User's guide for the *Diagram* Macros

Francis Borceux

UCL, Louvain-la-Neuve,Belgium. *

Contents

*bitnet:FBORCEUX@BUCLLN11

1 What are the *Diagram* Macros?

Diagram is a series of Macros intended to produce easily rather sophisticated diagrams which can appear when typing category theory. The range of arrows which can be produced includes plain morphisms, dotted morphisms, monomorphisms, epimorphisms, bimorphisms, isomorphisms, equalities, pairs, triples, quadruples and quintuples of parallel or adjoint arrows. Most arrows can be drawned in the sixteen basic directions of the compass-card (and this inspires the terminology). Some curved arrows are also available. All these arrows can be given a name, on either side of the arrow. For emphasizing, arrows can be printed in bold-face type. Except for some special features, the Macros will take care of choosing the correct length of each arrow as well as positioning correctly the various elements which appear in the diagram. The final result will be a diagram with rows and columns at a distance of 80 points (1.1in, 28mm) from each other; this standard distance can be changed for every individual diagram just by giving a scaling factor.

Diagram includes also a list of in-text symbols containing in particular the same variety of arrows in both the forward and the backward directions.

The *Diagram* Macros are LaTeX Macros which make an intensive use of the LaTeX picture environment and in particular of the `\vector` command. To use them for producing a document, it suffices to have in your TeX file a copy of the file *diagram* and to call it with the command `\input{diagram}`, in the preamble of your new document. The *Diagram* macros will give good results with 10pt, 11pt and 12pt styles.The following sections should provide you with full information on how to use the *Diagram* macros for producing your documents.

2 Inserting in-text arrows.

An in-text arrow is one which you include in a line of text, like for example when you write:

Let $f : A \mathrel{+\!\!\!\rightarrow} B$ be a monomorphism ...

Here are the available in-text arrows and the corresponding commands. The length of an in-text arrow is 20pt.

plain arrow`\ar` $\longrightarrow$
dotted arrow`\dotar` $\cdots\!\!\rightarrow$
monomorphism`\mono` $\mathrel{+\!\!\!\rightarrow}$
epimorphism`\epi` $\twoheadrightarrow$
bimorphism`\bimo` $\mathrel{+\!\!\!\twoheadrightarrow}$
isomorphism`\iso` $\xrightarrow{\cong}$
pair of parallel arrows`\biar` $\rightrightarrows$
equality`\eql` $=\!=$
pair of adjoint arrows`\adjar` $\rightleftarrows$

triple of parallel arrows`\triar` $\longrightarrow\!\!\!\!\!\rightarrow$

triple of adjoint arrows`\triadjar` $\leftarrow\!\!\!\!\!\rightarrow$

quadruple of parallel arrows`\quadriar` $\rightarrow\!\!\!\!\!\!\rightarrow$

quadruple of adjoint arrows`\quadriadjar` $\leftarrow\!\!\!\!\!\rightarrow$

quintuple of parallel arrows`\quintiar` $\rightarrow\!\!\!\!\!\!\!\rightarrow$

quintuple of adjoint arrows`\quintiadjar` $\leftarrow\!\!\!\!\!\!\rightarrow$

When you insert an in-text arrow, the *Diagram* Macros will take care of the spacing at both ends of the arrow. The in-text arrow commands can be used both in paragraph and in Math mode, but normally you should use them in Math mode to print the surrounding characters in Math italic. Compare indeed

`Let f:A\epi B be an epimorphism` Let f:A$\twoheadrightarrow$B be an epimorphism with

`Let $f:A\epi B$ be an epimorphism` Let $f : A \twoheadrightarrow B$ be an epimorphism

3 Backward in-text arrows.

It is a general rule of the *Diagram* Macros that to draw an arrow in some given direction, it suffices to preceed the name of the arrow by a code for the direction. To draw an in-text arrow in the backward direction, it suffices to add the prefix `bk` to the command for that arrow.

backward plain arrow`\bkar`

backward dotted arrow`\bkdotar`

backward monomorphism`\bkmono`

backward epimorphism`\bkepi`

backward bimorphism`\bkbimo`

backward isomorphism`\bkiso`

backward pair of parallel arrows`\bkbiar`

backward equality`\bkeql`

backward pair of adjoint arrows`\bkadjar`

backward triple of parallel arrows`\bktriar`

backward triple of adjoint arrows`\bktriadjar`

backward quadruple of parallel arrows`\bkquadriar`

backward quadruple of adjoint arrows`\bkquadriadjar`

backward quintuple of parallel arrows`\bkquintiar`

backward quintuple of adjoint arrows`\bkquintiadjar`

4 Naming the in-text arrows.

In a line of text, it is probably a better idea to type $f : A \mapsto B$ than $A \overset{f}{\mapsto} B$. Nevertheless the *Diagram* Macros allow you to give an upper name to the single and triple in-text arrows as well as an upper-lower name to the double in-text arrows. It is a general rule of the *Diagram* Macros that to give an upper name to a single or a triple arrow or an upper-lower name to a double arrow, it suffices to

1. type the first letter of the corresponding command as an upper-case letter
2. give the name(s) of the arrow(s) as argument(s) of the command.

