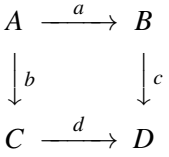
## The `tikz` package

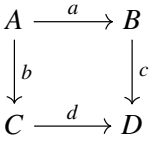
This is a general purpose graphics package. To load it for this document, I used:

```
\usepackage{tikz}
\usetikzlibrary{matrix,arrows,decorations.pathmorphing}
```

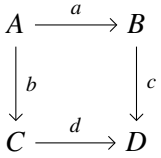
There are now three ways to enter commutative diagrams using `tikz`: with the package tikz-cd, with `matrix`, and directly with `tikz` (listed roughly in order of decreasing ease but increasing flexibility).



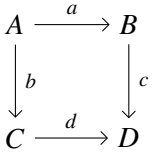
```
$\begin{CD}
A@>a>>B\\
@VVbV@VVcV\\
C@>d>>D
\end{CD}$
(amscd)
```



```
\begin{tikzcd}
A \arrow{r}{a} \arrow{d}{b}
              &B \arrow{d}{c}\\
C \arrow{r}{d} &D
\end{tikzcd}
(tikz-cd)
```



```
\begin{tikzpicture}
\matrix(m)[matrix of math nodes,
row sep=2.6em, column sep=2.8em,
text height=1.5ex, text depth=0.25ex]
{A&B\\
C&D\\};
\path[->,font=\scriptsize,>=angle 90]
(m-1-1) edge node[auto] {$a$} (m-1-2)
        edge node[auto] {$b$} (m-2-1)
(m-1-2) edge node[auto] {$c$} (m-2-2)
(m-2-1) edge node[auto] {$d$} (m-2-2);
\end{tikzpicture}
(matrix)
```



```
\begin{tikzpicture}[scale=1.5]
\node (A) at (0,1) {$A$};
\node (B) at (1,1) {$B$};
\node (C) at (0,0) {$C$};
\node (D) at (1,0) {$D$};
\path[->,font=\scriptsize,>=angle 90]
(A) edge node[above]{$a$} (B)
(A) edge node[right]{$b$} (C)
(B) edge node[right]{$c$} (D)
(C) edge node[above]{$d$} (D);
\end{tikzpicture}
(tikz)
```

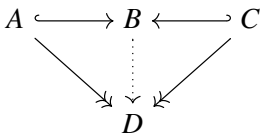
This is part of: Guide to Commutative Diagrams, `www.jmilne.org/not/CDGuide.html`
Last revised October 30, 2012

## Using `tikz-cd`

Load[1] this with `\usepackage{tikz-cd}`. As the code on p.1 illustrates, the syntax for `tikz-cd` is similar to that of array. Note that `tikz-cd` handles large objects and tall labels better than `amscd`:

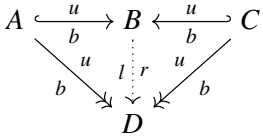
$$\begin{array}{ccc} A\times A\times A\times A\times A\times A & \xrightarrow{a} & B \\ \downarrow{\scriptstyle b} & & \downarrow{\scriptstyle c} \\ C & \xrightarrow{d} & D \end{array} \qquad \begin{array}{ccc} A & \xrightarrow{a} & B \\ \downarrow{\scriptstyle b} & & \downarrow{\scriptstyle c} \\ C & \xrightarrow{\overline{A^{A^A}}} & D \end{array}$$

The next example illustrates the use of different arrows in a commutative diagram:



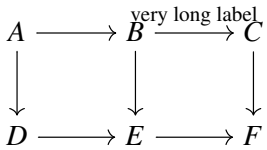
```
\begin{tikzcd}
A \arrow[hook]{r}\arrow[two heads]{rd}
&B \arrow[dotted]{d}\arrow[hookleftarrow]{r}
&C \arrow[two heads]{ld}\\
&D
\end{tikzcd}
```

Now an example with labels on the arrows:



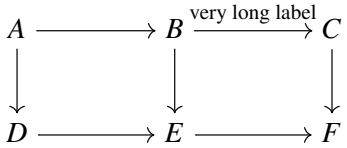
```
\begin{tikzcd}
A \arrow[hook]{r}{u}[swap]{b}
  \arrow[two heads]{rd}{u}[swap]{b}
&B \arrow[dotted]{d}{r}[swap]{l}
   \arrow[hookleftarrow]{r}{u}[swap]{b}
&C \arrow[two heads]{ld}{b}[swap]{u}\\
&D
\end{tikzcd}
```

