{$\m@th\vcenter{\hbox{$\scriptstyle#2$}}$}&&}}
%
\atdef@ V#1V#2V{\CD@check{V..V..V}{\color{\dye@}\llap{$\m@th\vcenter{\hbox
  {$\scriptstyle#1$}}$}
  \ifCDFat \Big\Downarrow \else
  \ifCDash \vphantom{\Bigg\Updownarrow}\text{\rotatebox{270}{\clap{$\mkern-8mu\relbar\:-\mkern-2mu}}
  \else \Big\downarrow \fi \fi
  \rlap{$\m@th\vcenter{\hbox{$\scriptstyle#2$}}$}&&}}
%
\atdef@ X#1X#2X{\CD@check{X..X..X}{\color{\dye@}\llap{$\m@th\vcenter{\hbox
  {$\scriptstyle#1$}}$}
  \ifCDFat \Big\Updownarrow \else
  \ifCDash \vphantom{\Bigg\Updownarrow}\text{\rotatebox{90}{\clap{$\mkern8mu\leftarrow\mkern-2mu}}
  \else \Big\updownarrow \fi \fi
  \rlap{$\m@th\vcenter{\hbox{$\scriptstyle#2$}}$}&&}}
%
\atdef@={\CD@check={\gdef\before@{\dye@}&\color{\before@}\enskip\mathrel
  {\vbox{\hrule\@width\minCDarrowwidth\vskip2\ex\@hrule\@width
  \minCDarrowwidth}}\enskip&}}
%
\atdef@|{\CD@check|{\color{\dye@}\Big\Vert&&}}
%
\atdef@\vert{\CD@check\vert{\color{\dye@}\Big\Vert&&}}
%
\atdef@.{\CD@check.{&&}}

```

The `\pretend` command has weird syntax that doesn't fit well with standard L^AT_EX 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

```