Printing Z and Object-Z LaTeX documents

Paul King

Department of Computer Science
University of Queensland
Australia, 4072
king@batserver.cs.uq.oz.au

May 29, 1990

1 Introduction

This note describes a package of LaTeX macros for printing Z and Object-Z specifications. The macros and this note are based originally on Mike Spivey's `zed.sty` macros and documentation. The package does several related things for you:

- It loads extra fonts and defines mnemonics for the Z symbols they contain.
- It defines macros for some Z symbols (e.g. æ) which don't appear in any of our fonts.
- It fixes the way TeX sets letters in mathematical formulas so that multi-character identifiers look better.
- It provides various brands of 'boxed mathematics' which appear in Z and Object-Z specifications.

The package is kept in a file `oz.sty` in the directory `/usr/local/lib/tex/localinputs`. This directory should be mentioned in your `TEXINPUTS` shell variable. To use the macros you just begin your LaTeX document with something like:

```
\documentstyle[11pt,oz]{article}
```

2 Schema Boxes

The example below shows a schema on the left and what you need to say to get it on the right.

BirthdayBook
$known : \mathbb{P}\, NAME$
$birthday : NAME \pfun DATE$

$known = \operatorname{dom} birthday$

The command `\ST` (read 'Such That') is the same as the previously used command `\where` which has been kept as an alias for upward compatibility. If you want a schema with no name, just a horizontal rule at the top, use the `schema*` environment instead. You can set various parameters (see Section 7) to change the box style, for example:
$BirthdayBook \mathrel{\widehat=} [\,\ldots\,]$
A generic schema is produced as follows.

Pool $[RESOURCE]$
$owner : RESOURCE \pfun USER$
$free : \mathbb{P}\, RESOURCE$

$(\operatorname{dom} owner) \cup free = RESOURCE$
$(\operatorname{dom} owner) \cap free = ?$

3 Axiomatic definitions

A 'liberal' axiomatic definition is produced as follows.

$limit : \mathbb{N}$

$limit \leq 65536$

A 'generic' axiomatic definition is produced as follows.

$[X, Y]$

$first : X \times Y \to X$

$\forall x : X;\ y : Y \bullet first(x, y) = x$

A 'unique' axiomatic definition is produced as follows.

$\pi : \mathbb{R}$

$\pi = 3.14159265\ldots$

4 Object-Z Class Boxes

Object-Z allows class types to be defined using a box very similar to the schema box previously described. It allows the previously described boxed environments (as well as nested sub-classes) to be placed within a class box. In addition, special names can be used for some of the boxed-environments when they appear within a class box. The following example illustrates a class definition.

Shape

$colour : Colour$

$perim : \mathbb{R}$

$perim > 0$

This class has 2 constants *colour* and *perim*.

$x, y : \mathbb{R}$

INIT

$x = y = 0$

Translate

$\Delta(x, y)$
$dx?, dy? : \mathbb{R}$

$x' = x + dx?$
$y' = y + dy?$

The `classcom` environment hasn't been seen before. It creates a paragraph of text with the same margins as used for schemas and other Z environments. It uses a special font intended for use when placing comments inside classes. A similar environment, `zpar`, uses the same margins but with the normal roman font.

The `\begin{init}` command is an abbreviation for `\begin{schema}{\Init}`. Similarly `\begin{state}` is a more meaningful synonym for `\begin{schema*}`.

You will be given LaTeX warning messages if you try to use a `state` environment outside of a class box or if you try to place an environment such as `syntax` inside a class. You can ask for additional help in these cases using the normal LaTeX `h` or `H` help commands. If you proceed with LaTeXing, the macros will attempt to do the best they can to do what you probably intend, even though you are violating the recommended nesting guidelines.

5 Controlling the Spacing within Equations and Boxes

Most of the special Z symbols are defined in a way that allows TeX to space them out correctly. Sometimes, however, you'll need to give TeX a helping hand if you want it to get the spacing right. For example, to get $map\,f$ you need to type `map\,f`. The `\,` gives you a thin space: if this is omitted, the input `map f` gives $mapf$, because TeX ignores spaces in math mode.

Sometimes it is useful to indent the left margin to emphasis the logical structure of the predicate. The command `\t1` does this by making the corresponding line in the output have one helping of indentation. As things get more nested, you can say `\t2`, `\t3`, and so on. But if you should ever get beyond `t9`, you'll need to use braces around the argument: `\t{10}`, and you'd better look for some way to simplify your specification! These little tab marks might look different to normal tabs but are never the less convenient. They're short, and they don't get longer as the tabbing gets deeper, within reason, so they can be tucked in neatly on the left, well away from the maths. The size of 'helping' you get with `\t` is determined by the `\zedtab` parameter (see Section 7).

If you want a more powerful aligning mechanism than tabbing then you can use the margin stack as shown in the example below. The command `\M` sets the future left margin to the current horizontal position and pushes the old value onto a margin stack. The command `\O` resets the left margin to its previous value (which is popped off the stack).