Long labels may cause problems:



```
\begin{tikzcd}
A\arrow{r}\arrow{d}
&B\arrow{r}{\text{very long label}}\arrow{d}
&C\arrow{d}\\
D\arrow{r}&E\arrow{r}&F
\end{tikzcd}
```

However, this can be fixed as follows:



```
\begin{tikzcd}[column sep=large]
A\arrow{r}\arrow{d}
...
```

`tikz-cd` does not have a problem with objects of different heights.

$$\hat{A} \longrightarrow \prod_{n\in Z} A_n \longrightarrow \prod_{n\in Z} A_n.$$

[1] Before using `tikz-cd`, check that your TEX installation is using version 2.10 of `pgf` — you can do this by running TEX on a file containing `\pgfversion`.

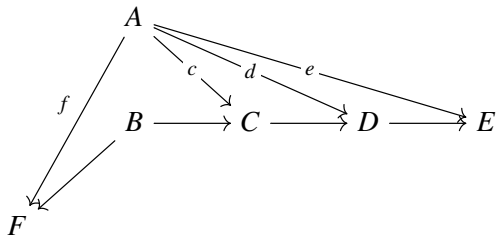
Curving arrows is easy.

```
\begin{tikzcd}
A\arrow[bend left]{r}\arrow[bend right]{r}&B
\end{tikzcd}
```

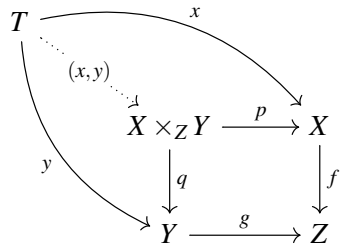
When a diagram is numbered, this is placed correctly:

$$\begin{array}{ccc} A & \xrightarrow{a} & B \\ \downarrow{\scriptstyle b} & & \downarrow{\scriptstyle c} \\ C & \xrightarrow{d} & D \end{array} \tag{1}$$

Two more examples:



```
\begin{tikzcd}
&A\arrow{ldd}[swap]{f}\arrow{rd}[description]{c}
  \arrow{rrd}[description]{d}
  \arrow{rrrd}[description]{e}\\
&B\arrow{ld}\arrow{r}&C\arrow{r}&D\arrow{r}&E\\
F
\end{tikzcd}
```

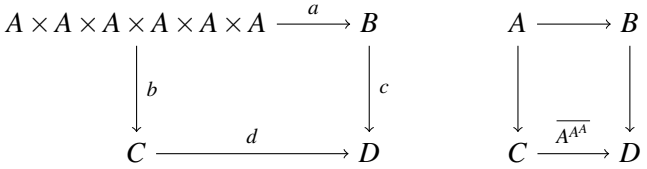


```
\begin{tikzcd}
T\arrow[bend left]{drr}{x}
\arrow[bend right]{ddr}[swap]{y}
\arrow[dotted]{dr}[description]{(x,y)} & & \\
& X \times_Z Y \arrow{r}{p} \arrow{d}{q} & X \arrow{d}{f} \\
& Y \arrow{r}{g} & Z
\end{tikzcd}
```

## Using `matrix`

The code on p.1 sets up a matrix named `m` with some options, and then places $A$, $B$, $C$, and $D$ at the four positions of a $2 \times 2$ matrix. The next line specifies normal arrows with labels in scriptsize and a nondefault arrow head, and the following line specifies an arrow from the (1,1) position of the matrix `m` to the (1,2) position with a label $a$ in the default position.

Note that `tikz` handles large objects and tall labels better than amscd:




To my eyes, the arrow heads are too small.[2] This can be fixed by adding `>=angle 90`, as an option to the path or to the whole picture:


`\path[->](1,1) edge (2,1);`


`\path[->,>=angle 90](1,1) edge (2,1);`

Here is the code for some arrows.


