

The DuckBoat — Beginners' Pond: CDs, but not Compact Disks

Herr Professor Paulinho van Duck

Abstract

In this installment, Prof. van Duck will show you some tips & tricks about `tikz-cd`, a package for high-quality typesetting of commutative diagrams.

1 \TeX .SE has a new moderator!

Hi, \mathbb{L} \TeX friends!

An exceptional event took place last November: a \TeX .SE moderator's election, quack!

Moderators are elected for life because they are somewhat compared to Supreme Court judges, who are appointed lifelong (in some countries, at least). Therefore, their elections are quite infrequent.

Of course, the role of moderator is not a life sentence. There is also the possibility of resigning.

Our former moderator Martin Scharrer made that choice last October. Martin had been in that role since \TeX .SE left beta in 2011. He is also the author of many valuable packages, like `standalone` and `adjustbox`. We are immensely grateful to him for all his past and future contributions to the $\mathbb{L}\text{\TeX}$ community.

I would also like to thank the friends who nominated themselves as candidates in the moderator election. There were a couple of well-known names but even relatively new users. I am particularly glad to have seen people from non-Western cultures, and I hope to have a woman, too, next time.

It was very hard to choose. Eventually, the winner was Werner Grundlingh! (Simply known as *Werner* on \TeX .SE: tex.stackexchange.com/users/5764/werner).

He needs no introduction for the \TeX .SE users since he is the third in the site ranking, after egreg and David Carlisle. He has always stood out for his patience, good temper, and his craving for helping others. Let me congratulate him. He has the skills to be a great mod, quack!



Unfortunately, there is also bad news. Our dear friend Brent Longborough passed away in December 2021. He gave an outstanding contribution both on \TeX .SE and with his packages and collaborations (for example, `arara`).

I like to remember his great sense of humor. He defined himself “old-ish IT geezer, young at heart.”

We will miss him!



Herr Professor Paulinho van Duck

doi.org/10.47397/tb/43-1/tb133duck-tikz-cd

As for TopAnswers \TeX , <https://topanswers.xyz/tex>, the site grows and starts appearing in browsers' search results. I suggest you try it. In particular, if your question on other sites remains unanswered, you can be sure it will be looked after on TopAnswers \TeX . Moreover, if you have problems with `beamer`, you will find the great expert samcarter answering there.

I take the occasion to thank her for her help setting the correct fonts in the example of Box 14.



If you are talking with a mathematician about CDs, you almost certainly are not referring to an old-fashioned(!) way to listen to music.

The acronym CDs stands for Commutative Diagrams. My math colleagues say they are graphs largely used in category theory. I do not know what category theory is, but I did not ask, in order to avoid a two-hour math lesson, quack!

However, judging by the number of questions about them, they are undoubtedly prevalent.

Box 1 shows a little example. It is the representation of $h \circ f = k \circ g$.

Box 1 – A commutative diagram

```

...
\usepackage{tikz-cd}
...
\begin{document}
\[
\begin{tikzcd}
A \ar[r, "f"] \ar[d, "g"] & B \\
C \ar[r, "k"] \ar[u, "h"] & D
\end{tikzcd}
\]
\end{document}

```

$$\begin{array}{ccc}
 A & \xrightarrow{f} & B \\
 g \downarrow & & \downarrow h \\
 C & \xrightarrow{k} & D
 \end{array}$$

Do you find it beautiful? Of course, you do!

In the following, I will show you how to create it with `tikz-cd`. There are also other packages for drawing commutative diagrams, for instance `xy`, but in my opinion `tikz-cd` is the simplest and the most customizable.

2 Quack Guide No. 7

A package for typing beautiful CDs

In a previous article of mine [1], I told you about `TikZ` matrices. Simply speaking, a `TikZ` CD is a `TikZ` matrix with some arrows added.

Let us start examining the code in Box 1.

First of all, you have to load the package. Alternatively, if you have already loaded TikZ in your preamble, you may take advantage of the `cd` library:

```
\usepackage{tikz}
\usetikzlibrary{cd}
```

Now you are ready to use the `tikzcd` environment. Please note that I have included it within `[...]` to show it as a displayed equation.

