AbsDiagXy Manual

Gerd Zeibig
Deptartment of Mathematics, Kent State University
gzeibig@math.kent.edu

October 20, 2003

1 Purpose and Method of AbsDiagXy

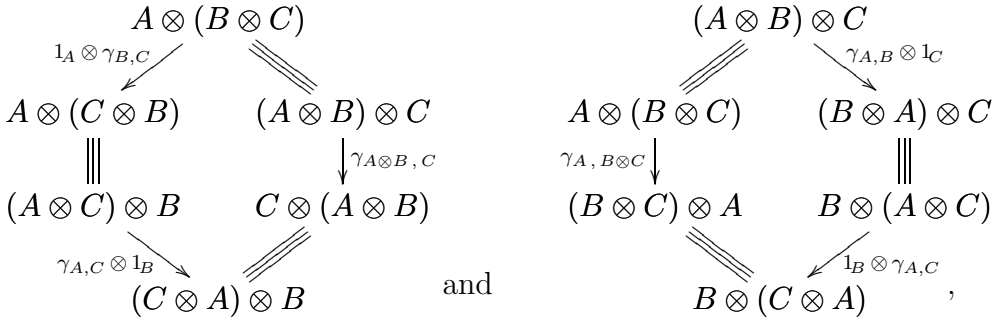
The aim of this package is to provide a simple way of typing complex, irregular diagrams. I wrote this package in order to type the article 'Products of Monoids' [4], which contains many diagrams of this type.

This package is a Latex front end that uses Michael Barr's package DiagXy, which in turn uses Kristoffer H. Rose's XY-pic. The name *AbsDiagXy* is intended to suggest that this package is based on *DiagXy*, but makes more extended use of *Abs*olute coordinates.

The package AbsDiagXy permits to plot diagrams in the following way: Given a diagram that needs to be typed,

- start by numbering its vertices $1, 2, 3, \ldots$.
- Next, define the objects one, two, three ... as the expressions which should constitute the vertices.
- Then, enter the positions at which the vertices are supposed to be placed.
- Finally, plot the arrows. At this point one can simply instruct AbsDiagXy to plot an arrow from object 8 to object 5, say. There is no need to label the source and the target again. Likewise, at this stage, there is no need to worry about the positions of the source and the target.

Example: The two hexagons



which state the axioms a braiding has to satisfy in a monoidal category, have been plotted using the following code:

```
\def\obone{A\otimes(B\otimes C)}%
\def\obtwo{A\otimes(C\otimes B)}%
```

```
\def\obthree{(A\otimes B)\otimes C}%
\def\obfour{(A\otimes C)\otimes B}%
\def\obfive{C\otimes (A\otimes B)}%
\def\obsix{(C\otimes A)\otimes B}%
%
\def\obseven{(A\otimes B)\otimes C}%
\def\obeight{A\otimes(B\otimes C)}%
\def\obnine{(B\otimes A)\otimes C}%
\def\obten{(B\otimes C)\otimes A}%
\def\obeleven{B\otimes(A\otimes C)}%
\def\obtwelve{B\otimes(C\otimes A)}%
%
%
\posone(400,0)%
\postwo(0,-300)%
\posthree(800,-300)%
\posfour(0,-600)%
\posfive(800,-600)%
\possix(400,-900)%
%
\posseven(2200,0)%
\poseight(1800,-300)%
\posnine(2600,-300)%
\posten(1800,-600)%
\poseleven(2600,-600)%
\postwelve(2200,-900)%
%
$$\bfig%
%
\arrow<1,2>|l|(.65)[1\!_A\,\otimes\,\gamma_{B,C}]%
\identify<1,3>%
\identify<2,4>%
\arrow<3,5>|r|[\gamma_{A\otimes B\,,\,C}]%
\arrow<4,6>|l|(.3)[\gamma_{A,C}\,\otimes\,1\!_B]%
\identify<5,6>%
%
\identify<7,8>%
\arrow<7,9>|r|(.65)[\gamma_{A,B}\,\otimes\,1\!_C]%
```

```
\arrow<8,10>|l|[\gamma_{A\,,\,B\otimes C}]%
\identify<9,11>%
\identify<10,12>%
\arrow<11,12>|r|(.3)[1\!_B\,\otimes\,\gamma_{A,C}]%
%
\place(1200,-850)[\mbox{and}]%
\place(2750,-850)[,]%
%
\efig$$%
```

2 Availability

The package AbsDiagXy can be downloaded from
`http://www.math.kent.edu/~gzeibig/absdiagxy/absdiagxy.zip`.