On next page is a list of the available named in-text arrows.

The name of an arrow will always be processed in Math mode at the script-style size; therefore the text characters will be printed in Math italic and you should not type any $ sign to introduce a mathematical symbol in the name of an arrow.

5 Emphasizing an arrow.

The dotted arrows are a first type of emphasized arrows; their construction is rather slow. It will be easier for LaTeX to print an arrow in bold face style: to realize this, just preceed the command for the arrow by the command `\B`; thus `$A\B\Mono{f}B$` produces $A \overset{f}{\mapsto} B$. That "bolding" process applies to both the "in text" and the "diagram" arrows.

plain arrow f`\Ar{f}` $\xrightarrow{f}$

dotted arrow f`\Dotar{f}` $\overset{f}{\cdots\!\!\rightarrow}$

monomorphism f`\Mono{f}` $\overset{f}{\rightarrowtail}$

epimorphism f`\Epi{f}` $\overset{f}{\twoheadrightarrow}$

bimorphism f`\Bimo{f}` $\overset{f}{\rightarrowtail\!\!\!\!\twoheadrightarrow}$

isomorphism f`\Iso{f}` $\xrightarrow{\cong f}$

pair of parallel arrows f, g`\Biar{f}{g}` $\underset{g}{\overset{f}{\rightrightarrows}}$

pair of adjoint arrows f, g`\Adjar{f}{g}` $\underset{g}{\overset{f}{\leftrightarrows}}$

backward plain arrow f`\Bkar{f}` $\xleftarrow{f}$

backward dotted arrow f`\Bkdotar{f}` $\overset{f}{\leftarrow\!\!\cdots}$

backward monomorphism f`\Bkmono{f}` $\overset{f}{\rightarrowtail}$

backward epimorphism f`\Bkepi{f}` $\overset{f}{\twoheadleftarrow}$

backward bimorphism f`\Bkbimo{f}` $\overset{f}{\rightarrowtail\!\!\!\!\twoheadleftarrow}$

backward isomorphism f`\Bkiso{f}` $\xleftarrow{\cong f}$

pair of parallel backward arrows f, g `\Bkbiar{f}{g}` $\underset{g}{\overset{f}{\leftrightarrows}}$

backward pair of adjointarrows f, g `\Bkadjar{f}{g}` $\underset{g}{\overset{f}{\leftrightarrows}}$

triple of parallel arrows f, g, h `\Triar{f}{g}{h}` $\begin{array}{l}\xrightarrow{f}\\ \xrightarrow{g}\\ \xrightarrow{h}\end{array}$

backward triple of parallel arrows f, g, h `\Bktriar{f}{g}{h}` $\begin{array}{l}\xleftarrow{f}\\ \xleftarrow{g}\\ \xleftarrow{h}\end{array}$

triple of adjoint arrows f, g, h `\Triadjar{f}{g}{h}` $\begin{array}{l}\xleftarrow{f}\\ \xrightarrow{g}\\ \xleftarrow{h}\end{array}$

backward triple of adjoint arrows f, g, h `\Bktriadjar{f}{g}{h}` $\begin{array}{l}\xrightarrow{f}\\ \xleftarrow{g}\\ \xrightarrow{h}\end{array}$

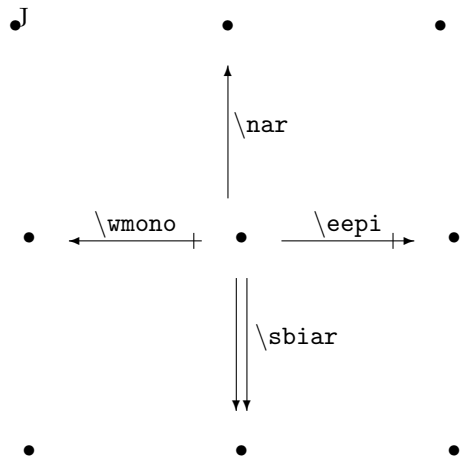
6 Primary diagram arrows.

The *diagram arrows* are those arrows specially designed to fit nicely in a diagram; their variety and their names are the same as those of in-text arrows: plain arrow (`ar`), dotted arrow (`dotar`), monomorphism (`mono`), epimorphism (`epi`), bimorphism (`bimo`), isomorphism (`iso`), pair of parallel arrows (`biar`), equality (`eql`), pair of adjoint arrows (`adjar`), triple of parallel arrows (`triar`), triple af adjoint arrows (`triadjar`), quadruple of parallel arrows (`quadriar`), quadruple of adjoint arrows (`quadriadjar`), quintuple of parallel arrows (`quintiar`) and quintuple of adjoint arrows (`quintiadjar`). You should never use an in-text arrow in a diagram nor a diagram arrow in a line of text: the technical characteristics of those pictures are just incompatible!

