Using bardjuniorsem.sty Style for

Expository Writing for the Mathematics Junior Seminar

Ethan Bloch

February 25, 2013

Contents

1 Introduction **3**

2 Document Format **3**

3 Sections **3**

4 Theorems, Definitions, Proofs and the Like **4**

5 Some Standard TeX Mistakes **6**

6 Common Mathematical Commands **7**

7 Comments **7**

8 Typesetting Simulated Computer Code **8**

9 Title Page **10**

10 Table of Contents **10**

11 Abstract **10**

12 Figures **11**

13 Bibliography **13**

Sample Senior Project Outline **15**

1 Introduction

This manual explains how to use the style file bardjuniorsem.sty. This style file is designed for expository writing projects, and takes care of a number of formatting issues such as the title page, abstract and correct margins; it also includes formatting for theorems, definitions and the like, as well as a few miscellaneous items. It is assumed that you already know the basics of LaTeX. Only the specific commands defined in this style file are discussed here.

The file bardjuniorsem.sty, as well as the associated template bardjuniorsem_template.tex, can be downloaded from the website
`http://math.bard.edu/bloch/tex.shtml`.

If you find any errors, or have any suggestions for improvement, please send an email to `bloch@bard.edu`.

2 Document Format

The basic form of the document is

```
\documentclass[11pt, oneside, reqno]{article}
\usepackage{amssymb, amsthm, amsmath, amsfonts}
\usepackage{bardjuniorsem}
\usepackage{graphics}
\usepackage{amsrefs}

\begin{document}

<Text of document>

\end{document}
```

3 Sections

The format for section headings is

```
\section{<Section name>}\label{<Label>}
```

Do not write the section number as part of the name, since the number is inserted automatically. To refer to the section, write

```
Section~\ref{<Label>}
```

4 Theorems, Definitions, Proofs and the Like

The format for theorems, definitions, proofs, etc. are shown below, using the particular case of theorems as an example. The analogous commands for lemmas, definitions, etc. are given in the table on the following page.

1.

Theorem 4.1. *This theorem is automatically numbered.*

```
\thm\label{thmA}
This theorem is automatically numbered.
\ethm
```

2.

Theorem 4.2 Smith's Theorem. *This theorem is automatically numbered, and it also has a name.*

```
\thm[Smith's Theorem]\label{thmB}
This theorem is automatically numbered, and it also has a name.
\ethm
```

3.

This is Theorem 4.1; it is automatically numbered, so you use `\ref` with the internal label.

```
This is Theorem~\ref{thmA}; it is automatically numbered,
so you use \verb@\ref@ with the internal label.
```

4.

Proof. How could this theorem not be true? □

```
\demo
How could this theorem not be true?
\edemo
```

5.

Proof of Theorem 4.2. Obviously, this theorem is true. □

```
\demoname{Proof of Theorem~\ref{thmB}}
Obviously, this theorem is true.
\edemoname
```

6.

If you want to number an equation, you write it as follows, without `\[ ... \]` or `$$ ... $$`.

$$A + B = C. \tag{4.1}$$

```
\begin{equation}\label{eqD}
A + B = C.
\end{equation}
```

7.

This is Equation 4.1; it is automatically numbered, so you use `\ref` with the internal label.

Symbol	**Command**
Definition	`\defn\label{<Label>} ... \edefn`
Theorem	`\thm\label{<Label>} ... \ethm`
Lemma	`\lem\label{<Label>} ... \elem`
Corollary	`\coro\label{<Label>} ... \ecoro`
Proposition	`\prop\label{<Label>} ... \eprop`
Conjecture	`\conj\label{<Label>} ... \econj`
Claim	`\clm\label{<Label>} ... \eclm`
Example	`\expl\label{<Label>} ... \eexpl`
Remark	`\remk\label{<Label>} ... \eremk`
Algorithm	`\alg\label{<Label>} ... \ealg`
Exercise	`\exer\label{<Label>} ... \eexer`
Problem	`\prob\label{<Label>} ... \eprob`

5 Some Standard TeX Mistakes

- To write a set such as
$$\{x \in \mathbb{R} \mid x^2 < \pi\}$$
we write

```
\{x \in \mathbb{R} \mid x^2 < \pi\}.
```

Observe that the curly brackets are written `\{` and `\}`, and that the vertical line is written `\mid`, NOT `|` (which would produce the wrong spacing).

- To write an inner product such as
$$\langle x, y \rangle$$
we write

```
\langle x, y \rangle.
```

Observe that the brackets are written `\langle` and `\rangle`, NOT `<` and `>` (which would produce the wrong spacing).

- To write a function such as
$$f \colon A \longrightarrow B$$
we write

```
f\colon A \to B.
```

Observe that the colon is written `\colon`, NOT `:` (which would produce the wrong spacing).

- DO NOT USE `\\` when writing text. It is the wrong way to start new lines and paragraphs, and will often mess things up. Use `\\` only inside mathematical constructions such as `align`, `matrix`, etc.

6 Common Mathematical Commands

Symbol	Command
$\{x \in X \mid \text{blah}\}$	`\{x \in X \mid \text{blah}\}`
$\mathbb{N}$	`\nn`
$\mathbb{Z}$	`\zz`
$\mathbb{Q}$	`\qqq`
$\mathbb{R}$	`\rr`
$\mathbb{R}^n$	`\rrr{n}`
$A \subseteq B$	`A \subseteq B`
$A \subsetneqq B$	`A \subsetneqq B`
$f\colon A \longrightarrow B$	`\func fAB`
$g \circ f$	`g \rc f`
$\bigcup_{i \in I} A_i$	`\bigcup_{i \in I} A_i`
$\bigcap_{i \in I} A_i$	`\bigcap_{i \in I} A_i`
$\prod_{i=1}^{n} A_i$	`\prod_{i=1}^n A_i`
$\mathcal{B}$	`\mathcal{B}`
$\mathbf{B}$	`\mathbf{B}`

7 Comments

To make a comment to yourself in your .tex document that is ignored when the document is compiled, use the `%` character. Any line in the .tex document that starts with `%` is ignored; if a `%` in inserted in the middle of a line, all the text to the right of the `%` symbol is ignored.