It may be freely copied, distributed, and used. However, if it is modified, it has to be renamed. Also, if it is made available to others, the entire zip-file, with all its contents, has to be made available.

AbsDiagXy is an extension of Michael Barr's package DiagXy. His package can be downloaded at `ftp://ftp.math.mcgill.ca/pub/barr/diagxy.zip`, or by anonymous ftp from `ftp.math.mcgill.ca//pub/barr/diagxy.zip`

DiagXy is based on Kristoffer Rose's famous XY-pic. XY-pic, Version 3.7, should already be installed on any computer system related to mathematics or computer science. Otherwise, it can be found at
`http://www.ctan.org/tex-archive/macros/generic/diagrams/xypic/`,
for instance. The XY-pic homepage is now located at
`http://www.tug.org/applications/Xy-pic/`.

At this point I would like to sincerely thank both authors for their beautiful packages, and for making them freely available to everyone.

3 Commands

In order to use AbsDiagXy in an article, the article's source-file has to load AbsDiagXy with the command

```
\input{absdiagxy}
```

This will automatically load DiagXy and Xy-pic.

Remark: AbsDiagXy uses a lot of variables to store the coordinates of the vertices. The default version of AbsDiagXy allows to store up to 50 vertices. Given that each vertex requires two variables, one for its x-coordinate and one for its y-coordinate, these vertices consume a total of 100 variables. In case this causes problems with other packages, I have included the following light versions in the `absdiagxy.zip` file: AbsDiagXy30, AbsDiagXy35, AbsDiagXy40 and AbsDiagXy45. They function exactly as AbsDiagXy does, except that they are only capable of plotting diagrams with up to 30, 35, 40 and 45 vertices, respectively. To activate one of these light versions, replace the command `\input{absdiagxy}` accordingly by `\input{absdiagxy30}`, `\input{absdiagxy35}`, `\input{absdiagxy40}`, or `\input{absdiagxy45}`.

3.1 Labelling and Positioning of Vertices

The vertices of the diagram to be typed need to be numbered. Then they are entered using the commands

```
\def\obone{...}%
\def\obtwo{...}%
\def\obthree{...}%
...
```

The numbers one to fifty have been spelled out in the Appendix.

Next, the positions of these vertices have to be entered using the following commands:

```
\posone(...,...)%
\postwo(...,...)%
\posthree(...,...)%
...
```

In each of these commands, the first coordinate is the x-coordinate of the corresponding object; the second coordinate is the y-coordinate. For example, the vertex number one is placed on the origin using the command `\posone(0,0)`. The command `\postwo(500,0)` then places the vertex number two 500 `\ul`'s to the right of the vertex one. The length of a unit length `\ul` is user-assignable; the default value is .01em.

The above part of typing a diagram can be done outside the Latex math-mode.

3.2 DiagXy Commands

Since AbsDiagXy is an extension of DiagXy, all the commands of DiagXy obviously can be used once AbsDiagXy is loaded. The inline commands such as `\to` have been particularly useful for me. For the DiagXy commands, see Michael Barr's DiagXy manual [1].

Further, for users accustomed to the syntax of DiagXy, the command `\diagxyarrow` has been included in AbsDiagXy. Its syntax is

```
\diagxyarrow<a,b>|p|/sh/[L]
```

This command plots an arrow from the vertex number `a` to the vertex number `b`. The remaining parameters are all optional and have exactly the same meaning as in the package DiagXy, see the explanations in [1]. Note, in particular, the possibility of using the symbol @ in the parameter `/sh/`.

3.3 AbsDiagXy Plotting Commands

The following commands serve to actually plot something. They have to be entered in math-mode, between the the commands `\bfig` and `\efig`. Usually one would want the diagram to appear in display-mode. This can be accomplished using the commands `$$\bfig` and `\efig$$`.

