# {cd-short}:

## A collection of shortcuts for {tikz-cd}.

March 19, 2022

Catherine Li

Abstract. The package {cd-short} provides a collection of custom commands and shortcuts for {tikz-cd}, in particular ones useful for category theory. These shortcuts and their usage are recorded in the following document.

Contents



Note that this package requires {tikz-cd}, {xparse}, and {wrapfig}.

## 1. Arrows

This section discusses arrow shortcuts. Visual depictions in the context of completed diagrams are provided in Section 2, as they will require the newly defined environment shortcuts as weell.

### 1.1. The basic diagram arrow shortcut

The following command is used to produce an arrow in a tikzcd environment:

\arr[⟨*options*⟩]{⟨*start*⟩}{⟨*end*⟩}⟨*labels*⟩

where a label in ⟨*labels*⟩ is of the form

[⟨*label options*⟩]^{⟨*label text*⟩} or [⟨*label options*⟩]_{⟨*label text*⟩},

depending respectively on whether the label is above or below the arrow (when pointing right). Additionally, {⟨*start*⟩} and {⟨*end*⟩} are the absolute positions of the starting point and end point of an arrow, and are written as as ⟨*row*⟩-⟨*column*⟩ when viewing the nodes of the diagram as a matrix.

For example, one might wish write something like

```
\arrow[dashed, from = 1-1, to = 1-2, "\phi" red, "g"']
```

in {tikz-cd} traditionally. Via this shortcut, the above becomes

\arr[dashed]{1-1}{1-2}[red]^\phi_g.

Note that this shortcut only admits two labels, one above and one below. Additionally, it inherits all arrow and label options directly from {tikz-cd}.

### 1.2. Shortcuts for common diagram arrows

Certain sorts of arrows are used very often in dealing with model categories. The chosen commands for these arrows mostly reflect that terminology. Replace \arr with any of the commands listed in the following table to obtain the indicated basic arrow shape.

| Command | Result | Command | Result | Command | Result |
|---|---|---|---|---|---|
| \arr | $\longrightarrow$ | \maps | $\longmapsto$ | \we | $\xrightarrow{\sim}$ |
| \cof | $\hookrightarrow$ | \fib | $\twoheadrightarrow$ | \fibcof | $\hookrightarrow\!\!\!\!\twoheadrightarrow$ |
| \acof | $\xhookrightarrow{\sim}$ | \afib | $\xrightarrow{\sim}\!\!\!\!\twoheadrightarrow$ | \afibcof | $\xhookrightarrow{\sim}\!\!\!\!\twoheadrightarrow$ |
| \rra | $\longleftarrow$ | \leads | $\rightsquigarrow$ | \sub | $\dashrightarrow$ |
| \Arr | $\Longrightarrow$ | \Maps | $\Longmapsto$ | \Eq | $=\!\!=$ |

### 1.3. In-line arrows

This package also defines the same arrows in-line. Add 'r' (or 'l') after the \ in each of the commands above (e.g. \rarr or \larr) to obtain the in-line version (or reversed in-line version). Indeed, each arrow will take the form

\⟨*command*⟩[⟨*length*⟩]^{⟨*label*⟩}_{⟨*label*⟩}.

The default length is 1.5em. Note that individual label options are not available, and ^ determines the top label and _ determines the bottom label regardless of direction. Then the example

$$\ldots \longrightarrow A \overset{f}{\underset{g}{\rightleftarrows}} B \xleftarrow[\sim]{\text{a long label}} C \overset{\psi}{=\!=} D \rightsquigarrow E \to \ldots$$ is produced by

```
\[\ldots \rarr A \rafibcof^f_g B \lwe[5em]^{\text{a long label}} C %
  \rEq^\psi D \rleads E \rarr[1em] \ldots\].
```

The table below displays a full list of these arrow commands.

