The `diagxy` package

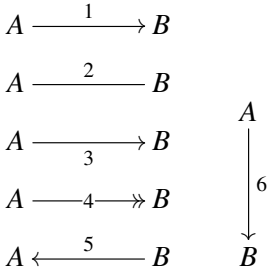
This is a front end to `Xy-pic` that contains templates for diagrams. [1] To load it for this document, I used

```
\usepackage[all,cmtip]{xy}
\usepackage{diagxy}
```

The simplest template is

`\morphism(x,y)|p|/{sh}/<dx,dy>[N‘N;L]`

where `(x,y)` gives the position of the start of the arrow in units of .01em, `|p|` gives the position of the label (**a**bove, **b**elow, **l**eft, **r**ight, or **m**id), `{sh}` gives the shape of the arrow (the part in parentheses in the table on p. 1 of the `xymatrix` guide), `<dx,dy>` gives the coordinates of the end of the arrow relative to the start, `N` is an object, and `L` is a label, as in:

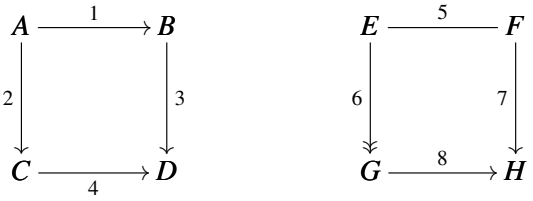



```
\[\bfig
\morphism(0,800)[A‘B;1]
\morphism(0,600)/{-}/[A‘B;2]
\morphism(0,400)|b|[A‘B;3]
\morphism(0,200)|m|/{->>}/[A‘B;4]
\morphism/{<-}/[A‘B;5]
\morphism(800,500)|r|<0,-500>[A‘B;6]
\efig\]
```

It is not possible to have labels both above and below an arrow.

For more complicated templates, such as that for a square, the syntax is similar:

```
\[\bfig
\square(0,0)[A‘B‘C‘D;1‘2‘3‘4]
\square(1200,0)|aaaa|/{-}‘{>>}‘>‘>/[E‘F‘G‘H;5‘6‘7‘8]
\efig\]
```

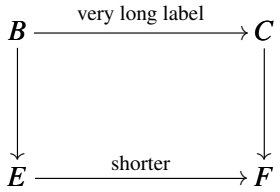


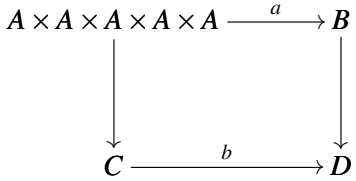
`square` doesn't adjust for long labels or large objects but its variant `Square` does:

```
\[\bfig
\Square|aaaa|[B‘C‘E‘F;\text{very long label}‘{}‘{}‘{\text{shorter}}]
\Square(2000,0)|aaaa|[A\times A\times A\times A‘B‘C‘D;a‘{}‘{}‘b]
\efig\]
```

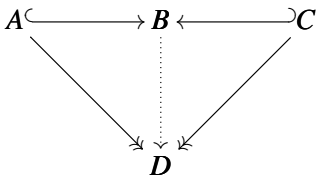
This is part of: Guide to Commutative Diagrams, `www.jmilne.org/not/CDGuide.html`
September 23, 2010

[1] If your TEX system doesn't have it, you can get `diagxy.tex` from the author's home page `ftp://ftp.math.mcgill.ca/pub/barr/`, rename it to diagxy.sty and place it somewhere your TEX system can find it. There is a comparison of `diagxy` with `xymatrix` at `http://www.emis.de/journals/TAC/style/diagxy-xymatrix.pdf`.





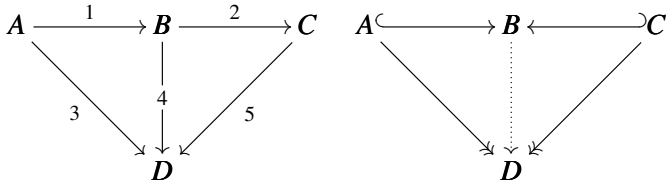
It is possible to combine templates to get more complicated diagrams, as in:




```
\morphism(0,500)/{^{(}->}/[A‘B;]
\morphism(0,500)/{->>}/<500,-500>[A‘D;]
\morphism(500,500)/{.>}/<0,-500>[B‘D;]
\morphism(1000,500)/{_{(}->}/<-500,0>[C‘B;]
\morphism(1000,500)/{->>}/<-500,-500>[C‘D;]
\efig\]
```

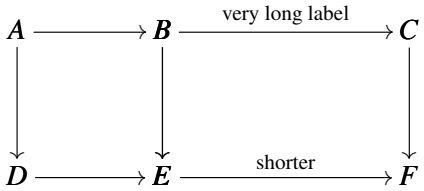
Fortunately, there is a template `Vtrianglepair` that makes this much easier:

```
\[\bfig
\Vtrianglepair[A‘B‘C‘D;1‘2‘3‘4‘5]
\Vtrianglepair(1200,0)/{^{(}->}‘{<-^{)}}‘{->>}‘{.>}‘{->>}/[A‘B‘C‘D;{}‘{}‘{}‘{}‘{}]
\efig\]
```