• The `\arrow` command.

This is the standard command to plot an arrow from one vertex to another. This command has two uses.

∗ The first, and recommended, use is as follows:

```
\arrow<a,b>‘curv,pshft‘/shape/|lpos|(rlpos)<xlshift,ylshift>[label]
```

Here `a` is the number of the source vertex of the arrow. It has to be written as a number, not spelled out in English. Likewise `b` is the number of the target vertex.

This pair of parameters is required. All the remaining parameters are optional, any subset of them may be used. However, the parameters that are used have to be typed in the above order.

◇ The parameter `curv` can be a (positive or negative) length, such as -.45em. A non-zero value gives the arrow a curving. The parameter `pshft` is a length, too. It shifts the arrow parallel to the direction it is pointing. If one of the parameters `curv`

or pshft is used, the other one also has to be entered. For instance, to get a 2em curving without parallel shift, use ‘2em,0em‘.

◇ The parameter /shape/ specifies the shape of the arrow. Its default is /->/. To get a dotted arrow, one could use /.>/. See [2] and [3] for more XY-pic arrow styles.

◇ The parameter |lpos| determines the label position in the following way: It can be any of the symbols a, b, l, r, m,pa, pb, pl, pr, or pm. The first five of these possibilities respectively place the label of the arrow above, below, to the left, to the right or in the middle of the shaft of the arrow. The last five possibilities have the same effect, except that they additionally plot to label of the arrow parallel to its shaft. The default setting of this parameter is a.

◇ The parameter (rlpos), the relative label position, is a number, usually between 0 and 1. The default value, .5, places the label of the arrow half way between the beginning and the end of its shaft. A value closer to 0 shifts the label more towards its source; a value closer to 1 shifts the label more towards its target.

◇ The parameter <xlshift,ylshift> consists in an ordered pair of lengths, as in <.07em,-.23em>. It allows to shift the label of the arrow relative to the position where it would be placed if this parameter were not used. This is particularly useful when labels are plotted parallel to their arrows. Indeed, in this case it is most often necessary to manually adjust the placement of the labels.

◇ The parameter [label] gives the actual expression by which the arrow is labelled. The expression label is automatically placed in math-mode. It will be produced in \scriptstyle size, unless this is changed. For instance, the the command \let\labelstyle=\textstyle causes the label to appear in \textstyle size.

∗ The second use of the command \arrow is

```
\arrow<a,b>"string"
```

It allows to plot an arrow from the object number a to the object number b, using the XY-pic command sequence string. For instance, the command \arrow<1,2>"@{=>}^{\labelstyle{f}}" has the same effect as the command \arrow<1,2>/=>/[f] - or as the command \arrow<1,2>/=>/|b|[f], depending on the positions of the vertices 1 and 2.

• The \identify command.

It connects two vertices by a headless triple arrow, that is, by an identification symbol. Its use is

```
\identify<a,b>‘curv,pshft‘
```

Again, the parameters ‘`curv,pshft‘` are optional, but if one of them is used, the other one also has to be entered. The parameters have the same meaning as in the command `\arrow`.

• The `\makeequal` command.

It connects two vertices by a headless double arrow, that is, by an equality symbol. Otherwise this command is exactly the same as the command `\identify`. Hence, its use is

```
\equiv<a,b>‘curv,pshft‘
```

Again, the parameters ‘`curv,pshft‘` are optional, but if one of them is used, the other one also has to be entered. The parameters continue to have the same meaning as in the command `\arrow`.

• The `\plotob` and `\plotobject` commands.

These two commands are synonyms of each other. They are used as in

```
\plotob<a>
```

They plot the object number `a` at its position, that is, at the position that has been chosen using the `\pos....(...,...)` command.

• The `\plotobjectsoneto` command.

It is used as in

```
\plotobjectsoneto<a>
```

This command plots the objects number 1, 2, ..., `a` at their positions.

• The `\plotallobjects` command.

This command does not admit or require any parameters. It plots all objects 1, 2, ... at their positions.

3.4 Other Commands

• The `\clearobjects` command.