Test

$x, y : \mathbb{N}$

$$\begin{array}{l} x + 1/x = 0 \Rightarrow y + 1/y = 0 \\ \phantom{x + 1/x = 0 \Rightarrow{}} y = x \end{array}$$

If a schema or other box contains more than one predicate below the line, it often looks better to add a tiny vertical space between them, as in this example:

AddBirthday

$\Delta BirthdayBook$
$n? : NAME$
$d? : DATE$

$n? \notin known$

$birthday' = birthday \cup \{n? \mapsto d?\}$

This is done with the command `\also`, which behaves syntactically like `\ST`. The command `\also` is provided *instead* of the optional argument to `\\` which LaTeX provides in other environments. If larger vertical spacing is required, the commands `\Also` and `\ALSO` may be used (giving 2 and 4 times as much space as `\also` respectively).

Normally, the contents of a schema box are kept on a single page. For large schemas it may be necessary to split the box across pages. You must specify which places are suitable for splitting using one of `\zbreak`, `\Zbreak` or `\ZBREAK`. If no split is performed at this point, a vertical space will be added as if the user had typed `\also`, `\Also`, or `\ALSO` respectively. You can also use the `\znewpage` command to force a page break within a box. (These breaking facilities will hopefully never be needed for schemas, but may become necessary for class specifications.)

6 Other Display Environments

The `zed` environment can be used to set multi-line formulas without an enclosing box: it is useful for given-set declarations, theorems, and the miscellaneous bits of mathematics that don't come in a box:

$$\begin{array}{l} \forall n : \mathbb{N} \bullet \\ \quad n + n \in \mathit{even}. \end{array}$$

The formula `\begin{zed} ... \end{zed}` may be abbreviated to `\[ ... \]`; the `zed` environment is a generalization of the `displaymath` environment of LaTeX, so this redefinition of commands is fairly benign. Notice that the maths is set flush left on the same indentation as schemas and their friends. Here too you can use `\also` for a little extra space between lines.

For algebraic-style proofs, there is the `argue` environment. This is like the `zed` environment, but the separation between lines is increased a little, and page breaks may occur between lines. When the left-hand side is long this style wastes less space than the LaTeX `eqnarray` style. The intended use is for arguments like this:

$$\begin{array}{l} \mathit{rev}(\mathit{append}(\mathit{cons}(x, s), t)) \\ \quad = \mathit{rev}(\mathit{cons}(x, \mathit{append}(s, t))) \\ \quad = \mathit{append}(\mathit{rev}(\mathit{append}(s, t)), \mathit{cons}(x, \mathit{nil})) \\ \quad = \mathit{append}(\mathit{append}(\mathit{rev}(t), \mathit{rev}(s)), \mathit{cons}(x, \mathit{nil})) \quad \text{by hypothesis} \\ \quad = \mathit{append}(\mathit{rev}(t), \mathit{append}(\mathit{rev}(s), \mathit{cons}(x, \mathit{nil}))) \\ \quad = \mathit{append}(\mathit{rev}(t), \mathit{rev}(\mathit{cons}(x, s))). \end{array}$$

Here is the input:

```
\begin{argue}
    rev(append(cons(x,s),t)) \\
\t1     = rev(cons(x,append(s,t))) \\
\t1     = append(rev(append(s,t)),cons(x,nil)) \\
\t1     = append(append(rev(t),rev(s)),cons(x,nil))
                \quad \hbox{by hypothesis} \\
\t1     = append(rev(t),append(rev(s),cons(x,nil))) \\
\t1     = append(rev(t),rev(cons(x,s))).
\end{argue}
```

The example below shows an inference rule (the optional argument to `\derive` gives a side-condition of the rule):

$$\frac{\Gamma \vdash P}{\Gamma \vdash \forall x \bullet P} \quad [\; x \notin \mathit{freevars}(\Gamma) \;]$$

The `syntax` environment is used for making displays like this:

$$\begin{array}{lcll} \mathit{EXPR} & ::= & \mathit{IDENT} & \text{– identifier} \\ & | & \mathit{EXPR}\;\mathit{EXPR} & \text{– application} \\ & | & \lambda\,\mathit{IDENT} \bullet \mathit{EXPR} & \text{– lambda-abstraction.} \end{array}$$

from input like this:

```
\begin{syntax}
  EXPR & \ddef & IDENT                     & identifier \\
       & \bbar & EXPR\;EXPR                & application \\
       & \bbar & \lambda IDENT \dot EXPR & lambda-abstraction.
\end{syntax}
```

This kind of thing is useful when you're describing a language, and it can also be used for data-type definitions as shown below. The optional final column was omitted below by leaving out the third `&`.