While an in-text arrow is always horizontal, the possibility must exist to give a diagram arrow a rather arbitrary direction. For that reason the name of a diagram arrow starts *always* with the prefix indicating its direction, even in the case of an horizontal right pointing arrow. The direction of an arrow is indicated by a direction of the compass-card, using the classical abbreviations:

north(`n`)south(`s`)east(`e`)west(`w`)

The command for producing a primary arrow is obtained by typing the abbreviation for its direction followed by the abbreviation for the type of arrow. For example the command `\wbimo` will produce a west pointing bimorphism. Here are other examples.



The case of adjoint arrows requires a comment. For east and west multiples of adjoint arrows, the direction is that of the lower arrow; for north and south pairs of adjoint arrows, the direction is that of the left arrow.

The standard length of a primary arrow is 50pt, but the length of an *horizontal* arrow is automatically adjusted when the adjacent vertices have too long

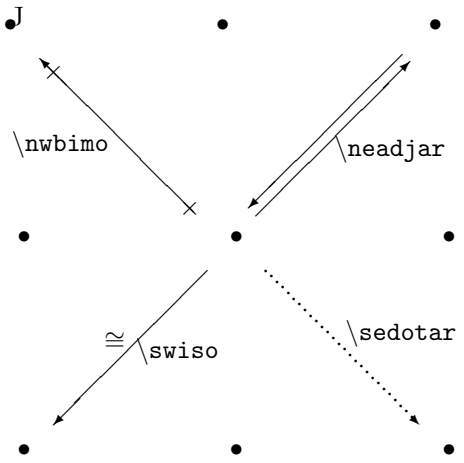
names.

7 Secondary diagram arrows.

The secondary diagram arrows are those which are oriented in the secondary directions of the compass-card; here are thus the possible directions and the corresponding abbreviations:

north-east (`ne`)Isouth-east (`se`)
north-west (`nw`)south-west (`sw`)

The variety of available arrows is reduced to the simple and double arrows: plain arrow (`ar`), dotted arrow (`dotar`), monomorphism (`mono`), epimorphism (`epi`), bimorphism (`bimo`), isomorphism (`iso`), pair of parallel arrows (`biar`), equality (`eql`) and pair of adjoint arrows (`adjar`). Just as for primary arrows, a secondary arrow is obtained by typing the abbreviation for its direction followed by the abbreviation for the type of arrow. For example the command `\swbiar` will produce a pair of sout-west pointing arrows. Here are some examples of what you can produce as secondary diagram arrows:



The case of a pair of adjoint arrows requires again a comment. The direction is always that of the lower arrow.

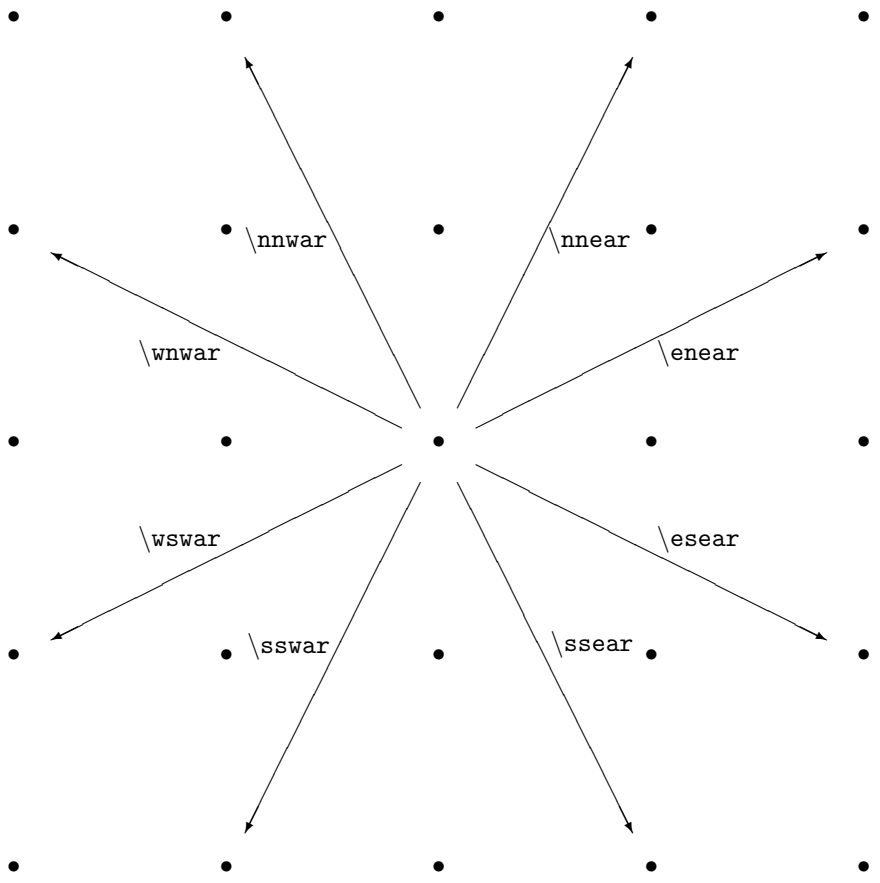
The secondary diagram arrows have a standard horizontal extent equal to 58pt.