This command does not admit or require any parameters. It clears all the objects (vertices) and resets all the vertex positions to (0,0). This command might be useful during the typing process. It could be included before the construction of a new diagram is started. It makes typing mistakes such as `\def\obeightteen{A\otimes B}` more obvious (note the erroneous `tt`). Indeed, if this kind of mistyping occurs in the definition of an object, no error message will be produced. If later an arrow such as `\arrow<18,15>` is requested, AbsDiagXy will take for this arrow's source the expression it finds in `\obeighteen`. If `\obeighteen` contained a different expression in a preceding diagram, AbsDiagXy might reuse this expression. If, however, the command `\clearobjects` has been inserted between these two diagrams, the source vertex will completely miss, rendering the typing mistake more visible.

This command should be deleted once the diagram is correctly typed, in order to avoid unnecessarily slow compiling of the source-file.

• The `\addmidspace command`.

This command is used as follows:

```
\addmidspace<x,y>
```

It only affects arrows that carry a label in their middle, that is, `\arrows` that have been given the parameter `|m|` or `|pm|`. It allows to specify the size of the hole that is punched in the arrow. More precisely, this command specifies how much bigger the hole is made, compared to the size of a hole in which the label of the arrow would tightly fit. The parameters `x` and `y` can be any lengths. The default value is `<1pt,4pt>`.

• The `\defaultaddmidspace` command.

This command does not admit or require any parameters. It resets `\addmidspace` to its default-value: it is a synonym for `\addmidspace<1pt,4pt>`.

4 Errant Spaces and Bugs

4.1 Errant Spaces

As Michael Barr explains in the manual [1] of his package DiagXy, in order to avoid seeing objects in double vision (objects almost on top of each other, but not exactly),

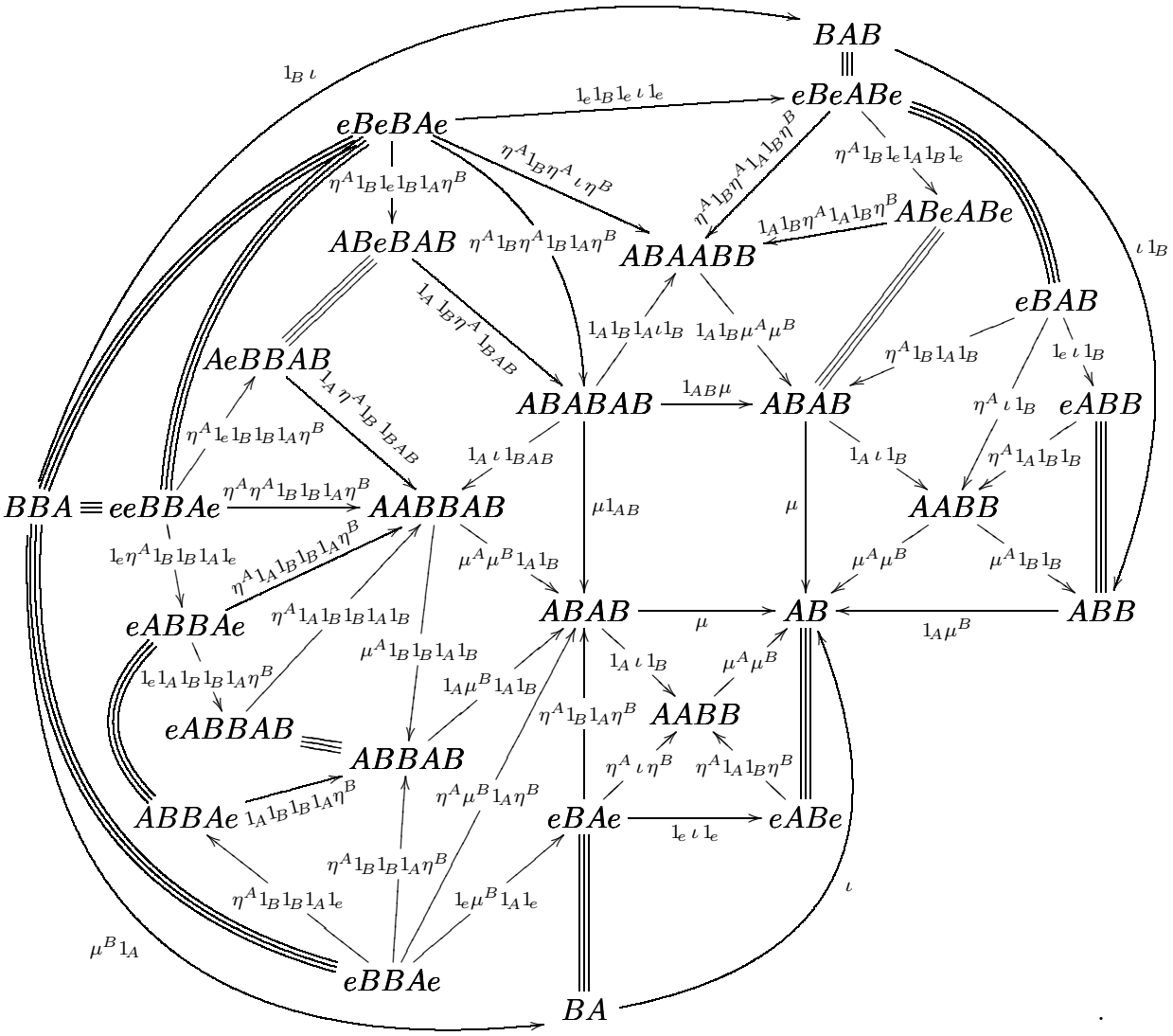
it is highly recommendable to use the symbol `%` extensively. This prevents unwanted errant spaces from appearing. The situation is similar with AbsDiagXy. To avoid problems, it is highly recommended to place the symbol `%` tightly, without any blanks, at the end of every line which stands between the commands `\bfig` and `\efig`. See the examples.