$$\begin{array}{lcl} \mathit{TYPE} & ::= & \mathit{givenT}\langle\!\langle \mathit{NAME} \rangle\!\rangle \\ & | & \mathit{powerT}\langle\!\langle \mathit{TYPE} \rangle\!\rangle \\ & | & \mathit{tupleT}\langle\!\langle \mathrm{seq}\ \mathit{TYPE} \rangle\!\rangle \\ & | & \mathit{schemaT}\langle\!\langle \mathit{IDENT} \twoheadrightarrow\!\!\!\!\!\!\mapsto \mathit{TYPE} \rangle\!\rangle \\ & | & \mathit{classT}\langle\!\langle \mathit{IDENT} \twoheadrightarrow\!\!\!\!\!\!\mapsto \mathit{ClassAttr} \rangle\!\rangle \end{array}$$

This can be compared with the layout adopted by the UQ Z editor (version 1).

$$\begin{array}{lcl} \mathit{TYPE} & ::= & \mathit{givenT}\langle\!\langle \mathit{NAME} \rangle\!\rangle \\ & & \quad | \; \mathit{powerT}\langle\!\langle \mathit{TYPE} \rangle\!\rangle \\ & & \quad | \; \mathit{tupleT}\langle\!\langle \mathrm{seq}\ \mathit{TYPE} \rangle\!\rangle \\ & & \quad | \; \mathit{schemaT}\langle\!\langle \mathit{IDENT} \twoheadrightarrow\!\!\!\!\!\!\mapsto \mathit{TYPE} \rangle\!\rangle \\ & & \quad | \; \mathit{classT}\langle\!\langle \mathit{IDENT} \twoheadrightarrow\!\!\!\!\!\!\mapsto \mathit{ClassAttr} \rangle\!\rangle \end{array}$$

The `sidebyside` environment allows a display as shown in the first two columns below to be produced from the text of the third column. Note the use of the `\comment` command.

Schema	
[declarations]	
$a < b$	[pred-1]
$aaaaa < bbbbb$	[pred-2]

This is a paragraph which has the same margins as the standard schemas do.

In fact, this environment was used throughout this note to display the examples beside the required input text. Incidentally, the above example shows that `sidebyside` environments can be nested; so what the author of this note typed to get the above display was:

```
\begin{sidebyside}
        \begin{sidebyside}
```

```
                ...
         \nextside
                ...
         \end{sidebyside}
\nextside
         ...
\end{sidebyside}
```

This resulted in the first two columns being equally spaced and together taking up as much space as the third column. You can have more than 2 columns without nesting by specifying an optional parameter to `sidebyside`. For example, the display below has three equally spaced columns obtained using `\begin{sidebyside}[3]`.

BirthdayBook	*BirthdayBook*	
$known : \mathbb{P}\ NAME$ $birthday : NAME \pfun DATE$	$known : \mathbb{P}\ NAME$ $birthday : NAME \pfun DATE$	Don't get carried away with `sidebyside` like this example does.
$known = \operatorname{dom} birthday$	$known = \operatorname{dom} birthday$	

7 Style Parameters

`\zedindent` The (horizontal) indentation for mathematical text. By default, this is the same as `\leftmargini`, the indentation used for list environments.

`\zedleftsep` The (horizontal) space between the vertical line on the left of schemas, etc., and the maths inside. The default is 1em.

`\zedtab` The unit of indentation used by `\t`. The default is 2em.

`\zedbar` The length of the horizontal bar in the middle of a schema. The default is 8em.

`\leftschemas` A declaration which makes schema names be set flush left. Use it in the document preamble.

`\zedlinethickness` The thickness of the lines that make up schema and class boxes. You can change the thickness with a command such as `\zedlinethickness=0.1pt`. This may be useful if you are creating overhead slides.

0.1pt

0.4pt (The default)

1pt

`\baselinestretch` The spacing for the text part of your document. It doesn't change the spacing within Z environments. It's default value is 1. A command such as `\def\baselinestretch{2}` will make your text double spaced, but not your Z environments.

`\zedbaselinestretch` The spacing for the Z environment part of your document. It's default value is 1.

`\zedsize` The size of the material within the Z part of your document. It doesn't affect the remainder of your document. For example, `\zedsize{\large}` will give you large Z symbols and equations but will not affect the size of the surrounding text.

`\zedcornerheight` The height of 'corners' that can be placed on the right hand side of the top and bottom lines of schema and class boxes. The default is 0em (i.e. no corners).

8 Symbols

Multi-letter identifiers have been changed to look better than they do with vanilla LaTeX: instead of *speci fications*, you get *specifications*. The letters haven't been spread apart, and the ligature *fi* has been used.

Almost all of the mathematical symbols of LaTeX can be used; some have been redefined—usually to fix the spacing so that it is suitable for Z specifications. The commands for obtaining additional symbols are listed below. Sometimes more than one command may produce a symbol you require. You should use the one that seems to be designed for the context you have in mind. This is because the spacing around (and size of) symbols has been chosen for their typical context.