The core of your CD is typed like an ordinary `array`, with `&` to separate the cells (the vertices of the diagram) and `\\` to separate the rows. Pay attention that the last row does not end with `\\`; otherwise, you will get an undesired empty line.

The new additions are the commands `\ar`, an abbreviation for `\arrows`. As it is obvious from their name, they allow you to draw your arrows, quack!

The first parameter of `\ar` is the direction: `r`, `l`, `d`, and `u`, which stand for right, left, down, up. You can combine two or more directions. For example, `dr` means your arrow will go diagonally from the current cell, where the `\ar` is positioned, to the one down and right from it. You can also repeat a direction. For instance, `rr` will go from the current cell to the second cell right to it.

What appears between quotes are the labels of the arrows. The option `swap` makes the label be placed on the right side of the arrow, relative to its direction (left is the default).



I will describe how to build and customize your diagram with the available options in the following.

Remember that, in general, options can be set either for a single element; or for the current CD as environment options:

```
\begin{tikzcd}[<options>]
```

or for all the CDs of your document by putting:

```
\tikzcdset{<options>}
```

in your preamble.

See Section 2 of the package documentation [3] for further details.

2.1 About vertices

Since a TikZ CD is actually a TikZ matrix, it inherits all matrix characteristics.

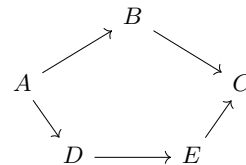
You can set the distance separating columns with `&[<width>]` and for rows with `\\[<width>]`.

If you would like to change these separators for all the rows/columns of your diagram, you can set them with the options `column sep/row sep`.

You may indicate either an explicit dimension or the predefined ones: `tiny`, `small`, `scriptsize`, `normal`, `large`, and `huge`.

Box 2 – Column and row separators

```
\begin{tikzcd}[column sep=4pt,
row sep=scriptsize]
&&[3pt]B\ar[dr]&&[3pt]\\
A\ar[urr]\ar[dr]&&&&C\\[2pt]
&&D\ar[rr]&&E\ar[ur]
\end{tikzcd}
```

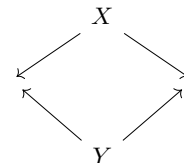


Please note how, in Box 2, the empty cells are used to position the vertices correctly. But be careful that if a cell is a target point of an arrow, it must have a text. Otherwise, you will get the error:

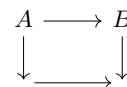
No shape named ... is known.

If you want to have an arrow pointing to an empty vertex, use `{}` for one vertex at a time, or the matrix option `nodes in empty cells`, for all the vertices at once.

```
\begin{tikzcd}
&&X\\
&&\\
&&Y
\end{tikzcd}
```



```
\begin{tikzcd}[
every matrix/.append
style={nodes in
empty cells}
]
A\ar[r]\ar[d]&&B\ar[d]
\ar[r]&&
\end{tikzcd}
```



If you would like to customize one vertex (for example, to draw it), you can use `|[<options>]|` before the node text:

```
\begin{tikzcd}
A\ar[r]&&|[draw]|B
\end{tikzcd}
```



If you would like to modify all the vertices of your diagram, you can use `cells={<options>}`, which is equivalent to `every cell/.append style={<options>}`

whereas

`every cell/.style={\options}`
replaces the default style of the vertices instead of adding options to it.

```
\begin{tikzcd}[
  cells={nodes=draw}]
  A\ar[r]&B
\end{tikzcd}
```



One useful node option is `alias=\langle cell name \rangle`. It allows you to refer to the vertex with `\langle cell name \rangle` when you are drawing your diagram.

Another way for referring to the vertices is `\langle matrix name \rangle-\langle row number \rangle-\langle column number \rangle` where `\langle matrix name \rangle` could be

`\tikzcdmatrixname`

or a name you set with

```
every matrix/.append style=
  {name=\langle matrix name \rangle}
```

Within a path you can also use `\tikztostart` (starting point of the path) and `\tikztotarget` (target point of the path).