4.2 Bugs

There are problems, which seem to be related to the (otherwise astounding) XY-pic package, concerning arrows which are labelled in their middle. These arrows should ideally have a little hole in their shaft, which serves to accommodate the label. However, if an arrow is curved, a bug typically renders half its shaft invisible. Likewise there is a problem with arrows that point to the left and which are requested to be parallely labelled in the middle, as expressed by the parameter `|pm|`. A second bug (with strong opinions opposite to the first bug's ones) tends to prevent any hole at all from appearing. I have tried to insert a little hole manually. However, in this situation it might also be a good idea to consider placing the label above, below, or aside the arrow.

5 A larger Example

The following diagram is taken from the article 'Products of Monoids' [4]:



.

It has been typed as follows:

```
\def\obone{BBA}%
\def\obtwo{BAB}%
%\def\obthree{ABB}%
\def\obfour{AB}%
\def\obfive{BA}%
\def\obsix{eBeBAe}%
\def\obseven{eBeABe}%
\def\obeight{eBAB}%
\def\obnine{eABB}%
\def\obten{ABeBAB}%
\def\obeleven{ABAABB}%
\def\obtwelve{AeBBAB}%
\def\obthirteen{ABABAB}%
\def\obfourteen{ABAB}%
\def\obfifteen{eeBBAe}%
\def\obsixteen{AABBAB}%
\def\obseventeen{AABB}%
\def\obeighteen{eABBAe}%
\def\obnineteen{ABAB}%
\def\obtwenty{ABBAB}%
\def\obtwentyone{AABB}%
\def\obtwentytwo{eBBAe}%
\def\obtwentythree{eBAe}%
\def\obtwentyfour{eABe}%
\def\obtwentyfive{ABB}%
\def\obtwentysix{ABeABe}%
%
\def\obtwentyseven{.}%
\def\obtwentyeight{\circledast}%
%
\def\obtwentynine{eABBAB}%
\def\obthirty{ABBAe}%
%
\posone(-100,0)%
\postwo(2640,1600)%
%\posthree(3650,1000)%
```

```
\posfour(2500,-350)%
\posfive(1750,-1700)%
\possix(1100,1300)%
\posseven(2640,1400)%
\poseight(3350,700)%
\posnine(3500,350)%
\posten(1100,900)%
\poseleven(2100,850)%
\postwelve(675,500)%
\posthirteen(1750,350)%
\posfourteen(2500,350)%
\posfifteen(330,0)%
\possixteen(1250,0)%
\posseventeen(3000,0)%
\poseighteen(400,-400)
\posnineteen(1750,-350)%
\postwenty(1150,-850)%
\postwentyone(2125,-700)%
\postwentytwo(1100,-1600)%
\postwentythree(1750,-1050)%
\postwentyfour(2500,-1050)%
\postwentyfive(3500,-350)%
\postwentysix(3000,1000)%
%
\postwentyseven(3500,-1700)%
\postwentyeight(1200,450)%
%
\postwentynine(550,-750)%
\posthirty(400,-1050)%
%
%
$$\bfig%
%
\identify<1,15>%
\arrow<15,16>(.53)[\eta^A \eta^A 1\!_B 1\!_B 1\!_A \eta^B]%
%
\arrow<13,16>|m|[1\!_A\,\iota\,1\!_{BAB}]%
%
```

```
\arrow<16,19>|m|[\mu^A \mu^B 1\!_A 1\!_B]%
\arrow<16,20>|m|(.585)[\mu^A 1\!_B 1\!_B 1\!_A 1\!_B]%
\arrow<20,19>|m|<.2em,0em>[1\!_A \mu^B 1\!_A 1\!_B]%
\arrow<15,18>|m|(.33)[1\!_e \eta^A 1\!_B 1\!_B 1\!_A 1\!_e]%