Throughout the lifetime of these macros a number of alternate control sequences for any symbol may have existed. A list of aliases has been set up so that old commands may still be used. It is recommended however that you stick to the recommended command names for symbols as these names may be supported by other tools. Within the table below non-recommended aliases are surrounded by brackets, e.g., `(\power)`. 2

8.1 Special Z Notation

Numbers

N	`\nat`
N_1	`\natone (\nplus)`
Z	`\integer (\num)`
R	`\real`
div mod	`\div \mod`
i^n	`i^n i\expon n`
succ	`\succ`
= ≠	`= \neq`
< ≤ 6	`< \leq \leqslant`
> ≥ ¿	`> \geq \geqslant`
∗ / + −	`* / + -`

Logic

¬	`\lnot`
∧	`\land (\wedge)`
∨	`\lor (\vee)`
∀	`\all (\forall)`
∃	`\exi (\exists)`
$\exists_1$	`\exione`
@	`\nexi (\nexists)`
•	`@ \dot (\spot)`
⇒	`\imp (\implies)`
⇔	`\iff`
true	`\true`
false	`\false`
B	`\bool`

Sets

{ | }	`\{ \cbar \}`
?	`\emptyset`
{ }	`\varemptyset`

$\in$	`\mem (\in)`
$\notin$	`\nem (\nmem \notin)`
P	`\pset (\power)`
F	`\fset (\finset)`
P_1 F_1	`\fsetone \psetone`
$\cup$	`\uni (\union)`
$\cap$	`\int (\inter)`
$\subset$	`\psubs (\subset)`
$\subseteq$	`\subs (\subseteq)`
$\supset$	`\psups (\supset)`
$\supseteq$	`\sups (\supseteq)`
$\times$	`\prod (\cross)`
min *max*	`\min \max`
$\#$	`\#`
$\bigcup$	`\duni (\dunion)`
$\bigcap$	`\dint (\dinter)`
$..$	`\upto`

Relations and Functions

λ μ	`\lambda \mu`	
dom	`\dom`	
ran	`\ran`	
C	`\dres`	
Ȼ	`\dsub (\ndres)`	
B	`\rres`	
Ƀ	`\rsub (\nrres)`	
$\oplus$	`\fovr`	
$\circ$	`\cmp`	
$\circ_9$	`\fcmp (\comp)`	
$(\!\vert$ $\vert\!)$	`\limg \rimg`	
id	`\id`	
R^{-1}	`R^{-1}`	`R\inv`
R^+	`R^+`	`R\tcl`
R^*	`R^*`	`R\rtcl`
R^k	`R^k`	`R\iter k`
iter $0\,R$	`iter \, 0 \, R`	
$\mapsto$	`\map`	
$\leftrightarrow$	`\rel`	
$\rightarrow$	`\tfun (\fun)`	
æ	`\tinj (\inj)`	
$\twoheadrightarrow$	`\tsur (\surj)`	
$\nrightarrow$	`\pfun`	
æ	`\pinj`	
⤀	`\psur (\psurj)`	
⇻	`\ffun`	
æ	`\finj`	
æ→	`\bij`	

Sequences

seq	`\seq`

seq_1	`\seqone`
$\langle\rangle$	`\emptyseq`
$\langle \quad \rangle$	`\lseq \rseq`
head *tail*	`\head \tail`
front *last*	`\front \last`
rev	`\rev`
next	`\next`
in	`\inseq`
$\subseteq$	`\prefix`
suffix	`\suffix`
squash	`\squash`
$\frown$	`\cat`
$\frown/$	`\dcat`
$\fatsemi/$	`\dcmp`
$\oplus/$	`\dovr`
$\upharpoonleft$	`\ires`
$\upharpoonright$	`\sres \filter`
partitions	`\partitions`
disjoint	`\disjoint`

Schemas

Δ	`\Delta`
$\equiv$ Ξ	`\equiv \Xi`
pred	`\pred`
pre	`\pre`
post	`\post`
$\upharpoonright$	`\zproject \project`
$\wedge$	`\zand`
$\vee$	`\zor`
$\fatsemi$	`\zcmp (\semi)`
$\exists$	`\zexi`
$\forall$	`\zall`
$\neg$	`\znot`
$\setminus$	`\zhide (\hide)`
$/$	`\zfor`
$\Rightarrow$	`\zimp`
$\Leftrightarrow$	`\zeq`
$\oplus$	`\zovr`
$\gg$	`\zpipe`
θ	`\theta`
$[\quad \mid \quad]$	`\lsch \zbar \rsch`
..	.

