The `tikz-cd` package

Florêncio Neves*

Version 0.1†

The general-purpose drawing package TikZ can be used to typeset commutative diagrams and other kinds of mathematical pictures, generating high-quality results (see for example [1] or [2]). The purpose of this package is to make the process of creation of such diagrams easier by providing a convenient set of macros and reasonable default settings. This package also includes an arrow tip library that match closely the arrows present in the Computer Modern typeface.

PGF version 2.10 is required.

Contents

1 Basic usage

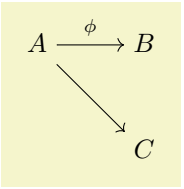
Commutative diagrams are created with the `tikzcd` environment. Its content describes a matrix, like the `\matrix` command in TikZ or the `align` environment in LaTeX. Everything is typeset in math mode, but you will probably want use `tikzcd` inside an `equation` environment or inside `\[ \]`, so that the diagram will be placed on a new line and centered.

* *E-mail:* florencioneves@gmail.com

† This is a preliminary version. Future versions may not be backwards-compatible. Comments are welcome.

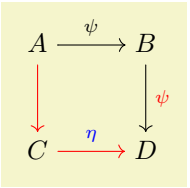
1.1 Inserting arrows

Inside the `tikzcd` environment, the command `\arrow` is provided to allow insertion of arrows. In its simplest form, it takes one argument, a string containing the characters `l`, `r`, `u` or `d`, standing for left, right, up and down, that determine the arrow target. A label can be placed on this arrow by providing a second argument.



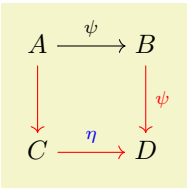

```
\begin{tikzcd}
  A \arrow{ld} \arrow{l}{\phi} & B \\
                               & C
\end{tikzcd}
```

You can control the behavior of the arrow by placing an argument inside square braces before the direction parameter. It may contain anything that can be passed as an argument to a Ti*k*Z's `edge` operator. Similarly, the label can be modified by an argument in square braces right before it. It may contain anything that can be passed to a `node` operator.




```
\begin{tikzcd}
  A \arrow{l}{\psi} \arrow[color=red]{d}
        & B \arrow{d}[color=red]{\psi} \\
  C \arrow[color=red]{l}[color=blue]{\eta}
        & D
\end{tikzcd}
```

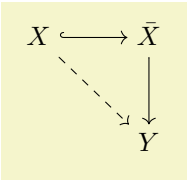
If you want to save typing, the command `\ar` is provided as a shortcut to `\arrow`. There are also commands `\lar`, `\rar`, `\uar` and `\dar`, that act like `\arrow{l}`, `\arrow{r}` and so forth. They can take up to three optional argument: one in square braces to control the arrow, one in square braces to control the label, and one in curly braces to specify a label. Thus, the previous diagram can be rewritten as follows.




```
\begin{tikzcd}
  A \lar{\psi} \dar[color=red]
        & B \dar[color=red]{\psi} \\
  C \lar[color=red][color=blue]{\eta}
        & D
\end{tikzcd}
```

1.2 Changing arrow tips

If you are familiar with Ti*k*Z, you certainly noticed that you can produce different kids of arrows by passing arguments like `right hook->` to the `\arrow` command. This package provides an alternative way of doing this. Namely, there are styles named after LaTeX arrow producing commands for this purpose. Thus, instead of `\arrow[right hook->]{l}`, you can use `\arrow[hookrightarrow]{l}`, as in the example below.




```
\begin{tikzcd}
  X \arrow[hookrightarrow]{l} \arrow[dashed]{ld}
                          & \bar{X} \arrow{d}\\
                          & Y
\end{tikzcd}
```

The complete list of arrow types available so far is given below. It will be made comprehensive in the future.

`rightarrow`	yelds $\rightarrow$
`leftarrow`	yelds $\leftarrow$
`hookrightarrow`	yelds $\hookrightarrow$
`hookleftarrow`	yelds $\hookleftarrow$
`hookrightarrow`	yelds $\hookrightarrow$
`rightharpoonup`	yelds $\rightharpoonup$
`rightharpoondown`	yelds $\rightharpoondown$
`leftharpoonup`	yelds $\leftharpoonup$
`leftharpoondown`	yelds $\leftharpoondown$

2 Changing the appearance of diagrams

In the future, a customization scheme using `pgfkeys` will be implemented. For now, you can change the appearance of diagrams by understanding some of `tikz-cd`'s internals. The `tikzcd` environment generates codes with the following basic structure:

```
\begin{tikzpicture}[ ... ]
  \matrix (m) [ ... ] {
      &  ... & \\
      &  ... & \\};
  \path[ ... ] ...;
\end{tikzpicture}
```

There are styles

`/commutative diagrams/picture style`,
`/commutative diagrams/matrix style`, and
`/commutative diagrams/path style`

that are applied at the relevant places. By appending things to these styles, you can control the behavior of diagrams quite arbitrarily. The arrow tip styles are stored in `/commutative diagrams/current arrows`.

3 Computer Modern arrow tips

By using the mechanism explained in §1.2, it is not necessary to know the technical details in this section.

The naming scheme of the Computer Modern-like arrow tips provided by this package follows that of `pgf`'s `arrows` library, as described in Section 23 of the `pgf` manual. You can of course use these arrow tips outside the `tikzcd` environment; in this case, you will probably want to know that in order to match the Computer Modern font at size 10 pt, it is necessary to set `line width` to 0.4 pt; for larger font sizes, scale this parameter accordingly.

Basic arrow tips

`cm to`	yelds $\longleftrightarrow$
`cm to reversed`	yelds $\longleftrightarrow$
`cm bold to`	yelds $\longleftrightarrow$ (with a line 50% thicker)
`cm |`	yelds $\mapsto\!\!\!-\!\!\!\mapsfrom$
`cm o`	yelds $\circ\!\!-\!\!\circ$
`cm *`	yelds $\bullet\!\!-\!\!\bullet$
`cm cap`	yelds $\frown$

Hooks

`cm left hook`	yelds $\hookleftarrow$
`cm right hook`	yelds $\hookrightarrow$

Double arrow tips

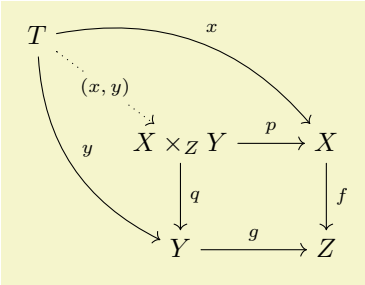
`cm double to`	yelds $\Longleftrightarrow$
`cm double to reversed`	yelds $\Longleftrightarrow$

Partial arrow tips

`cm left to`	yelds $\leftarrow\!\!-$
`cm left to reversed`	yelds $\leftarrow\!\!-$
`cm right to`	yelds $-\!\!\rightarrow$
`cm right to reversed`	yelds $-\!\!\rightarrow$

4 Some examples

The following example is taken from [2].



```
\begin{tikzcd}
  T
  \arrow[bend left]{dll}{x}
  \arrow[bend right]{ddl}{y}
  \arrow[dotted]{dl}[description]{(x,y)} & & \\
      & X \times_Z Y \arrow{l}{p} \arrow{d}{q} & X \arrow{d}{f} \\
      & Y \arrow{l}{g}                         & Z
\end{tikzcd}
```

References

[1] Felix Lenders, *Commutative diagrams using TikZ*. Available at http://www.felixl.de/commu.pdf.

[2] James Milne, *Guide to commutative diagrams*. Available at http://www.jmilne.org/not/CDGuide.html.