\arrow<18,16>`0em,-.1em`|pa|(.53)<.4em,-.95em>[\eta^A 1\!_A 1\!_B 1\!_B 1\!_A \eta^B]%
\arrow<13,19>|r|[\mu 1\!_{AB}]%
\arrow<15,12>|m|<-1.36em,0em>[\eta^A 1\!_e 1\!_B 1\!_B 1\!_A \eta^B]%
\arrow<12,16>|pa|<-.8em,-1.15em>[1\!_A\, \eta^A 1\!_B \,1\!_{BAB}]%
%
\arrow<13,14>[1\!_{AB} \mu]%
\arrow<19,4>|b|(.46)[\mu]%
\arrow<14,4>|l|[\mu]%
%
\identify<12,10>%
\arrow<10,13>|pb|(.47)<1.25em,1.3em>[1\!_A\, 1\!_B \eta^A \,1\!_{BAB}]%
\arrow<6,10>|m|<-.22em,0em>[\eta^A 1\!_B 1\!_e 1\!_B 1\!_A \eta^B]%
\identify<1,6>`2em,0em`%
\arrow<6,13>`2em,0em`|m|[\eta^A 1\!_B \eta^A 1\!_B 1\!_A \eta^B]%
\arrow<6,11>`0em,.2em`|pa|(.58)<-.6em,-.45em>[\eta^A 1\!_B \eta^A \,\iota\,\eta^B]%
\arrow<13,11>|m|[1\!_A 1\!_B 1\!_A \iota 1\!_B]%
\arrow<11,14>|m|[1\!_A 1\!_B \mu^A \mu^B]%
%
\arrow<6,7>|pa|<0em,.25em>[1\!_e 1\!_B 1\!_e\,\iota\,1\!_e]%
\arrow<7,11>|pa|(.98)<-.1em,0em>[\eta^A 1\!_B \eta^A 1\!_A 1\!_B \eta^B]%
%
\identify<7,8>`2em,0em`%
%
\arrow<14,17>|m|[1\!_A\,\iota\,1\!_B]%
\arrow<17,4>|m|[\mu^A \mu^B]%
\arrow<17,25>|m|[\mu^A 1\!_B 1\!_B]%
\arrow<9,17>|m|<-.25em,0em>[\eta^A 1\!_A 1\!_B 1\!_B]%
\identify<9,25>%
%\arrow<25,4>`1em,0em`|b|[1\!_A \mu^B]%
\arrow<25,4>|b|[1\!_A \mu^B]%
\arrow<8,14>|m|[\eta^A 1\!_B 1\!_A 1\!_B]%
\arrow<8,9>|m|[1\!_e\,\iota\,1\!_B]%
\arrow<8,17>|m|[\eta^A\,\iota\,1\!_B]%
%
```

```
\arrow<7,26>|m|[\eta^A 1\!_B 1\!_e 1\!_A 1\!_B 1\!_e]%
\identify<26,14>%
\arrow<26,11>|pa|<0em,.3em>[1\!_A 1\!_B \eta^A 1\!_A 1\!_B \eta^B]%
%
\identify<2,7>%
\arrow<1,2>‘7em,.33em‘[1\!_B\,\iota]%
%
%\identify<3,9>%
%\identify<3,25>%
%\arrow<3,4>‘4em,0em‘[1\!_A \mu^B]%
\arrow<2,25>‘5.5em,.2em‘%
\arrow<2,25>‘0em,5.7em‘//|r|[\iota\,1\!_B]%
%
\arrow<19,21>|m|[1\!_A\,\iota\,1\!_B]%
\arrow<21,4>|m|[\mu^A \mu^B]%
%
\identify<1,22>‘-4.5em,0em‘%
\arrow<22,23>|m|<-.25em,0em>[1\!_e \mu^B 1\!_A 1\!_e]%
\arrow<22,20>|m|(.53)<.4em,0em>[\eta^A 1\!_B 1\!_B 1\!_A \eta^B]%
\arrow<22,19>|m|[\eta^A \mu^B 1\!_A \eta^B]%
\arrow<23,19>|m|<-.15em,0em>[\eta^A 1\!_B 1\!_A \eta^B]%
\arrow<23,21>|m|[\eta^A\,\iota\,\eta^B]%
\arrow<23,24>|b|[1\!_e\,\iota\,1\!_e]%
\arrow<24,21>|m|<.15em,0em>[\eta^A 1\!_A 1\!_B \eta^B]%
\identify<24,4>%
%
\arrow<1,5>‘-7.5em,-.5em‘|l|[\mu^B 1\!_A]%
\identify<5,23>%
\arrow<5,4>‘-5em,-.2em‘|r|<.2em,0em>[\iota]%
%
\plotob<27>%
%\plotob<28>%
%
\identify<18,30>‘-2em,-.45em‘%
\identify<29,20>%
\arrow<18,29>|m|[1\!_e 1\!_A 1\!_B 1\!_B 1\!_A \eta^B]%
\arrow<29,16>|m|(.52)<-.15em,0em>[\eta^A 1\!_A 1\!_B 1\!_B 1\!_A 1\!_B]%
\arrow<30,20>|pb|(.6)<-.2em,.32em>[1\!_A 1\!_B 1\!_B 1\!_A \eta^B]%
```

```
\identify<29,20>%
\arrow<22,30>|m|[\eta^A 1\!_B 1\!_B 1\!_A 1\!_e]%
\identify<15,6>‘2em,0em‘%
%
\efig$$%
```