In Section 2.4 I will show how node naming is indispensable for drawing complex diagrams.

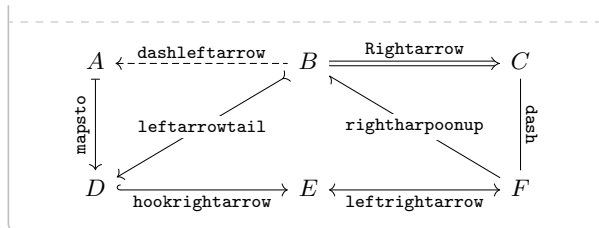
2.2 Go wild with the arrows

The package `tikz-cd` allows a huge number of arrow tips and shapes.

Box 3 shows some examples. For the complete list, see Section 1.3 of the package documentation.

Box 3 – Some arrow types

```
\begin{tikzcd}[
  labels={font=\ttfamily\scriptsize},
  column sep=7em, row sep=9ex
]
A\ar[r, dashleftarrow,
  "\text{dashleftarrow}"]
\ar[d, mapsto, "\text{mapsto}"
  {rotate=90, anchor=south}]&
B\ar[r, Rightarrow, "\text{Rightarrow}"]&
C\ar[d, dash, "\text{dash}"
  {rotate=-90, anchor=south}]\
D\ar[r, hookrightarrow,
  "\text{hookrightarrow}", swap]
\ar[ur, leftarrowtail,
  "\text{leftarrowtail}" description]&
E\ar[r, leftrightarrow,
  "\text{leftrightarrow}", swap]&
F\ar[ul, rightharpoonup,
  "\text{rightharpoonup}" description]
\end{tikzcd}
```



You can also set options for arrows, at either the environment or document level, with `arrows=\langle options \rangle` which appends options to the default, or `every arrow/.style=\langle options \rangle` which replaces the default.

Let us examine some of them (label options will be explained in the next Section).



If the arrows of `tikz-cd` are not enough for you, you may use all the types provided by TikZ, setting `arrow style=tikz`.

For example, if you would like to have stealth arrows throughout your document, you can set

```
\usetikzlibrary{arrows.meta}
\tikzcdset{
  arrow style=tikz,
  arrows={>={Stealth}}
}
```

in your preamble, and all your arrows will appear like this:



When your CD has many vertices, it may be boring or difficult to indicate the correct number of `r`, `l`, `d`, and `u` you need to reach the target point.

The arrow options `to/from=\langle argument \rangle` may help you. The argument could be in the form `\langle row number-column number \rangle`; a string of `r`, `l`, `d`, and `u`; or `\langle cell name \rangle`.

You may even use `to` and `from` together and draw your arrows at the end of the diagram.

The following code is an alternative way to produce the diagram in Box 2:

```
\begin{tikzcd}[
  column sep=3pt,
  row sep=scriptsize
]
&&[3pt]B\ar[dr]&&[3pt]
A\ar[to=1-3]&&&
|[alias=nodeC]|C\ar[from=nodeE]\|[2pt]
&&D\ar[from=ul]&&
|[alias=nodeE]|E\ar[from=3-2]
\ar[from=1-3, to=nodeC]
```

```
\end{tikzcd}
```



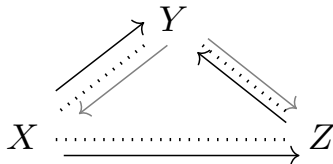
It may happen that you need to shift your arrows to better position them. You can easily do this with `shift left/right=<dimension>` (the parameter is optional).

It is also possible to use `xshift=<dimension>`, `yshift=<dimension>` or `shift={<coordinate>}`.

Box 4 shows some shiftings. The dotted lines represent the default positions. Some arrows are drawn in gray and the CD is enlarged with `\scalebox` to better visualize the positioning. For the `ampersand` replacement option see Section 2.4, Box 14.