8 Ternary diagram arrows.

The ternary diagram arrows are those oriented along the diagonals of a rectangle of sides 1,2; they have a standard horizontal extent equal to 66pt or 132pt, according to the case. Taking some liberty with trigonometry and geography, we shall name them after the ternary directions of the compass-card:

north-north-east (`nne`)east-north-east (`ene`)
east-south-east (`ese`)south-south-east (`sse`)
south-south-west (`ssw`)west-south-west (`wsw`)
west-north-west (`wnw`)north-north-west (`nnw`)

The only type of ternary arrows provided by the *Diagram* Macros are the plain arrows. Again you will obtain the command for a ternary arrow by preceeding the abbreviation `ar` with the prefix indicating the direction of the arrow. Here are thus the eight available ternary arrows:



9 Naming the diagram arrows.

The rules for naming the diagram arrows are analogous to those for naming the in-text arrows, with the additional possibility of naming the arrow on either of its sides. It suffice to type the first or the last letter of the command as an upper-case letter and give the name of the arrow as an argument. Here is an exhaustive list of the available possibilities.

Eql: An equality cannot be named

Quadriar, quadriadjar, quintiar, quintiadjar Quadruple and quintuple arrows cannot be named

Vertical ar, mono, epi, bimo, dotar: A first upper-case letter produces a left name and a last upper-case letter produces a right name; thus type `\Smono{f}` to draw a south monomorphism with left name f and `\nepI{g}` to draw a north epimorphism with right name g.

Non-vertical ar, mono, epi, bimo, dotar: A first upper-case letter produces an upper name and a last upper-case letter produces a lower name; thus type `\War{f}` to draw a west arrow with upper name f and `\nebimO{g}` to draw a north-east bimorphism with lower name g.

Horizontal iso: A first upper-case letter produces an upper name and a last upper-case letter produces a lower name; thus type `\Wiso{f}` to draw a west isomorphism with upper name f and `\eisO{g}` to draw an east isomorphism with lower name g.

Non-horizontal iso: Type a first upper-case letter to name it; the name will appear on one side of the arrow and the "isomorphism symbol" on the other side; thus `\Nwiso{f}` draws a north-west isomorphism with name f.

Vertical biar, adjar: Type a first upper-case letter to name them and give successively the name of the left arrow and that of the right arrow; thus `\Sbiar{f}{g}` draws a pair of south arrows with left name f and right name g.

Non-vertical biar, adjar: Type a first upper-case letter to name them and give successively the name of the upper arrow and that of the lower arrow; thus `\Seadjar{f}{g}` will produce a south-east pair of adjoint arrows with upper name f and lower name g.

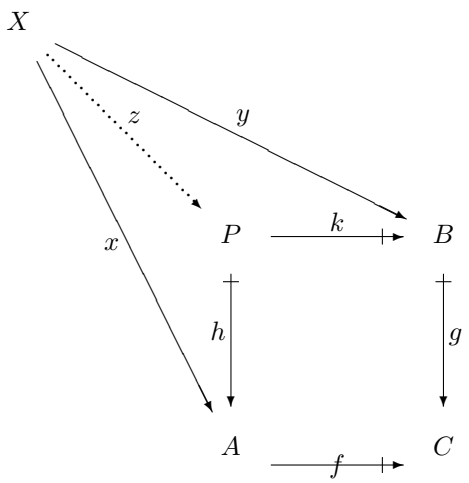
Vertical triar, triadjar A first upper-case letter produces left names and a last upper-case letter produces right names; give successively the names of the left, central and right arrow; thus `\Ntriar{f}{g}{h}` draws a triple of north arrows with left names f, g, h.

Horizontal triar, triadjar A first upper-case letter produces upper names and a last upper-case letter produces lower names; give successively the names of the upper, central and lower arrow; thus `\Wtriar{f}{g}{h}` draws a triple of west arrows with left names f, g, h.

The name of an arrow will always be processed in Math mode at the textstyle size; therefore the text characters will be printed in Math italic and you should not type any $ sign to introduce a mathematical symbol in the name of an arrow. The name of an arrow will automatically be positioned in order not to bump into the corresponding arrow.

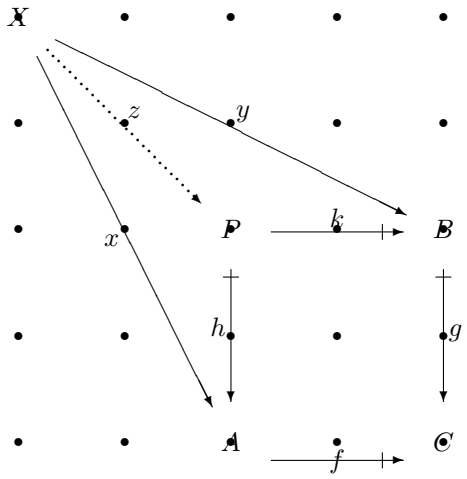
10 Conceiving a diagram.

We are now arriving at the central section of this user's guide: the one which describes the basic principle for realizing a diagram. This will be better explained on an example. Let us therefore suppose you want to construct the following diagram.