`\path[->](1,1) edge (2,1);`


`\path[|->](1,1) edge (2,1);`


`\path[-](1,1) edge (2,1);`


`\path[right hook->](1,1) edge (2,1);`


`\path[->>](1,1) edge (2,1);`


`\path[dotted,->](1,1) edge (2,1);`

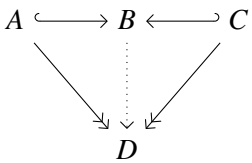

`\path[dashed,->](1,1) edge (2,1);`


`\path[*->](1,1) edge (2,1);`


`\draw[double distance = 1.5pt](1,1) -- (2,1);`


`http://tex.stackexchange.com/questions/12678/`

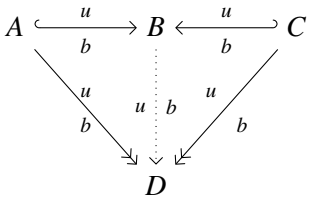
The next example illustrates the use of the different arrows in a commutative diagram




```
\begin{tikzpicture}[>=angle 90]
\matrix(a)[matrix of math nodes,
row sep=3em, column sep=2.5em,
text height=1.5ex, text depth=0.25ex]
{A&B&C\\
&D\\};
\path[right hook->](a-1-1) edge (a-1-2);
\path[->>](a-1-1) edge (a-2-2);
\path[dotted,->](a-1-2) edge (a-2-2);
\path[left hook->](a-1-3) edge (a-1-2);
\path[->>](a-1-3) edge (a-2-2);
\end{tikzpicture}
```

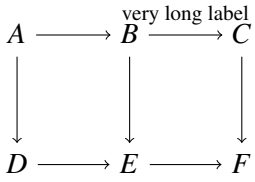
[2] See `http://tex.stackexchange.com/questions/37320/` for an erudite discussion of this problem, with solutions.

Now an example with labels on the arrows:



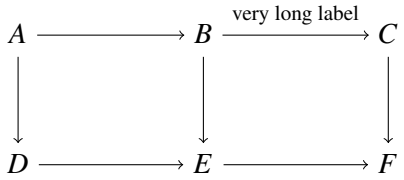

```
\path[right hook->,font=\scriptsize]
(a-1-1) edge node[above]{$u$}
              node[below]{$b$} (a-1-2);
\path[->>,font=\scriptsize]
(a-1-1) edge node[above]{$u$}
              node[below]{$b$} (a-2-2)
(a-1-3) edge node[above left]{$u$}
              node[below right]{$b$} (a-2-2);
\path[dotted,->,font=\scriptsize]
(a-1-2) edge node[left]{$u$}
              node[right]{$b$} (a-2-2);
\path[left hook->,font=\scriptsize]
(a-1-3) edge node[above]{$u$}
              node[below]{$b$} (a-1-2);
```

Long labels may cause a problem:




```
\begin{tikzpicture}
\matrix(m)[matrix of math nodes,
row sep=3em, column sep=2.5em,
text height=1.5ex, text depth=0.25ex]
{A&B&C\\
D&E&F\\};
\path[->,font=\scriptsize]
(m-1-1) edge (m-1-2)
        edge (m-2-1)
(m-1-2) edge node[auto] {very long label} (m-1-3)
        edge (m-2-2)
(m-1-3) edge (m-2-3)
(m-2-1) edge (m-2-2)
(m-2-2) edge (m-2-3);
\end{tikzpicture}
```

However, this can be fixed by setting `column sep=5.0em`.




`tikz` does not have a problem with objects of different heights.

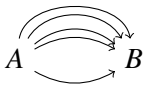
$$\hat{A} \longrightarrow \prod_{n\in Z} A_n \longrightarrow \prod_{n\in Z} A_n.$$

But that is because of the options `text height=1.5ex, text depth=0.25ex`. When you omit

them, you get:

$$\hat{A} \longrightarrow \prod_{n\in Z} A_n \longrightarrow \prod_{n\in Z} A_n.$$

Curving arrows is easy.




```
\begin{tikzpicture}
\matrix(m)[matrix of math nodes,
row sep=3em, column sep=2.8em,
text height=1.5ex, text depth=0.25ex]
{A&B\\};
\path[->]
(m-1-1) edge [bend left] (m-1-2)
        edge [bend left=40] (m-1-2)
        edge [bend left=60] (m-1-2)
        edge [bend left=80] (m-1-2)
        edge [bend right] (m-1-2);
\end{tikzpicture}
```

Arrows may not attach themselves correctly to the nodes:

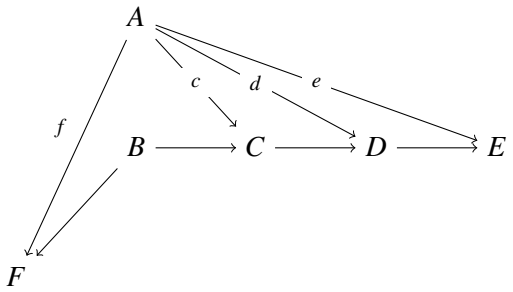
$$0 \longrightarrow \mathrm{Br}(L/K) \longrightarrow \bigoplus\nolimits_v \mathrm{Br}(L^v/K_v) \begin{array}{l} \nearrow H^2(L/K) \\ \searrow \mathbb{Q}/\mathbb{Z} \end{array}$$

To fix this, use

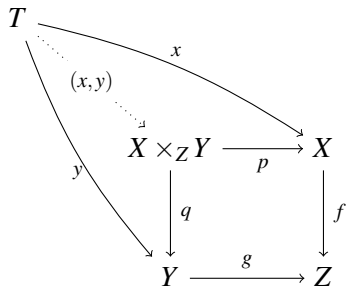
```
(m-2-3.north east) edge (m-1-4)
(m-2-3.south east) edge (m-3-4);
```

$$0 \longrightarrow \mathrm{Br}(L/K) \longrightarrow \bigoplus\nolimits_v \mathrm{Br}(L^v/K_v) \begin{array}{l} \nearrow H^2(L/K) \\ \searrow \mathbb{Q}/\mathbb{Z} \end{array}$$

Two more examples:




```
\[
\begin{tikzpicture}[descr/.style={fill=white}]
\matrix(m)[matrix of math nodes, row sep=3em, column sep=2.8em,
text height=1.5ex, text depth=0.25ex]
{&A\\&B&C&D&E\\F\\};
\path[->,font=\scriptsize]
(m-1-2) edge node[above left] {$f$} (m-3-1)
        edge node[descr] {$c$} (m-2-3)
        edge node[descr] {$d$} (m-2-4)
        edge node[descr] {$e$} (m-2-5);
\path[->]
(m-2-2) edge (m-3-1)
        edge (m-2-3);
\path[->]
(m-2-3) edge (m-2-4);
\path[->]
(m-2-4) edge (m-2-5);
\end{tikzpicture}
\]
```

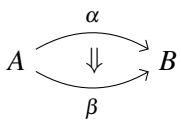



```
\[
\begin{tikzpicture}[descr/.style={fill=white}]
\matrix(m)[matrix of math nodes, row sep=3em, column sep=2.8em,
text height=1.5ex, text depth=0.25ex]
{T\\&X\times_Z Y&X\\&Y&Z\\};
```

```
\path[->,font=\scriptsize]
(m-1-1) edge [bend left=10] node[above] {$x$} (m-2-3)
(m-1-1) edge [bend right=10] node[below] {$y$} (m-3-2);
\path[->,dotted,font=\scriptsize]
(m-1-1) edge node[descr] {$(x,y)$} (m-2-2);
\path[->,font=\scriptsize]
(m-2-2) edge node[below] {$p$} (m-2-3)
(m-2-2) edge node[right] {$q$} (m-3-2);
\path[->,font=\scriptsize]
(m-2-3) edge node[right] {$f$} (m-3-3);
\path[->,font=\scriptsize]
(m-3-2) edge node[above] {$g$} (m-3-3);
\end{tikzpicture}
\]
```

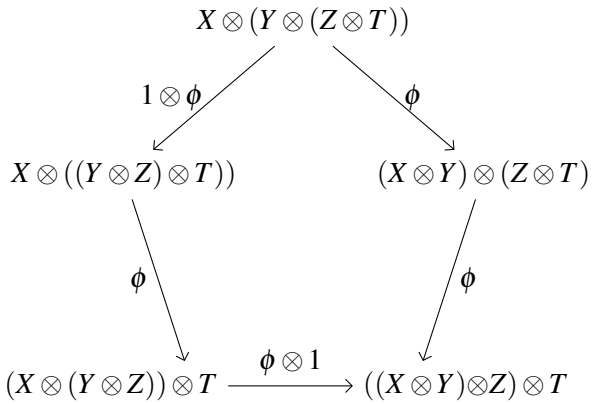
### Using `tikz` directly

Instead of using a matrix grid, you can use `tikzpicture` directly to construct a diagram.




```
\begin{tikzpicture}
\node (A) at (-1,0) {$A$};
\node (B) at (1,0) {$B$};
\node at (0,0) {\rotatebox{270}{$\Rightarrow$}};
\path[->,font=\scriptsize,>=angle 90]
 (A) edge [bend left] node[above] {$\alpha$} (B)
     edge [bend right] node[below] {$\beta$} (B);
\end{tikzpicture}
```

(`rotatebox` requires `graphicx`.)