Bags

$[\![\quad]\!]$	`\lbag \rbag`
bag	`\bag`
$[\![]\!]$	`\emptybag`
items	`\items`
count	`\bagcount`
$\uplus$	`\buni`
in	`\inbag`

Definitions and Declarations

::=	`\ddef`
$\vert$	`\bbar`
==	`\defs`
$\widehat{=}$	`\sdef`
,	`\varsdef`
, ; :	`, ; :`
⟪ ⟫	`\lang \rang`

Miscellaneous

[ ]	`[ ]`
( )	`( )`
! ?	`! ?`
let	`\zlet`
where	`\zwhere`
in	`\zin`
⦉ ⦊	`\lblot \rblot`
¡*Bump*	`\intern Bump`
Init	`\Init`
Exit	`\Exit`

8.2 Other Special Notation

Temporal Logic

$\Box$ ⊟	`\always \uptilnow`
	`(\henceforth)`
$\bigcirc$ ⊖	`\atnext \atlast`
$\Diamond$ ⟠	`\eventually \previously`

Proofs

Theorem	`\TH`
Proof	`\PR`
Lemma	`\LE`
$\Box$ $\Box$	`\qed (\ETH) \Qed`
Λ Ξ	`\QED \BLACKQED`
$\vdash$	`\shows (\thrm)`
ﬃ	`\vDash`
$\sqsupseteq$	`\refines`
$\underset{\approx}{\mathrm{A}}$	`\weakrefine`

Object Theory

$\succ$	`\subclass \isa`
%	`\weaksubclass \islikea`
$\prec$	`\supclass`
-	`\weaksupclass`
⤜	`\hasa \instantiates`
⤛	`\instancein`
@ $\sqsubseteq$	`\subtype \subtypeeq`
A $\sqsupseteq$	`\suptype \suptypeeq`

Orders

monotonic	`\mono`
total_order	`\torder`
partial_order	`\porder`

Word Styles

word	`\word{word}`
word	`\keyword{word}`
word	`\boldword{word}`
$\underline{\textit{word}}$	`\underword{word}`
$\underline{\text{word}}$	`\underkeyword{word}`
$\underline{\textbf{word}}$	`\underboldword{word}`
‘`word`’	`\String{word}`
“`word`”	`\STRING{word}`
$a\ \underline{\mathit{rel}}\ b$	`a \infix{rel} b`

8.3 Special Letter Fonts

Greek

α	`\alpha`
β	`\beta`
$\gamma\ \Gamma$	`\gamma \Gamma`
$\delta\ \Delta$	`\delta \Delta`
$\epsilon\ \varepsilon$	`\epsilon \varepsilon`
ζ	`\zeta`
η	`\eta`
$\theta\ \vartheta\ \Theta$	`\theta \vartheta \Theta`
ι	`\iota`
κ –	`\kappa \varkappa`
$\lambda\ \Lambda$	`\lambda \Lambda`
μ	`\mu`
ν	`\nu`
$\xi\ \Xi$	`\xi \Xi`
$\pi\ \varpi\ \Pi$	`\pi \varpi \Pi`
$\rho\ \varrho$	`\rho \varrho`
$\sigma\ \varsigma\ \Sigma$	`\sigma \varsigma \Sigma`
τ	`\tau`
$\upsilon\ \Upsilon$	`\upsilon \Upsilon`
$\phi\ \varphi\ \Phi$	`\phi \varphi \Phi`
χ	`\chi`
$\psi\ \Psi$	`\psi \Psi`
$\omega\ \Omega$	`\omega \Omega`

Caligraphic

$\mathcal{A}$	`\mathcal{A}`
$\mathcal{B}$	`\mathcal{B}`
$\mathcal{C}$	`\mathcal{C}`
$\mathcal{D}$	`\mathcal{D}`
$\mathcal{E}$	`\mathcal{E}`

$\mathcal{F}$	\mathcal{F}
$\mathcal{G}$	\mathcal{G}
$\mathcal{H}$	\mathcal{H}
$\mathcal{I}$	\mathcal{I}
$\mathcal{J}$	\mathcal{J}
$\mathcal{K}$	\mathcal{K}
$\mathcal{L}$	\mathcal{L}
$\mathcal{M}$	\mathcal{M}
$\mathcal{N}$	\mathcal{N}
$\mathcal{O}$	\mathcal{O}
$\mathcal{P}$	\mathcal{P}
$\mathcal{Q}$	\mathcal{Q}
$\mathcal{R}$	\mathcal{R}
$\mathcal{S}$	\mathcal{S}
$\mathcal{T}$	\mathcal{T}
$\mathcal{U}$	\mathcal{U}
$\mathcal{V}$	\mathcal{V}
$\mathcal{W}$	\mathcal{W}
$\mathcal{X}$	\mathcal{X}
$\mathcal{Y}$	\mathcal{Y}
$\mathcal{Z}$	\mathcal{Z}

BlackBoard Bold

A	\mathbb A
B	\mathbb B
C	\mathbb C
D	\mathbb D
E	\mathbb E
F	\mathbb F
G	\mathbb G
H	\mathbb H
I	\mathbb I
J	\mathbb J
K	\mathbb K
L	\mathbb L
M	\mathbb M
N	\mathbb N
O	\mathbb O
P	\mathbb P
Q	\mathbb Q
R	\mathbb R
S	\mathbb S
T	\mathbb T
U	\mathbb U
V	\mathbb V
W	\mathbb W
X	\mathbb X
Y	\mathbb Y
Z	\mathbb Z