| Command | Result | Command | Result | Command | Result |
|---|---|---|---|---|---|
| \rarr | $\longrightarrow$ | \rmaps | $\longmapsto$ | \rwe | $\xrightarrow{\sim}$ |
| \larr | $\longleftarrow$ | \lmaps | $\longmapsfrom$ | \lwe | $\xleftarrow{\sim}$ |
| \rcof | $\hookrightarrow$ | \rfib | $\twoheadrightarrow$ | \rfibcof | $\hookrightarrow\!\!\!\!\twoheadrightarrow$ |
| \lcof | $\hookleftarrow$ | \lfib | $\twoheadleftarrow$ | \lfibcof | $\twoheadleftarrow\!\!\!\!\hookleftarrow$ |
| \racof | $\xhookrightarrow{\sim}$ | \rafib | $\xrightarrow{\sim}\!\!\!\!\twoheadrightarrow$ | \rafibcof | $\xhookrightarrow{\sim}\!\!\!\!\twoheadrightarrow$ |
| \lacof | $\xhookleftarrow{\sim}$ | \lafib | $\xleftarrow{\sim}\!\!\!\!\twoheadleftarrow$ | \lafibcof | $\xhookleftarrow{\sim}\!\!\!\!\twoheadleftarrow$ |
| \rrra | $\longleftarrow$ | \rleads | $\rightsquigarrow$ | \rsub | $\dashrightarrow$ |
| \lrra | $\longrightarrow$ | \lleads | $\leftsquigarrow$ | \lsub | $\dashleftarrow$ |
| \rArr | $\Longrightarrow$ | \rMaps | $\Longmapsto$ | \rEq | $=\!\!=$ |
| \lArr | $\Longleftarrow$ | \lMaps | $\Longmapsfrom$ | \lEq | $=\!\!=$ |

## 2. Environments

In this section, we discuss various environment shortcuts; in particular, we discuss a general shortcut for a centered, displayed `{tikz-cd}` diagram, shortcuts for in-line and wrapped diagrams, and shortcuts for commonly seen diagrams (squares, squares with lifts, etc.).

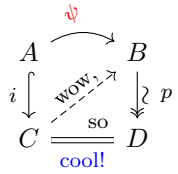
### 2.1. Making a basic display-style diagram

To produce a centered, display-style diagram, use the following environment:

```
\begin{cd}[⟨options⟩]
  ⟨environment content⟩
\end{cd}
```

Again, all options are inherited directly from `{tikz-cd}`.

Now consider the example diagram to the right. Below we have displayed its construction in both `{cd-short}` and `{tikz-cd}`; note that arrows with an additional label `\sim` are constructed using an additional invisible arrow.




| {cd-short} | {tikz-cd} |
|---|---|
| `\begin{cd}` | `\[\begin{tikzcd}` |
| `  A & B \\` | `  A & B \\` |
| `  C & D` | `  C & D` |
| `  \arr[bend left]{1-1}{1-2}%` | `  \ar[bend left, from = 1-1, to = 2-2,%` |
| `  [rotate=30, red]^\psi` | `  "\psi" {rotate = 30, red}]` |
| `  \cof{1-1}{2-1}_i` | `  \ar[hook, from = 1-1, to = 2-1, "i"']` |
| `  \afib{1-2}{2-2}^p` | `  \ar[two heads, from = 1-2, to = 2-2,%` |
| `  \Eq{2-1}{2-2}[near end]%` | `  "p" inner sep = 2ex]` |
| `  ^{\text{so}}[blue]%` | `  \ar[phantom, from = 1-2, to = 2-2,%` |
| `  _{\text{cool!}}` | `  "\mathclap{\sim}" inner sep = 1ex,%` |
| `  \sub[sloped]{2-1}{1-2}%` | `  sloped, shift left]` |
| `  [inner sep=1pt]%` | `  \ar[equal, from = 2-1, to = 2-2,%` |
| `  ^{\text{wow,}}` | `  "\text{so}" near end, "\text{cool!}"%` |
| `\end{cd}` | `  blue]` |
| | `  \ar[dashed, sloped, from = 2-1, to = 1-2%,` |
| | `  "\text{wow,}" inner sep = 1pt]` |
| | `\end{tikzcd}\]` |

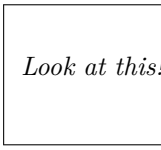
### 2.2. Shortcuts for diagram positioning

This document contains four general diagram environments: `cd`, the basic display-style diagram (which we saw above); `il-cd`, an in-line diagram; `rw-cd`, a text-wrapped diagram on the right; and `lw-cd`, a text-wrapped diagram on the left.