Pentagon Axiom

```
\begin{tikzpicture}
\node (P0) at (90:2.8cm) {$X\otimes (Y\otimes (Z\otimes T))$};
\node (P1) at (90+72:2.5cm) {$X\otimes ((Y\otimes Z)\otimes T))$} ;
\node (P2) at (90+2*72:2.5cm) {$\mathllap{(X\otimes (Y\otimes Z))}\otimes T$};
\node (P3) at (90+3*72:2.5cm) {$((X\otimes Y)\mathrlap{\otimes Z)\otimes T}$};
\node (P4) at (90+4*72:2.5cm) {$(X\otimes Y)\otimes (Z\otimes T)$};
\draw
(P0) edge[->,>=angle 90] node[left] {$1\otimes\phi$} (P1)
(P1) edge[->,>=angle 90] node[left] {$\phi$} (P2)
(P2) edge[->,>=angle 90] node[above] {$\phi\otimes 1$} (P3)
(P4) edge[->,>=angle 90] node[right] {$\phi$} (P3)
(P0) edge[->,>=angle 90] node[right] {$\phi$} (P4);
\end{tikzpicture}
```

Here I used `\mathllap` and `\mathrlap` to adjust the positions of the nodes. They require the package `mathtools`.

When you number a displayed commutative diagram

```
\begin{equation}
\begin{tikzpicture}
......
```

```
\end{tikzpicture}
\end{equation}
```

$$\begin{array}{ccccc} A & \xrightarrow{a} & B & \xrightarrow{b} & C \\ \downarrow{\scriptstyle c} & & \downarrow{\scriptstyle d} & & \downarrow{\scriptstyle e} \\ D & \xrightarrow{f} & E & \xrightarrow{g} & F \end{array} \tag{2}$$

the number appears below the level of the diagram. To centre the number, use:

```
\begin{equation}
\begin{tikzpicture}[baseline=(current bounding box.center)]
......
\end{tikzpicture}
\end{equation}
```

$$\begin{array}{ccccc} A & \xrightarrow{a} & B & \xrightarrow{b} & C \\ \downarrow{\scriptstyle c} & & \downarrow{\scriptstyle d} & & \downarrow{\scriptstyle e} \\ D & \xrightarrow{f} & E & \xrightarrow{g} & F \end{array} \tag{3}$$

Here are two examples with multiple arrows:

$$Y \times_X Y \underset{p_2}{\overset{p_1}{\rightrightarrows}} Y \longrightarrow X$$

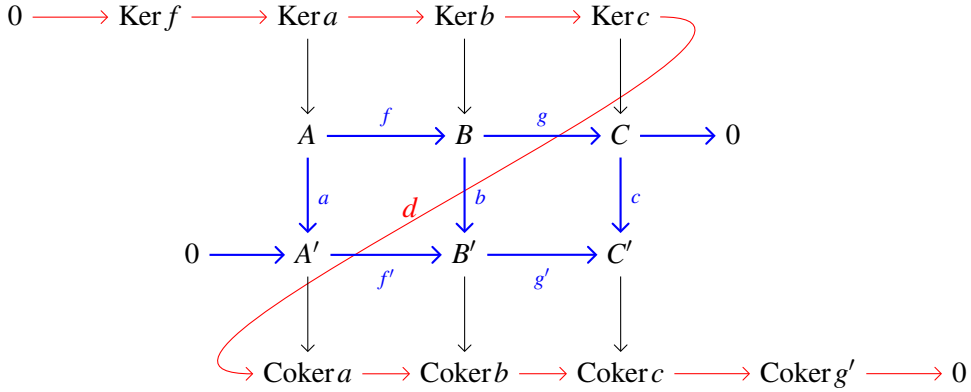
```
\begin{tikzpicture}
\node (a) at (0,0) {$Y\times_X Y$};
\node (b) at (2,0) {$Y$};
\node (c) at (3.5,0) {$X$};
\path[->,font=\scriptsize,>=angle 90]
([yshift= 2pt]a.east) edge node[above] {$p_1$} ([yshift= 2pt]b.west)
([yshift= -2pt]a.east) edge node[below] {$p_2$} ([yshift= -2pt]b.west)
(b) edge (c);
\end{tikzpicture}
```