Box 4 – Shifting

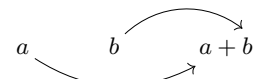
```
\scalebox{1.5}{%
\begin{tikzcd}[ampersand replacement=&]
  \&Y\ar[dr, shift left, gray]
  \ar[from=dr, shift left]
  \ar[dr, dash, dotted, thick]
  \ar[d1, xshift=1.5ex, gray]
  \ar[from=d1, yshift=1.5ex]
  \ar[d1, dash, dotted, thick]\\
  X\ar[rr, shift={ (2pt,-4pt) }]
  \ar[rr, dash, dotted, thick] \&\&Z
\end{tikzcd}%
}
```



You can also explicitly set a starting/ending point with `start/end anchor=` `{[<coordinate transformations>]}<anchor>`

Box 5 – Start/end anchor

```
\begin{tikzcd}
  a\ar[start anchor={ [xshift=-2pt,
  yshift=2pt]south east},
  end anchor=210,bend right=30, rr}
  & b\ar[start
  anchor={ [shift={ (-2pt,-2pt) } ]north
  east},
  end anchor=50,bend left=40,r}
  & a+b
\end{tikzcd}
```



As you see in Box 5, if you are tired of straight arrows, you can use options inherited from TikZ to bend them:

- `bend right/left=<angle>`, to curve the arrow
- `out/in=<angle>`, to set the angle at which the arrow leaves/reaches the vertices
- `loop`, possibly with `above/right/below/left` or setting the in/out angles
- `looseness=<number>` to choose the “level of bending”.

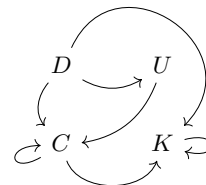
For more details, see Sections 74.3 and 74.4 of the TikZ manual [4].

Box 6 shows some of these options. Please note the environment option `bezier bounding box`. It is useful to set the correct bounding box when there is `looseness` in order to avoid unwanted empty space around your diagram. To use it just add the `bbox` TikZ library [2] in your preamble:

```
\usetikzlibrary{bbox}
```

Box 6 – Bent arrows

```
\begin{tikzcd}[bezier bounding box]
  D\ar[r, bend right]
  \ar[dr, bend left=100, looseness=2]
  \ar[d, out=230, in=130]
  & U\ar[d1, bend left]\\
  C\ar[r, bend right=70]
  \ar[loop, in=180,out=210,looseness=8]
  & K\ar[loop right]
\end{tikzcd}
```



Another useful option is `shorten=<dimension>`. It allows to shorten or, if `<dimension>` is negative, elongate your arrow.

This option acts on both sides of the arrows, if you would like to shorten/elongate only one side, use the standard TikZ options `shorten >=<dimension>` or `shorten <=<dimension>`.

Box 7 – Shorten/elongate arrows

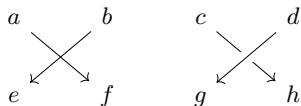
```
\scalebox{1.5}{\begin{tikzcd}[ampersand
replacement=\&]
A\ar[r, dash, dotted, thick]
\ar[r, shift left=1.5mm, shorten=.2cm]
\&B\ar[l, shift left=1.5mm, shorten=-.1cm,
gray]
\&C\ar[r, dash, dotted, thick]
\ar[r, shift left=1.5mm, shorten >=.2cm]
\&D\ar[l, shift left=1.5mm, shorten
<=-.1cm, gray]
\end{tikzcd}}
```



Finally, it is possible to use **crossing over** to show an arrow in foreground. Compare:

Box 8 – Crossing over

```
\begin{tikzcd}
a\ar[dr]&\&b\ar[dl]&
c\ar[dr]&\&d\ar[dl, crossing over]&
e&\&f&
&\&g&\&h
\end{tikzcd}
```



2.3 Adding labels

The general syntax for labels is `"{label text}"{label options}` where the curly braces are mandatory only if `label text` or `label options` contain a comma.

You can set the label options at the environment level specifying `labels=<options>` which appends the style to the default, or `every label/.style=<options>` which replaces the default style.

When including them in `\tikzcdset` in your preamble, they will be valid at the document level, as usual.

In Box 3, for example, I set the label font at the environment level. Note that since the labels are typed in math mode by default, to have them in text mode you have to use `\text{label text}`.