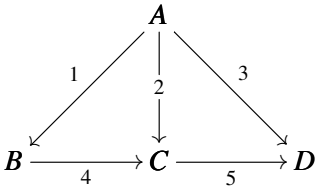
By combining two `Square`s, one can build more complicated diagrams:

```
\[\bfig
\Square[A‘B‘D‘E;{}‘{}‘{}‘{}]
\Square(500,0)|aaaa|[B‘C‘E‘F;\text{very long label}‘{}‘{}‘{\text{shorter}}]
\efig\]
```

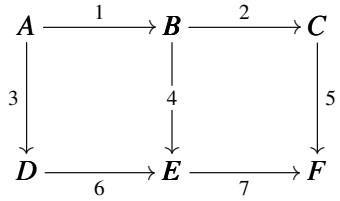


Here are some other templates.

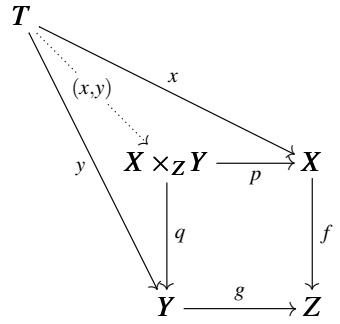
```
\[\bfig
\Atrianglepair[A‘B‘C‘D;1‘2‘3‘4‘5]
\efig
\]
```



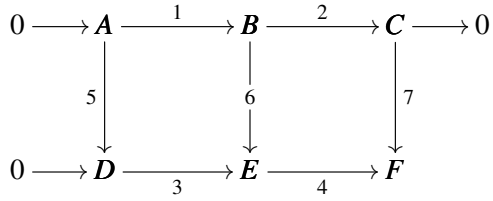
```
\[\bfig
\hSquares[A‘B‘C‘D‘E‘F;1‘2‘3‘4‘5‘6‘7]
\efig
\]
```




```
\[\bfig
\pullback|brra|[X\times_ZY‘X‘Y‘Z;p‘q‘f‘g]%
/>‘{.>}‘>/[T;x‘(x,y)‘y]
\efig
\]
```



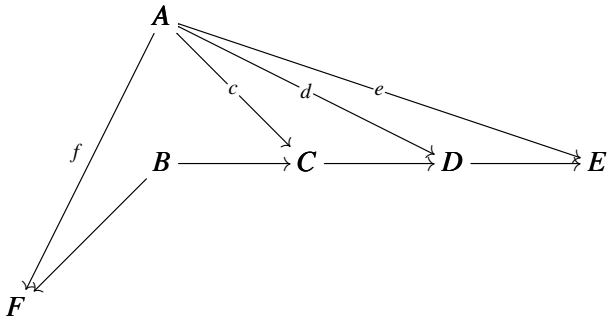

```
\[\bfig
\iiixii {7}<300>[A‘B‘C‘D‘E‘F;1‘2‘3‘4‘5‘6‘7]
\efig
```



Which 0s appear is determined by the first number in braces, which must be between 0 and 15 (it is 7 in the above example), and depends on the binary expansion of the number, as illustrated by the examples at right:

	1	2	4	8	
5	1		1		0 0
7	1	1	1		0 0 0
14		1	1	1	0 0 0

The diagram

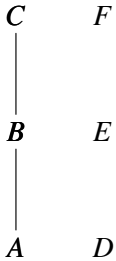


doesn't fit any template, but `\diagxy` offers an alternative method of building diagrams:

```
\[\bfig
\node a(500,1000)[A]
\node b(500,500)[B]
\node c(1000,500)[C]
\node d(1500,500)[D]
\node e(2000,500)[E]
\node f(0,0)[F]
\arrow[a‘f;f]
\arrow|m|[a‘c;c]
\arrow|m|[a‘d;d]
\arrow|m|[a‘e;e]
\arrow[b‘f;{}]
\arrow[b‘c;{}]
\arrow[c‘d;{}]
\arrow[d‘e;{}]
\efig\]
```

The line `\node a(500,1000)[A]` places the object A at $(500, 1000)$ and labels it with a (for internal purposes). The line `\arrow[a‘f;f]` runs an arrow from the node "a" to the node "f" and labels it with f.

If there is no arrow between nodes, then the nodes don't print, but you can add empty arrows:




```
\[\bfig
\node a(0,0)[A]
\node b(0,400)[B]
\node c(0,800)[C]
\node d(300,0)[D]
\node e(300,400)[E]
\node f(300,800)[F]
\arrow/{-}/[a‘b;{}]
\arrow/{-}/[b‘c;{}]
\arrow/{}/[a‘d;{}]
\arrow/{}/[b‘e;{}]
\arrow/{}/[c‘f;{}]
\efig\]
```

Personally, I find this to be the most convenient way to enter complicated diagrams.