$$\mathsf{S}(Z) \underset{i^!}{\overset{i^*}{\leftleftarrows}} \mathsf{S}(X) \underset{j_*}{\overset{j_!}{\leftleftarrows}} \mathsf{S}(U).$$

```
\begin{tikzpicture}[descr/.style={fill=white},text height=1.5ex, text depth=0.25ex]
\node (a) at (0,0) {$\mathsf{S}(Z)$};
\node (b) at (2.5,0) {$\mathsf{S}(X)$};
\node (c) at (5,0) {$\mathsf{S}(U).$};
\path[->,font=\scriptsize,>=angle 90]
([yshift= 9pt]b.west) edge node[above] {$i^{\ast}$} ([yshift= 9pt]a.east)
```

```
(a.east) edge node[descr] {$i_{\ast}$} (b.west)
([yshift= -9pt]b.west) edge node[below] {$i^!$} ([yshift= -9pt]a.east)
([yshift= 9pt]c.west) edge node[above] {$j_!$} ([yshift= 9pt]b.east)
(b.east) edge node[descr] {$j^{\ast}$} (c.west)
([yshift= -9pt]c.west) edge node[below] {$j_*$} ([yshift= -9pt]b.east);
\end{tikzpicture}
```

One final example: the extended snake lemma says that the exact commutative diagram in blue gives rise to the exact sequence in red.




```
\begin{tikzpicture}[>=angle 90,scale=2.2,text height=1.5ex, text depth=0.25ex]
%%First place the nodes
\node (k-1) at (0,3) {$0$};
\node (k0) [right=of k-1] {$\Ker f$};
\node (k1) [right=of k0] {$\Ker a$};
\node (k2) [right=of k1] {$\Ker b$};
\node (k3) [right=of k2] {$\Ker c$};
\node (a1) [below=of k1] {$A$};
\node (a2) [below=of k2] {$B$};
\node (a3) [below=of k3] {$C$};
\node (a4) [right=of a3] {$0$};
\node (b1) [below=of a1] {$A'$};
\node (b0) [left=of b1] {$0$};
\node (b2) [below=of a2] {$B'$};
\node (b3) [below=of a3] {$C'$};
\node (c1) [below=of b1] {$\Coker a$};
\node (c2) [below=of b2] {$\Coker b$};
\node (c3) [below=of b3] {$\Coker c$};
\node (c4) [right=of c3] {$\Coker g'$};
\node (c5) [right=of c4] {$0$};
%%Draw the red arrows
\draw[->,red,font=\scriptsize]
(k-1) edge (k0)
(k0)  edge (k1)
(k1)  edge (k2)
(k2)  edge (k3)
```

```
(c1)  edge (c2)
(c2)  edge (c3)
(c3)  edge (c4)
(c4)  edge (c5);
%%Draw the curvy red arrow
\draw[->,red]
(k3) edge[out=0,in=180,red] node[pos=0.55,yshift=5pt] {$d$} (c1);
%%Draw the black arrows
\draw[->]
(k1) edge (a1)
(k2) edge (a2)
(k3) edge (a3)
(b1) edge (c1)
(b2) edge (c2)
(b3) edge (c3);
%%Draw the thick blue arrows
\draw[->,font=\scriptsize,blue,thick]
(a1) edge node[auto] {$f$} (a2)
(a2) edge node[auto] {$g$} (a3)
(a3) edge (a4)
(a1) edge node[auto] {$a$} (b1)
(a2) edge node[auto] {$b$} (b2)
(a3) edge node[auto] {$c$} (b3)
(b0) edge (b1)
(b1) edge node[below] {$f'$} (b2)
(b2) edge node[below] {$g'$} (b3);
\end{tikzpicture}
```

For the last diagram, I added the following lines to the preamble

```
\usepackage{amsmath}
\DeclareMathOperator{\Coker}{Coker}
\DeclareMathOperator{\Ker}{Ker}
\usetikzlibrary{positioning}
```