This diagram contains several items: five vertices A, B, C, P, X, four primary arrows, one secondary arrow and two ternary arrows; one of the arrows is emphasized. All those items are very different in nature, size, orientation, position ... The first basic rule for conceiving a diagram is to ignore completely all those differences and just treat equally all those items; this allows a maximum of flexibility and simplicity. To realize this, the *Diagram* Macros consider a formal pattern of points, at a 40pt distance from each other, both horizontally and vertically. Then each item of the diagram is *centered* at one of the points of the pattern, using the *picture* environment.

Let us for example visualize the pattern of points on which the previous diagram has been constructed.



The complete pattern needs not be a square nor even a rectangle. It is just compulsory to start all the lines from a same (left) column and leave no hole at all in any line; but lines need not be of equal length. To fill in the holes which could appear inside a line, it suffices to attach empty items at the corresponding points of the pattern. Here is for example the set of items corresponding to the previous diagram.

X				
	\Sedotar{z}	\Esear{y}		
	\sseaR{x}	P	\Eepi{k}	B
		\Smono{h}		\smonO{g}
		A	\eepI{f}	C

11 Typing a diagram.

Let us assume you have determined the nature and the position of each item of your diagram. Here is what you should type in order to give that information to the *Diagram* Macros.

1. Type the command `\DIAG` to start the typing of the diagram; this command will in particular start a *center* environment; it can be given both in paragraph and in math mode.
2. Type between curly brackets the description of *every* item; this description will automatically be processed in Math mode. The empty items will thus appear as {} in your input.
3. *Separate* two consecutive items in a same row by the "next item command", which is `\n`. The command `\n` may not appear before the first item of a row nor after the last item of a row.
4. *Separate* two consecutive rows of items by the "next row command" which is `\nn`. The command `\nn` may not appear before the first row of items nor after the last row of items.
5. Type the command `\diag` to indicate the end of the diagram typing.

When you type a diagram using the procedure just indicated, notice that between the two commands `\DIAG` and `\diag`

- You can put as much space as you want in a line to make your input file easier to read;
- You should never use a $ sign to type a mathematical symbol, except if you did include that symbol in a box.

With all that in mind, the diagram of the previous section can be typed as follows

```
\DIAG
{X}                                                    \nn
{} \n{\Sedotar{z}}\n{\Esear{y}}                        \nn
{} \n{\sseaR{x}}  \n{P}        \n{\Eepi{k}}\n{B}       \nn
{} \n{}           \n{\Smono{h}}\n{}       \n{\smonO{g}}\nn
{} \n{}           \n{A}        \n{\eepI{f}}\n {C}
\diag
```

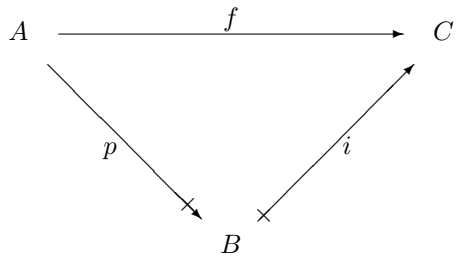
or equivalently as

```
\DIAG{X}\n{}\n{}\n{}\n{}\nn{}\n{\Sedotar{z}}
\n{\Esear{y}}\n{}\n{}\nn{}\n{\sseaR{x}}\n{P}
\n{\Eepi{k}}\n{B}\nn{}\n{}\n{\Smono{h}}\n{}
\n{\smonO{g}}\nn{}\n{}\n{A}\n{\eepI{f}}\n{C}\diag
```

The *Diagram* Macros will automatically center the diagram horizontally on the page and will take care of the spacing with the preceeding and the following text. When a diagram is rather big, you should make it a figure or a table (see LaTeXbook, 3.5.1 page 60) to improve the quality of the page-making. If you want to align horizontally various small diagrams, consider these just as a unique wide diagram where some columns are completely empty.

12 The variable length option.

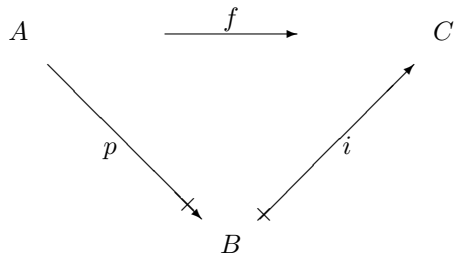
Suppose you want to produce the following diagram:



Following the instructions of the previous sections you will type:

```
\DIAG
{A}\n{}          \n{\Ear{f}}\n{}          \n{C}\nn
{} \n{\seepI{p}}\n{}       \n{\nemonO{i}}\nn
{} \n{}          \n{B}      \nn
\diag
```

and you will get



which is not exactly what you wanted!

The *Diagram* Macros assume that a primary arrow connects two vertices attached at the two horizontally adjacent points of the pattern, and a corresponding assumption is made for the secondary and the ternary arrows. The east arrow of the previous diagram does not satisfy that assumption: it connects vertices situated at a four intervals distance. The *Diagram* Macros provide you with a *variable length option* which allows you to modify the length of an arrow in order to take care of such particularities. The "variable length option" is a special feature, which means that the *Diagram* Macros need your help to handle the problem. In fact, what you have to do is to decide the new length of the arrow.

