

SOME SAMPLE DIAGRAMS IN L^AT_EX

ANNA MARIE BOHMANN

Here are some sample diagrams in both xymatrix and in tikz. I've only provided very basic written description of how to produce them—the idea is that you should look at the different variations in the source code and mess around with it until you see what works.

XYMatrix. First, xymatrix: To use xymatrix, you create a matrix of entries and then use their relative position in the matrix to specify how the arrows link them up. The arrows can be easily labeled.

The basic xymatrix command is `\xymatrix{}`. It requires the package xy, and I usually pass the “all” option. To center a diagram, include it as a “display math”¹ element by enclosing the whole command in `\[... \]`.

Here's a basic diagram with no arrows:

$$\begin{array}{ccc} X & & Y & & Z \\ & & & & \\ & & U & & V \end{array}$$

The line break is created with `\\` and the column breaks with `&`. Here's the same diagram with a number of added arrows:

$$\begin{array}{ccc} X & \longrightarrow & Y & \longleftarrow & Z \\ \uparrow & & \nearrow & & \uparrow \\ U & \longrightarrow & V & & \end{array}$$

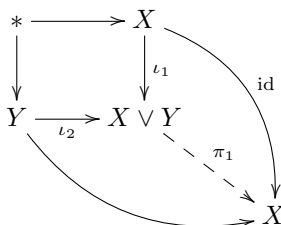
Each arrow is specified separately with a command of the form `\ar[d]` where *d* is replaced by a direction, such as *u* (up), *d* (down), *l* (left), *r* (right), or a combination such as *ul* (up and left) or *ull* (up and two columns left).

To label an arrow, use `\ar[d]^{name}` or `\ar[d]_{name}` depending on whether you want the label to appear on the counterclockwise or clockwise side of the arrow. (Mostly one just experiments to find the best side.)

$$\begin{array}{ccc} X & \xrightarrow{f} & Y & \xleftarrow{g} & Z \\ \uparrow^{i_x} & & \nearrow & & \uparrow^{i_y} \\ U & \xrightarrow{\pi} & V & & \end{array}$$

¹i.e. an unlabeled equation environment

You can do much more complicated stuff too!



Most of the diagrams I create use `xymatrix` and `xypic`.

Troubleshooting: One of the most frequent errors in `xymatrix` diagrams is having blank line before the `}` closing your `\xymatrix` command.

TikZ. TikZ is another way of inputting diagrams (and lots of other types of “pictures”) directly into your latex code. There’s a matrix feature for tikz, too, but instead of using it, the diagrams below feature the “positioning” library, which works quite generally. TikZ code is inserted into your document using the `tikzpicture` environment, which you can enclose in a `center` environment if you want it centered.

Troubleshooting: Note every line of tikz code ends in a semicolon.

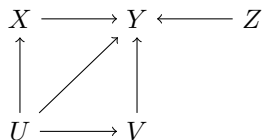
To make a tikz diagram, you create “nodes” and then link them together by drawing arrows. Each node has a label so you can reference it in addition to what gets printed at the node. For example, the command `\node[right=of X] (Y) {Y}`; below says to make a node called “Y” to the right of the node called “X” and print the output of the command `Y` at that spot. Note everything is not in automatically in mathmode, so you have to put each entry in mathmode separately—that’s why the command has the dollar signs in it.

Here’s the basic diagram with no arrows again:

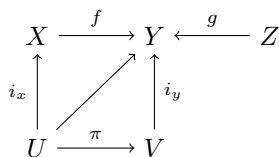
$$X \qquad Y \qquad Z$$

$$U \qquad V$$

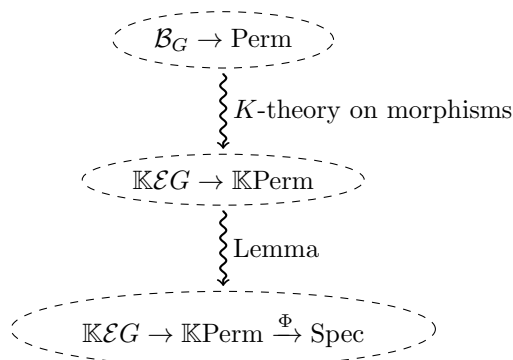
Now we add arrows: Each is given by a `\draw` command indicating what nodes to draw between.



And now we add the labels on the arrows. Here each label is itself a node! I’ve created a special node type called “label” that automatically typesets itself in the middle of the path I’m drawing and uses a smaller font size.



Using tikz like this gives you a lot of flexibility. Here is a crazy diagram from a talk I gave. I've added lots of fancy styles to the nodes and arrows, but the basic syntax is as above.



This diagram would be hard to do in xymatrix.

TikZ-cd. You can also use the package tikz-cd (“tikz commutative diagram”), which is a package built on tikz that provides an xymatrix-like set of commands. The basic command is provided by the *tikzcd* environment, which is usually placed in a $\backslash[$ and $\backslash]$ environment to center is and put everything in mathmode.

Here’s our same diagrams again:

$$\begin{array}{ccc}
 X & Y & Z \\
 & & \\
 U & V &
 \end{array}$$

The “arrow” command here is $\backslash\text{arrow}\{\}$.

$$\begin{array}{ccc}
 X & \longrightarrow & Y & \longleftarrow & Z \\
 \uparrow & & \nearrow & & \uparrow \\
 U & \longrightarrow & V & &
 \end{array}$$

Notice that the arrow directions are placed in curly braces { and } instead of the square brackets [and] used in xymatrix.

Labels automatically appear on the left/top of arrows, but adding the “swap” command moves them to the other side:

$$\begin{array}{ccc}
 X & \xrightarrow{f} & Y & \xleftarrow{g} & Z \\
 \uparrow i_x & & \nearrow & & \uparrow i_y \\
 U & \xrightarrow{\pi} & V & &
 \end{array}$$

And here's another version of diagram at the end of the xymatrix section:

$$\begin{array}{ccc}
 * & \longrightarrow & X \\
 \downarrow & & \downarrow \iota_1 \\
 Y & \xleftarrow{\iota_2} & X \vee Y \\
 & \searrow & \downarrow \text{dashed} \\
 & & X
 \end{array}
 \begin{array}{l}
 \text{id} \\
 \curvearrowright \\
 \end{array}$$

This time it has some gratuitous arrow tails and tips in addition to the bendy arrows. These might be nicer in with tikzcd than the default ones in xymatrix.

Troubleshooting: A blank line before your `\end{tikzcd}` command will create an error, similar to the xymatrix case.

REFERENCES

- [1] Kristoffer H. Rose, “XY-pic User’s Guide,”
<http://tug.org/applications/Xy-pic/soft/xy/doc/xyguide.pdf>
- [2] Till Tantau, “TikZ and PGF: Manual for version 1.18”
<https://www.bu.edu/math/files/2013/08/tikzpgfmanual.pdf>
 Note: newer versions are available and even longer
- [3] “tikzcd: Commutative diagrams with TikZ”
<http://ctan.math.washington.edu/tex-archive/graphics/pgf/contrib/tikz-cd/tikz-cd-doc.pdf>