6 Appendix

Below are the numbers one to fifty, listed in the spelling used by AbsDiagXy.

one, two, three, four, five, six, seven, eight, nine, ten, eleven, twelve, thirteen, fourteen, fifteen, sixteen, seventeen, eighteen, nineteen, twenty, twentyone, twentytwo, twentythree, twentyfour, twentyfive, twentysix, twentyseven, twentyeight, twentynine, thirty, thirtyone, thirtytwo, thirtythree, thirtyfour, thirtyfive, thirtysix, thirtyseven, thirtyeight, thirtynine, forty, fortyone, fortytwo, fortythree, fortyfour, fortyfive, fortysix, fortyseven, fortyeight, fortynine, fifty.

References

[1] Michael Barr, *A new diagram Package (Version 2003-05-10)*, available i.e. from ftp://ftp.math.mcgill.ca/pub/barr/diagxy.zip

[2] Kristoffer H. Rose, XY-pic *User's Guide*, February 16, 1999, available i.e. from the XY-pic Homepage at http://www.tug.org/applications/Xy-pic/

[3] Kristoffer H. Rose, Ross Moore, XY-pic *Reference Manual*, available i.e. from the XY-pic Homepage at http://www.tug.org/applications/Xy-pic/

[4] Gerd Zeibig, *Products of Monoids*, submitted. Preprint available at http://www.math.kent.edu/∼gzeibig/articles/productsofmonoids.pdf