The options `rotate=<degree>` and `anchor=<anchor>` are necessary to put the label along the vertical ar-

rows. If you prefer to write them over the arrow, just use `description` (label text with a white background). You can change the background color with `background color=<color>` or do not have it at all using `marking`.

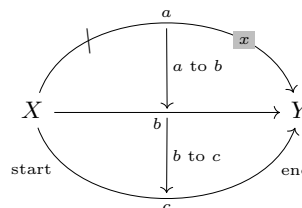
The TikZ options for positioning nodes along a path could also be used (`near start`, `near end`, and so on).

Since labels are nodes, you can even name them and use them as starting or target point of an arrow.

Please note that you can also have more than one label on the same arrow.

Box 9 – Labels

```
\begin{tikzcd}[column sep=3cm]
X\ar[r, "b"{name=B, below left},
" "{name=middle, inner sep=0pt}]
\ar[r, bend left=70, "a"{name=A,
"/"{marking, near start},
"x"{description, near end, background
color=lightgray}]
\ar[r, bend right=70, "c"{name=C, below},
"\text{start}"{very near start, swap},
"\text{end}"{very near end, swap}]
& Y
\ar[from=A, to=middle, shorten <=2pt,
"a\text{ to }b"]
\ar[from=middle, to=C, shorten =2pt,
"b\text{ to }c"]
\end{tikzcd}
```



See Section 2 of the package documentation [3] for further details.

2.4 Other tips & tricks

Sometimes your vertices are not simply letters but longer expressions. Maybe you do not like the standard positioning:

```
\begin{tikzcd}
a+b\ar[r]\ar[d]&
c\ar[d]&
d\ar[r]& e-f
\end{tikzcd}
```

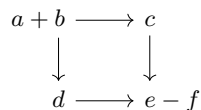
There are ways to change the node alignment and modify the arrow path.

The former may be achieved, for example, taking advantage of the TikZ possibility to set the matrix's column style.

For the latter, the appropriate coordinates can be used in a `from/to` or the path can be designed with `to path`.

Box 10 – Column style

```
\begin{tikzcd}[
  /tikz/column 1/.append
    style={nodes={anchor=base east}},
  /tikz/column 2/.append
    style={nodes={anchor=base west}}
]
|[alias=first]|a+b\ar[r]\ar[d,
  from={first.south -| second}]&
c\ar[d, to path={--
  (\tikztotarget.north -|
  \tikztostart)}]\ \
|[alias=second]|d\ar[r] & e-f
\end{tikzcd}
```



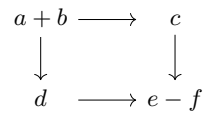
Here, as usual in TikZ, `-|` means “horizontal coordinate of the point before `-` and vertical coordinate of the point after `|`”. For further details see Section 3.2 of the package documentation [3].

If the standard alignment is OK for you, but you would like, for example, to have the same length of the horizontal arrows without calculating it manually, you can set the node width to a given dimension.

In the following, the width of the longest node text is used, taking advantage of the option `text width=width(<string>)`.

Box 11 – Node width

```
\begin{tikzcd}[
  /tikz/column 1/.append style={
    nodes={text width=width("$a+b$"), text
    centered}},
  /tikz/column 2/.append style={
    nodes={text width=width("$e-f$"), text
    centered}}
]
a+b\ar[r]\ar[d]& c\ar[d]\ \
d\ar[r] & e-f
\end{tikzcd}
```



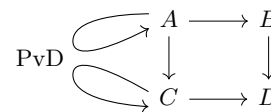
For complex diagrams, it is possible to add code after your diagram is drawn, with `remember picture and overlay` or `execute at end picture`

The difference between these two methods is that what you draw with the former is not included in the bounding box of the picture. To visualize this, compare the alignment of the following two examples.

In Box 12, the entire picture is centered, including what gets drawn at end (the “PvD” and curves).