Miscellaneous

″	\hslash
˜	\hbar
¨	\backepsilon
g	\eth
i	\beth
j	\gimel
k	\daleth
–	\complement
—	\intercal
$\aleph$	\aleph
∇	\nabla
˜	\hbar
$\imath$	\imath
$\jmath$	\jmath
ℓ	\ell
$\wp$	\wp
$\Re$	\Re
$\Im$	\Im
f	\mho
z	\digamma

8.4 Shapes

$\Box$	\Box
Λ	\square
Ξ	\blacksquare
$\diamond$	\diamond
$\Diamond$	\Diamond
Σ	\lozenge
Υ	\blacklozenge
B	\vartriangleright
C	\vartriangleleft
H	\blacktriangledown
I	\blacktriangleright
J	\blacktriangleleft
M	\vartriangle
N	\blacktriangle
O	\triangledown
$\triangle$	\triangle
$\triangleleft$	\triangleleft
$\triangleright$	\triangleright
$\bigtriangleup$	\bigtriangleup
$\bigtriangledown$	\bigtriangledown
$\clubsuit$	\clubsuit
$\diamondsuit$	\diamondsuit
$\heartsuit$	\heartsuit
$\spadesuit$	\spadesuit
$\ast$	\ast
$\star$	\star
z	\maltese
F	\bigstar

$\bigcirc$	`\bigcirc`
$\circ$	`\circ`
$\bullet$	`\bullet`
Π	`\centerdot`
$\cdot$	`\cdot`
$\cdots$	`\cdots`
$\ldots$	`\ldots`
$\vdots$	`\vdots`
$\ddots$	`\ddots`

Circled Operations

s	`\circledS`
″	`\circledcirc`
˜	`\circledast`
¨	`\circleddash`
r	`\circledR`
©	`\copyright`
$\oplus$	`\fovr`
$\ominus$	`\ominus`
$\otimes$	`\otimes`
$\oslash$	`\oslash`
$\odot$	`\odot`

Boxed operators

Γ	`\boxdot`
∆	`\boxplus`
Θ	`\boxtimes`
ﬁ	`\boxminus`

8.5 Arrow Symbols

Left Arrows

$\leftarrow$	`\leftarrow`	`\gets`
$\Leftarrow$	`\Leftarrow`	
$\hookleftarrow$	`\hookleftarrow`	
$\leftharpoonup$	`\leftharpoonup`	
$\leftharpoondown$	`\leftharpoondown`	
$\longleftarrow$	`\longleftarrow`	
$\Longleftarrow$	`\Longleftarrow`	
ȷ	`\twoheadleftarrow`	
`	`\leftleftarrows`	
œ	`\leftarrowtail`	
W	`\Lleftarrow`	
8	`\nleftarrow`	
:	`\nLeftarrow`	

Right Arrows

$\rightarrow$	`\rightarrow \to`
$\Rightarrow$	`\Rightarrow`
$\mapsto$	`\map`
$\longmapsto$	`\longmapsto`
$\hookrightarrow$	`\hookrightarrow`
$\rightharpoonup$	`\rightharpoonup`
$\rightharpoondown$	`\rightharpoondown`
$\leadsto$	`\leadsto`
$\longrightarrow$	`\longrightarrow`
$\Longrightarrow$	`\Longrightarrow`
ı	`\twoheadrightarrow`
´	`\rightrightarrows`
æ	`\rightarrowtail`
ˊ	`\rightsquigarrow`
V	`\Rrightarrow`
9	`\nrightarrow`
;	`\nRightarrow`

Left Right Arrows

$\leftrightarrow$	`\leftrightarrow`
$\Leftrightarrow$	`\Leftrightarrow`
Ω	`\rightleftharpoons`
$\longleftrightarrow$	`\longleftrightarrow`
$\Longleftrightarrow$	`\Longleftrightarrow`
Ω	`\rightleftharpoons`
ﬀ	`\leftrightharpoons`
ø	`\leftrightarrows`
Æ	`\rightleftarrows`
!	`\leftrightsquigarrow`
¡	`\nLeftrightarrow`
=	`\nleftrightarrow`

Up Down Arrows

$\uparrow$	`\uparrow`
$\Uparrow$	`\Uparrow`
$\downarrow$	`\downarrow`
$\Downarrow$	`\Downarrow`
$\updownarrow$	`\updownarrow`
$\Updownarrow$	`\Updownarrow`
˘	`\upuparrows`

˘	`\downdownarrows`
¯	`\upharpoonright`
˚	`\downharpoonright`
¸	`\upharpoonleft`
ß	`\downharpoonleft`

Miscellaneous

↗	`\nearrow`
↘	`\searrow`
↙	`\swarrow`
↖	`\nwarrow`
Φ	`\circlearrowright`
Ψ	`\circlearrowleft`
Œ	`\Lsh`
Ø	`\Rsh`
”	`\looparrowleft`
#	`\looparrowright`
x	`\curvearrowleft`
y	`\curvearrowright`