Not all arrows have been provided with a "variable length option"; here are the possibilities:

- All the primary arrows, named or unamed, admit a variable length option.
- The four secondary plain arrows, named or unamed, admit a variable length option; the other secondary arrows are not provided with the variable length option.
- The ternary arrows do not have the variable length option.

To take advantage of the variable length option, it suffices to follow the name of the corresponding command with the lower-case letter `v` and give to that command an additional last argument which will be

- The new length of the arrow in the case of a primary arrow
- The new horizontal extent of the arrow in the case of a secondary arrow;

those two lengthes must be expressed in points. The new arrow will still be centered with respect to the corresponding point of the pattern. For example typing `\Earv{f}{130}` for the east arrow of the previous diagram will produce the desired result as in the first diagram. Typing `\smonOv{g}{30}` will produce a south monomorphism with right name g and length 30 points, typing `\Wadjarv{f}{g}{100}` will produce a west pair of adjoint arrows with names f, g and length 100 points and typing `\near{82.5}` will produce an unamed north-east arrow with horizontal extent 82.5 points.

The standard length of a primary arrow is 50pt; since each interval of the pattern has length 40pt, the new length of a primary arrow will be

$$50 + (n \times 40)$$

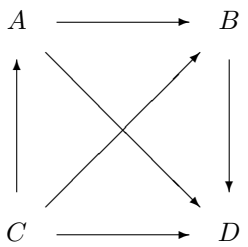
where n is the number of intervals the arrow crosses. In the same way the horizontal extent of a secondary arrow is 58pt, so that the new horizontal extent of a secondary arrow will be

$$58 + (n \times 40)$$

where n is the number of horizontal intervals the arrow crosses.

13 Superposing items in a diagram.

Sometimes you will like to superpose two items at the same point of the pattern underlying your diagram, like in the following example:



The command for superposing two items is `\cross{1}{2}` where 1 and 2 stand for the two items; the two arguments 1 and 2 are automatically processed in Math mode. The previous diagram can thus be typed as

```
\DIAG
{A}    \n {\ear}                  \n {B}    \nn
{\nar} \n {\cross{\near}{\sear}} \n {\sar} \nn
{C}    \n {\ear}                  \n {D}
\diag
```

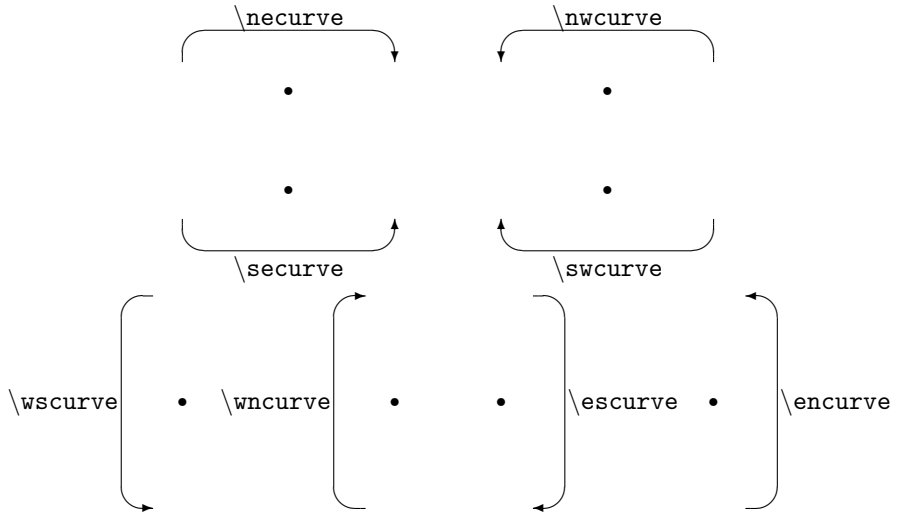
When superposing two arrows, the `\cross` command can have the undesired effect that one of the arrows will bump into the name of the other arrow. A special command has been designed to correct such a defect and is explained in section 17. For that reason, the "cross" option could be considered as a special feature of the *Diagram* Macros: one which requires your help.

14 Drawing curved arrows.

Drawing curved arrows is another special feature of the *Diagram* Macros, thus a process where the *Diagram* Macros need you help. In fact you will have to decide the length of the curved arrow and possibly you will have to adjust the spacing around the diagram (an easy procedure for that will be described in section 17).

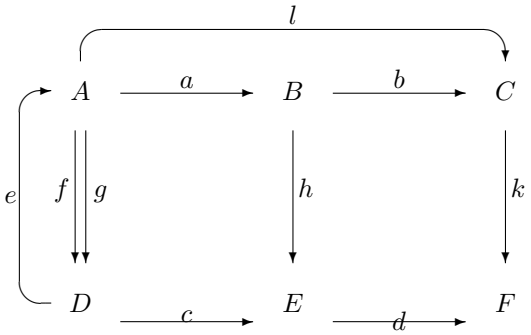
Just plain curved arrows in the primary directions are provided by the *Diagram* Macros. The first letter of the command indicates where, with respect to the diagram, the arrow has to stand (this determines the concavity of the arrow); the second letter indicates the overall direction of the arrow; then follows the abbreviation `curve` for "curved arrow". As usual, the two first indications are given using the compass-card terminology: `n,s,e,w`. Typing the first letter as an upper-case letter will produce a named arrow, the name being given as a first argument of the command. It is compulsory to give the length of the arrow as a last argument. By length of the arrow we mean clearly its horizontal or vertical extent (according to the case), expressed in points. For example `\Necurve{f}{160}` will produce a curved arrow with name f and length 160 points, to be positioned at the north side of a diagram and with overall east direction.