Box 12 – Execute at end picture

```
\begin{tikzcd}[
  every matrix/.append style={name=mycd,},
  execute at end picture={
    \node[left=of \tikzcdmatrixname]
      (leftofm) {PvD};
    \draw[->] plot [smooth, tension=7]
      coordinates
        {(\tikzcdmatrixname-1-1.160)
        (leftofm.north east)
        (\tikzcdmatrixname-1-1.200)};
    \draw[->] plot [smooth, tension=7]
      coordinates
        {(mycd-2-1.160) (leftofm.south east)
        (mycd-2-1.200)};
  }
]
A \ar[r]\ar[d] & B \ar[d] \ \
C \ar[r] & D
\end{tikzcd}
```



In Box 13, only the actual CD is centered, what is added afterwards is not.

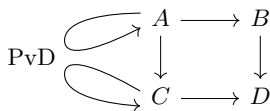
Box 13 – Remember picture and overlay

```
\begin{tikzcd}[
  every matrix/.append style={name=mycd,},
  remember picture
```

```

]
A \ar[r]\ar[d] & B \ar[d] \\
C \ar[r]      & D
\end{tikzcd}%
\begin{tikzpicture}[
  overlay, remember picture
]
\node[left=of \tikzcdmatrixname]
  (leftofm) {PvD};
\draw[->] plot [smooth, tension=7]
  coordinates
  {(\tikzcdmatrixname-1-1.160)
  (leftofm.north east)
  (\tikzcdmatrixname-1-1.200)};
\draw[->] plot [smooth, tension=7]
  coordinates
  {(mycd-2-1.160) (leftofm.south east)
  (mycd-2-1.200)};
\end{tikzpicture}

```



Choose one or the other method depending on your needs.



Last but not least, when you use a `tikzcd` in a `beamer` frame or as argument to a command, you get the error:

Single ampersand used with wrong catcode.

You could also have strange errors or undesired alignments when your vertices or labels are math matrices or text tables or, in general, any environment that uses `&`.

These problems can be easily solved using `ampersand replacement=<macro name>` and then using `<macro name>` instead of `&`.

Box 14 – Ampersand replacement

```

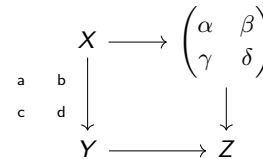
\documentclass{beamer}
\usepackage{tikz-cd}
\tikzcdset{ampersand replacement=\&}
\begin{document}
\begin{frame}
\[\begin{tikzcd}
X\ar[r]\ar[d,"&\text{\begin{tabular}{cc}
& a & b \\ c & & d
\end{tabular}}"]
& \begin{pmatrix}
\alpha & \beta \\
\gamma & \delta
\end{pmatrix}
\end{tikzcd}
\]
\end{frame}
\end{document}

```

```

\end{pmatrix}\ar[d] \\
Y\ar[r] \& Z
\end{tikzcd}
\]
\end{frame}
\end{document}

```



3 Conclusions

If you have been affected by Stendhal syndrome after looking at the commutative diagrams drawn with `tikz-cd`, remember:

*For a quack math,
ask a TikZ duck!*

References

- [1] C. Maggi. The DuckBoat — Beginners' Pond: You do not need to be Neo to cope with a TikZ matrix. *TUGboat* 41(1):20–25, 2020. tug.org/TUGboat/tb41-1/tb127duck-matrix.pdf
- [2] marmotghost. *Bounding boxes for Bézier curves*. mirrors.ctan.org/graphics/pgf/contrib/tikz-bbox/pgfmanual4bbox.pdf. Package page: ctan.org/pkg/tikz-bbox.
- [3] A. Stoffel. `{tikzcd}` — *Commutative diagrams with TikZ. Version 1.0*. mirrors.ctan.org/graphics/pgf/contrib/tikz-cd/tikz-cd-doc.pdf. Package page: ctan.org/pkg/tikz-cd.
- [4] T. Tantau. *The TikZ and PGF packages. Manual for Version 3.1.9a*. mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf. Package page: ctan.org/pkg/pgf.

◇ Herr Professor Paulinho van Duck
Quack University Campus
Sempione Park Pond
Milano, Italy
[paulinho dot vanduck \(at\) gmail dot com](mailto:paulinho.vanduck@gmail.com)