8.6 Relations

$\ll$	\ll
$\gg$	\gg
n	\lll \llless
o	\ggg \gggtr
@	\sqsubset
A	\sqsupset
$\sqsubseteq$	\sqsubseteq
$\sqsupseteq$	\sqsupseteq
$\ni$	\owns
$\vdash$	\vdash
ﬂ	\Vdash
ﬄ	\vDash
$\dashv$	\dashv
ﬃ	\Vvdash
$\models$	\models
0	\nvdash
1	\nVdash
2	\nvDash
3	\nVDash
$\perp$	\perp
$\sim$	\sim
$\simeq$	\simeq
h	\eqsim
v	\backsim
w	\backsimeq
s	\thicksim
ø	\nsim
$\approx$	\approx
$\not\approx$	\napprox
t	\thickapprox
u	\approxeq
$\asymp$	\asymp
$\cong$	\cong
Æ	\ncong
$\doteq$	\doteq
+	\Doteq \doteqdot
:	\risingdotseq
;	\fallingdotseq
l	\lessdot
m	\gtrdot
P	\eqcirc

$	`\circeq`
l	`\bumpeq`
m	`\Bumpeq`
,	`\triangleq`
(	`\multimap`
$\propto$	`\propto`
˙	`\varpropto`
.	`\lesssim`
&	`\gtrsim`
/	`\lessapprox`
'	`\gtrapprox`
$\prec$	`\xprec`
$\succ$	`\xsucc`
$\preceq$	`\preceq`
$\succeq$	`\succeq`
-	`\precsim`
%	`\succsim`
w	`\precapprox`
v	`\succapprox`
0	`\eqslantless`
1	`\eqslantgtr`
2	`\curlyeqprec`
3	`\curlyeqsucc`
4	`\preccurlyeq`
¡	`\succcurlyeq`
5	`\leqq`
=	`\geqq`
6	`\leqslant`
¿	`\geqslant`
7	`\lessgtr`
?	`\gtrless`
Q	`\lesseqgtr`
R	`\gtreqless`
S	`\lesseqqgtr`
T	`\gtreqqless`
Γ	`\lvertneqq`
Δ	`\gvertneqq`
Θ	`\nleq`
Λ	`\ngeq`
Ξ	`\nless`
Π	`\ngtr`
Σ	`\nprec`

<table>
<tr><td>Υ</td><td>\nsucc</td></tr>
<tr><td>Φ</td><td>\lneqq</td></tr>
<tr><td>Ψ</td><td>\gneqq</td></tr>
<tr><td>Ω</td><td>\nleqslant</td></tr>
<tr><td>ff</td><td>\ngeqslant</td></tr>
<tr><td>fi</td><td>\lneq</td></tr>
<tr><td>fl</td><td>\gneq</td></tr>
<tr><td>ffi</td><td>\npreceq</td></tr>
<tr><td>ffl</td><td>\nsucceq</td></tr>
<tr><td>ı</td><td>\precnsim</td></tr>
<tr><td>ȷ</td><td>\succnsim</td></tr>
<tr><td>`</td><td>\lnsim</td></tr>
<tr><td>´</td><td>\gnsim</td></tr>
<tr><td>ˇ</td><td>\nleqq</td></tr>
<tr><td>˘</td><td>\ngeqq</td></tr>
<tr><td>¯</td><td>\precneqq</td></tr>
<tr><td>˚</td><td>\succneqq</td></tr>
<tr><td>¸</td><td>\precnapprox</td></tr>
<tr><td>ß</td><td>\succnapprox</td></tr>
<tr><td>æ</td><td>\lnapprox</td></tr>
<tr><td>œ</td><td>\gnapprox</td></tr>
<tr><td>b</td><td>\Subset</td></tr>
<tr><td>c</td><td>\Supset</td></tr>
<tr><td>j</td><td>\subseteqq</td></tr>
<tr><td>k</td><td>\supseteqq</td></tr>
<tr><td>”</td><td>\nsubseteqq</td></tr>
<tr><td>#</td><td>\nsupseteqq</td></tr>
<tr><td>$</td><td>\subsetneqq</td></tr>
<tr><td>%</td><td>\supsetneqq</td></tr>
<tr><td>(</td><td>\subsetneq</td></tr>
<tr><td>)</td><td>\supsetneq</td></tr>
<tr><td>*</td><td>\nsubseteq</td></tr>
<tr><td>+</td><td>\nsupseteq</td></tr>
<tr><td>D</td><td>\trianglerighteq</td></tr>
<tr><td>E</td><td>\trianglelefteq</td></tr>
<tr><td>4</td><td>\ntrianglerighteq</td></tr>
<tr><td>5</td><td>\ntrianglelefteq</td></tr>
<tr><td>6</td><td>\ntriangleleft</td></tr>
<tr><td>7</td><td>\ntriangleright</td></tr>
<tr><td>G</td><td>\between</td></tr>
<tr><td>|</td><td>| \vert \mid</td></tr>
<tr><td>‖</td><td>\| \Vert \parallel</td></tr>
</table>