For more clarity, here are the eight unamed curved arrows (of length 80 points) and the corresponding commands. The bullets indicate the position of the formal centers (?!) of these arrows; each bullet corresponds thus to the point which will coincide with the corresponding point of the pattern.



Most often you will have to include a curved arrow at a point of the pattern which is already occupied by another item, a vertex or an arrow; just use the

`\cross` command of the previous section to superpose the two items. The length of a curved arrow will generally be a multiple of 80 points, since this is the normal distance between two vertices, in a primary direction. Here is an example of a diagram containing curved arrows:



It should thus be typed as follows:

```
\DIAG
{A}\n{\Ear{a}} \n{\cross{B}{\Necurve{l}{160}}}
                          \n{\Ear{b}}\n{C}      \nn
{\cross{\Wncurve{e}{80}}{\Sbiar{f}{g}}}
   \n{}        \n{\saR{h}}\n{}       \n{\saR{k}}\nn
{D}\n {\eaR{c}}\n{E}      \n{\eaR{d}}\n{F}
\diagv{25}{13}{0}
```

15 The free slope arrows.

When you have to produce a given diagram of vertices and arrows, you must clearly choose adequately the position of each vertex so that all the arrows will be in one of the sixteen directions considered by the *Diagram* Macros. Anybody who has read this user's guide will easily produce an example of a diagram where this requirement cannot be fulfilled. Nevertheless such a situation is rather unusual since, for example, the most popular book on category theory *(Saunders MAC LANE – Categories for the working mathematician – Springer,1971)* . . . does not contain any such diagram! [Well, I have been lucky: Saunders did not draw the connecting morphism in the snake lemma!]

The slope of an arrow is a pair (n, m) of integers: when you move n points left, you move m points up; as usual, a negative value of n or m indicates a movement in the opposite direction. For example, here are the slopes of the various diagram arrows:

- The primary arrows have slope $(\pm 1, 0)$ [horizontal arrows] or $(0, \pm 1)$ [vertical arrows];
- The secondary arrows have slope $(\pm 1, \pm 1)$;
- The ternary arrows have slope $(\pm 1, \pm 2)$ or $(\pm 2, \pm 1)$ according to the case.

In fact, LaTeXallows you to draw arrows in thirty two other directions, using the picture environment and the commands `\put` and `\vector`; the corresponding available slopes are $(\pm 1, \pm 3)$, $(\pm 3, \pm 1)$, $(\pm 1, \pm 4)$, $(\pm 4, \pm 1)$, $(\pm 2, \pm 3)$, $(\pm 3, \pm 2)$, $(\pm 3, \pm 4)$,
$(\pm 4, \pm 3)$. The length of an arrow is always specified by giving the horizontal extent of the arrow, except in the case of vertical arrows where the actual length is given. So you can just design any arrow or set of arrows using the picture environment and choose the corresponding picture as an item of your diagram; the center of your picture will be positioned automatically at the corresponding point of the pattern. This is exactly what the *Diagram* Macros are doing any time you ask them to draw an arrow.

You can avoid some typing (but just that!), by using the "free slope arrow Macro" to produce directly a diagram arrow, like for primary, secondary and ternary arrows. It has the following form where, as a matter of convention, the origin of the coordinates is the point of the pattern at which you want to include the free slope arrow:

`\Freear{n}{a}{b}{x}{y}{u}{v}{l}`
where

- `n` is the name of the arrow;
- `(a,b)` are the coordinates (in points) of the place where you want the lower left corner of that name to be positioned;

- `(x,y)` are the coordinates of the origin of the arrow;
- `(u,v)` is the slope of the arrow (it must be one of the slopes just indicated);
- `l` is the horizontal extent (in points) of the arrow, or its length in the case of a vertical arrow.

If you do not want to name the arrow, just type, following the usual *Diagram* Macros convention:

```
\freear{x}{y}{u}{v}{l}
```

Clearly, the "free slope arrow" option is not a *Diagram* Macros feature: it is just plain LaTeX job.

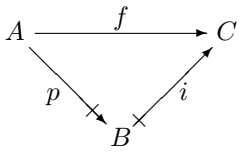
16 The scaling option.

The "variable length" option has allowed us to design properly the diagram of §12. But you can possibly not be satisfied with the final product, because the resulting diagram is unusually big for the few arrows it does contain. This is easily corrected!