For example, replace `cd` from above with `il-cd` to obtain an in-line diagram environment, so that

```
\textit{Look at this!}
\begin{il-cd}
  A & B \\
  C &
  \arr{1-1}{1-2}
  \arr{1-1}{2-1}
  \sub{2-1}{1-2}
\end{il-cd}
\textit{Isn't that nice?}
```

becomes

*Look at this!*  *Isn't that nice?*

Additionally, we can replace `cd` with `lw-cd` or `rw-cd` to obtain a text-wrapped diagram on the left or right, respectively. More specifically, we gain additional arguments

```
\begin{lw-cd}[⟨options⟩]"⟨caption⟩"
  ⟨environment content⟩
\end{lw-cd}
```

where ⟨*options*⟩ are standard options for a `{tikz-cd}` diagram as above, and ⟨*caption*⟩ is an optional text providing a caption for the diagram. Note that this environments defaults the width and height of the diagram to be the smallest possible, according to `{wrapfig}`. Now consider the following example:

```
\begin{rw-cd}[row sep = 1 em, column sep = 20em]"A very fat (or long) square."
    A & B \\
    C & D
    \arr{1-1}{1-2}
    \arr{1-1}{2-1}
    \arr{1-2}{2-2}
    \arr{2-1}{2-2}
\end{rw-cd}
⟨a paragraph of random text, denoted below by quotation marks⟩
```

yields the following (not including the additional left-hand margin):

> "This is a random paragraph, full of nonsense words. Indeed, the square was made particularly fat so that I'd have to type fewer random words. Placeholder text packages don't work well with `{wrapfig}`, so it's just me typing over here. Whoopee. This is a random paragraph, full of nonsense words. Indeed, the rectangle was made particularly fat so that I'd have to type fewer random words. Placeholder text packages don't work well with `{wrapfig}`, so it's just me typing over here. Whoopee. This is a random paragraph, full of nonsense words."

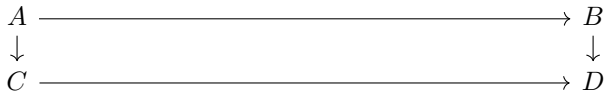



FIGURE 1: A very fat (or long) square.

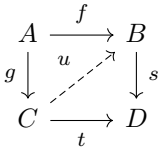
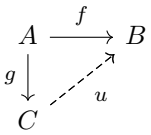
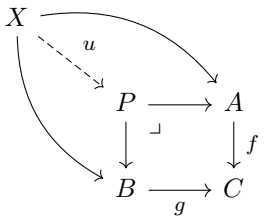
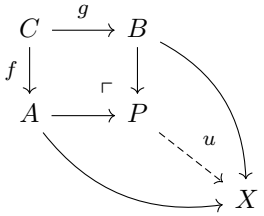
Of course, `lw-cd` works the same way.

### 2.3. Shortcuts for common diagrams

Here we discuss shortcuts for common diagram shapes. All diagrams provided by these shortcuts are in-line by default. Simply enclose them with `\[\]` to obtain a display-style diagram. All arrow labels are optional, i.e. one only needs to provide the mandatory arguments `{`⟨*object*⟩`}{`⟨*object*⟩`}...{`⟨*object*⟩`}` to obtain the diagram.

There is an fairly easy pattern to these shortcuts. In general, an arrow's label follows its domain; additionally, a label is preceded by `^` or `_` depending on whether its arrow is in the "upper" or "lower" direction of the diagram (i.e. clockwise/top-right vs. counterclockwise/left-bottom, respectively). These are reversed when considering the dual case, i.e. arrow labels follow codomains, `^` and `_` are switched. Furthermore, there is one special case: lifts/extensions always come last and are preceded by `/`.

The following table displays all available shortcuts thus far.

| Command | Description | Result |
|---|---|---|
| `\sqr{A}^f_g{B}^s{C}_t{D}/u` | The lift is optional; excluding `/{`⟨*label*⟩`}` removes the arrow from the diagram. If you want an unlabeled lift, write `/{}`. |  |
| `\tri{A}^f_g{B}{C}/u` | To produce a solid arrow instead of the dashed extension, write `_{`⟨*label*⟩`}` instead of `/{`⟨*label*⟩`}`. |  |
| `\pullback{A}^f{C}{B}_g{P}[X]/u` | Excluding `[X]/u` would return just the square in the lower right. Note that the order of arguments `{A}{C}{B}` corresponds to the diagram $A \to C \leftarrow B$. |  |
| `\pushout{A}_f{C}{B}^g{P}[X]/u` | Excluding `[X]/u` would return just the square in the upper left. Note that the order of arguments `{A}{C}{B}` corresponds to the diagram $A \leftarrow C \to B$. |  |