$\vert\vert\vert$	`\interleave`
p	`\shortmid`
q	`\shortparallel`
pq	`\shortinterleave`
,	`\nparallel`
-	`\nmid`
.	`\nshortmid`
/	`\nshortparallel`

8.7 Binary Operations

f	`\curlywedge`
g	`\curlyvee`
Y	`\veebar`
Z	`\barwedge`
[	`\doublebarwedge`
$\pm$	`\pm`
$\mp$	`\mp`
$\times$	`\times`
n	`\ltimes`
o	`\rtimes`
h	`\leftthreetimes`
i	`\rightthreetimes`
¿	`\divideontimes`
$\div$	`\divides`
$\uplus$	`\uplus`
$\sqcap$	`\sqcap`
$\sqcup$	`\sqcup`
d	`\Cup \doublecup`
e	`\Cap \doublecap`
$\backslash$	`\backslash`
$\backslash$	`\setminus`
r	`\smallsetminus`
$\wr$	`\wr`
$\lhd$	`\lhd`
$\rhd$	`\rhd`
$\unlhd$	`\unlhd`
$\unrhd$	`\unrhd`
¯	`\restriction`
$\amalg$	`\amalg`

$\top$	`\top`
$\bot$	`\bot`
‘	`\smallsmile`
a	`\smallfrown`
$\smile$	`\smile`
$\frown$	`\frown`
t	`\pitchfork`
u	`\dotplus`
$\Join$	`\Join`
$\bowtie$	`\bowtie`

8.8 Miscellaneous Symbols

†	`\dagger`
‡	`\ddagger`
§	`\sectionsymbol`
¶	`\P`
“	`\angle`
]	`\measuredangle`
ˆ	`\sphericalangle`
$\prime$	`\prime`
8	`\backprime`
$\surd$	`\surd`
$\smallint$	`\smallint`
$\flat$	`\flat`
$\natural$	`\natural`
$\sharp$	`\sharp`
∂	`\partial`
∞	`\infty`
U	`\yen`
)	`\therefore`
*	`\because`
X	`\checkmark`

8.9 Variable-sized Symbols

These symbols come in two sizes which do not vary with the point size of your font. The big size can be obtained by preceding the symbol command with the command `\displaystyle`.

$\textstyle\sum$ $\displaystyle\sum$	`\sum`
$\textstyle\prod$ $\displaystyle\prod$	`\product`
$\textstyle\coprod$ $\displaystyle\coprod$	`\coprod`
$\textstyle\int$ $\displaystyle\int$	`\integral`
$\textstyle\oint$ $\displaystyle\oint$	`\oint`

$\cap \bigcap$	`\bigcap`
$\cup \bigcup$	`\bigcup`
$\sqcup \bigsqcup$	`\bigsqcup`
$\vee \bigvee$	`\bigvee`
$\wedge \bigwedge$	`\bigwedge`
$\odot \bigodot$	`\bigodot`
$\otimes \bigotimes$	`\bigotimes`
$\oplus \bigoplus$	`\bigoplus`
$\uplus \biguplus$	`\biguplus`

8.10 Delimiters

These symbols can be made large to surround large formula. E.g.,

$$\left\lfloor \sum_{i=1}^{n} x^i \right\rfloor$$

was generated using

```
\left\lfloor...\right\rfloor
```

( )	`( )`
$\{ \}$	`\{ \}`
$\lfloor \rfloor$	`\lfloor \rfloor`
$\lceil \rceil$	`\lceil \rceil`
$\langle \rangle$	`\langle \rangle`
p q	`\ulcorner \urcorner`
x y	`\llcorner \lrcorner`
$\uparrow$	`\uparrow`
$\downarrow$	`\downarrow`
$\updownarrow$	`\updownarrow`
$\Uparrow$	`\Uparrow`
$\Downarrow$	`\Downarrow`
$\Updownarrow$	`\Updownarrow`

8.11 Math Accents

$\hat{a}$	`\hat{a}`
$\widehat{a}$	`\widehat{a}`
$\widehat{aa}$	`\widehat{aa}`
$\widehat{aaa}$	`\widehat{aaa}`
$\tilde{a}$	`\tilde{a}`
$\widetilde{a}$	`\widetilde{a}`
$\widetilde{aa}$	`\widetilde{aa}`

$\widetilde{aaa}$	`\widetilde{aaa}`
$\check{a}$	`\check{a}`
$\breve{a}$	`\breve{a}`
$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`
$\bar{a}$	`\bar{a}`
$\vec{a}$	`\vec{a}`
$\dot{a}$	`\dotaccent{a}`
$\ddot{a}$	`\ddot{a}`

8.12 Size Commands

μ $\scriptstyle\mu$ $\scriptscriptstyle\mu$	`\mu \zsmall\mu \zSmall\mu`
μ $\large\mu$ $\Large\mu$	`\zbig\mu \zBig\mu \zBIG\mu`