Replacing the initial command `\DIAG` by the corresponding "variable scale" command `\DIAGV{n}` will scale the original pattern of points by n% and adjust correspondingly the length of every arrow in the diagram. For example typing

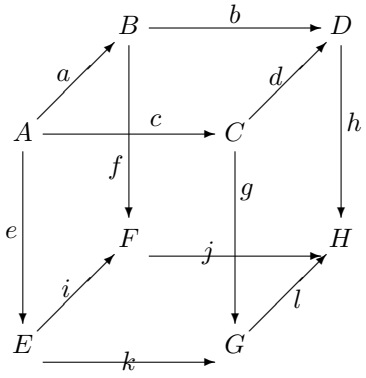
```
\DIAGV{50}
{A}\n{}          \n{\Earv{f}{130}}\n{}          \n{C}\nn
{} \n{\seepI{p}}\n{}              \n{\nemonO{i}}\nn
{} \n{}          \n{B}
\diag
```

will produce the following result



Since text characters (and various other things) are unaffected by the scaling process, you should avoid reducing exagerately a diagram. In most cases, a reduction up to 50% gives excellent results. On the other hand, an enlargement can be helpful when handling vertices with very long names.

This "variable scale" option is particularly useful when designing "three dimensional" diagrams, which require very often a rather dense pattern of reference points and many "long" arrows produced via the "variable length" option. For example to design the following cube



it suffices to type

```
\DIAGV{50}
{} \n{} \n{B} \n{} \n{\Earv{b}{130}} \n{} \n{D} \nn
{} \n{\Near{a}} \n{} \n{} \n{} \n{\Near{d}} \n{} \nn
{A} \n{} \n{\cross{\Earv{\movename{c}{10}{0}}{130}}
  {\Sarv{\movename{f}{0}{-13}}{130}}}
  \n{} \n{C} \n{} \n{\saRv{h}{130}} \nn
{} \nn
{\Sarv{e}{130}} \n{} \n{F} \n{}
  \n{\cross{\eaRv{\movename{j}{-10}{0}}{130}}
  {\saRv{\movename{g}{0}{13}}{130}}}
  \n{} \n{H} \nn
{} \n{\Near{i}} \n{} \n{} \n{} \n{\neaR{l}} \nn
{E} \n{} \n{\eaRv{k}{130}} \n{} \n{G}
\diag
```

where the `\movename` command is explained in the next section.

17 Final adjusments.

The *Diagram* Macros are intended to provide you with a powerful tool for designing diagrams, so in normal circumstances, no adjusment at all should be necessary. But in some unusual cases, for example when you use the special features of the *Diagram* Macros, you can want to interact for correcting some imperfections. Some Macros have been designed to help you realizing this.

If the vertices of a same row have names of different heights, those names will no longer stand at the same altitude since they are centered at the corresponding points of the pattern. There is a "normalizing" command `\N` which you can apply to the "high vertices" in order to correct this defect: instead of typing `{vertex}` type `{\N{vertex}}`, where `vertex` stands here for the name of the vertex.

You can want to move the name of an arrow, especially when you have used the `\cross` command to superpose two named arrows. If f is the name of the arrow, instead of `{f}` you just type `{\movename{f}{n}{m}}` where n and m are integers (positive or negative); the name of the arrow will be moved `n` pt right and `m` pt up. In the same way the commands `\movearrow{\Ear{f}}{n}{m}` and `\moveverterx{A}{n}{m}` will move respectively the east arrow f and the vertex A, `n` pt right and `m` pt up.

Because the curved arrows are very "excentric" with respect to their formal center, they will cause problems of spacing when drawn along an outer edge of a diagram. Drawing a curved arrow along the upper or the lower edge of a diagram will generally cause the diagram to overlap the surrounding text; drawing a curved arrow along just one vertical side of a diagram will cause bad horizontal centering. Those defects can be corrected by using the "variable spacing" option of the diagram commands. Just replace the final command `\diag` by the corresponding "variable spacing" command `\diagv{t}{l}{b}` where

- `t` is the vertical space, expressed in points, to be added at the top of the diagram;
- `l` is the horizontal space, expressed in points, to be added left of the diagram;
- `b` is the vertical space, expressed in points, to be added at the bottom of the diagram.

This command will not work properly if your diagram is wider than the textwidth; in that case, include the diagram in a mini-page environment (see LaTeXbook, page 98) wide enough to contain it and position this mini-page in your document using, for example, the commands `\hspace*` and `\vspace*`.

Finally, some unexpected troubles can occur if you introduce in a diagram items which have not been designed in a way which is compatible with the internal structure of the *Diagram* Macros. You will most often get rid of the problem by including the new item in a box of formal dimensions (0,0).

18 Some special characters.

This last section contains some *text* symbols which you can want to use in connection with the *Diagram* Macros.

The "commutative diagram" symbol ↻ can be produced by giving the command `\com` and the "adjoint" symbol ⊣ is obtained via the command `\adj`.

Finally here are some commands for designing natural transformations, with or without name.

`\nat` 

`\Nat{F}{α}{G}` 

`\binat` 

`\Binat{F}{α}{G}{β}{H}` 

In the case of named natural transformations, the names of the functors and the natural transformations are processed in Math mode; thus do not include any $ sign for putting a mathematical symbol in those names. Here is how those symbols will be positioned in a line of text:

A  B  C