Unfortunately, the standard LaTeX program does not give a method for commenting out long sections of the .tex file, other than using lots of `%` symbols. There is a nice LaTeX package called verbatim.sty that solves this problem. This package can be downloaded from
http://math.bard.edu/bloch/bardtex.htm. Place the package in the same folder in your computer as your .tex file. To use this package, you must first insert

```
\usepackage{verbatim}
```

into your file prior to `\begin{document}`.

Then, simply type `\begin{comment}` before the text that you want commented out, and `\end{comment}` after the text.

8 Typesetting Simulated Computer Code

The simplest way to write a small amount of simulated computer code is to use the `verbatim` environment. *Everything* within this environment is typeset in typewriter font exactly as it appears in the .tex file—obeying spaces and line breaks as in the .tex file, and not recognizing any special symbols.

To obtain

```
This is     verbatim text, where $ signs are
just $ signs, and commands such as \huge are ignored.
```

we simply write

```
\begin{verbatim}
This is     verbatim text, where $ signs are
just $ signs, and commands such as \huge are ingored.
\end{verbatim}
```

For arbitrarily long verbatim text, the LaTeX package called verbatim.sty must be used. This package can be downloaded from http://math.bard.edu/bloch/bardtex.htm. Place the package in the same folder in your computer as your .tex file. To use this package, you must first insert

```
\usepackage{verbatim}
```

into your file prior to `\begin{document}`.

Then, use the `verbatim` environment as above.

There is another package, called moreverb.sty, that contains some very useful variants of the `verbatim` environment; these environments can be used to make typesetting simulated computer code easier. This package can be downloaded from http://math.bard.edu/bloch/bardtex.htm. To use this package, you must first insert

```
\usepackage{moreverb}
```

into your file prior to `\begin{document}`.

The `verbatimtab` environment from the moreverb.sty package allows tab characters to be expanded to a given number of spaces (without this package, LaTeX considers tab characters to be single spaces. In the `verbatimtab` environment, tab characters are by default set to equal eight space characters. In the following examples, the ▷ character denotes a tab character.

To obtain

```
This        is        verbatimtab        environment.
```

we write

```
\begin{verbatimtab}
This▷is▷verbatimtab▷environment.
\end{verbatimtab}
```

An optional argument after the defintion of `verbatimtab` environment can be used to set tab characters to any desired number of spaces.

To obtain

```
This    is    verbatimtab    environment.
```

we write

```
\begin{verbatimtab}[4]
This▷is▷verbatimtab▷environment.
\end{verbatimtab}
```

The `verbatimcmd` environment from the moreverb.sty package is generally like the `verbatim` environment, except that in the `verbatimcmd` environment any LaTeX command that start with a `\` is obeyed, and and `{ }` is respected. Hence, in the `verbatimcmd` environment styles such as bold and italics can be used. Mathematics symbols can also be used in the `verbatimcmd` environment, though only those that start with the `\` symbol; therefore `$a^{x+y}$` would not be recognized (because neither `$` nor `^` starts with the `\` symbol), but it could be replaced with the less commonly used `\(a\sp{x+y}\)`, where `\(  \)` are another way of starting and ending in-line math mode, and `\sp` means superscript (we can also use `\[  \]` as another way of starting and ending display math mode, and `\sb` is used for subscript).

To obtain

```
This        is a^{x+y}        in verbatimcmd.
```

we write

```
\begin{verbatimcmd}
This        is \(a\sp{x+y}\)        in {\bf verbatimcmd}.
\end{verbatimcmd}
```

The `listing` environment from the moreverb.sty package is like the `verbatim` environment, except that it numbers lines. The `listing` environment has two parameters, the first (in square brackets) specifies the step between numbered lines (a value of 1 makes every line numbered), and the second parameter (in curly brackets) specifies the number of the first line. The `listing` environment handles tab characters similarly to the `verbatimtab` environment, though tab characters are unalterably set to equal eight space characters.

To obtain

```
     This is the environment that
8    numbers lines.  In this example,
     every other line is numbered,
10   and the first line is set at line 7.
```

we write

```
\begin{listing}[2]{7}
This is the environment that
numbers lines.  In this example,
every other line is numbered,
and the first line is set at line 7.
\end{listing}
```

The `listingcont` environment from the moreverb.sty package is like the `listing` environment, except that it does not have any parameters, and it continues numbering lines where the previous `listing` or `listingcont` environment left off. The `listing*` and `listingcont*` environments from the moreverb.sty package are very much like the `listing` and `listingcont` environments respectively, except for two differences: they write ␣ for each blank space in the original text, and they do not handle tab characters.

9 Title Page

When you are ready to put your project into final form, add the title page command as follows, immediately after `\begin{document}`. Make sure you put in your own title, name, and the month and year.

```
\titlepgjrsem{<Project title>}{<Your name>}{<Month>}{<Year>}
```

10 Table of Contents

When you are ready to put your project into final form, insert

```
\tableofcontents
\newpage
```

into your text immediately after the title page.

11 Abstract

To put an abstract in your project, insert

```
\abstr
```

after the titlepage and before the table of contents, and then put in the text of your dedication. The appropriate heading, pagebreaks and the like will be done automatically.

12 Figures

The standard way to insert graphics into LaTeX documents is to use either the graphics.sty or the graphicx.sty package. The two packages are essentially the same, but use slightly different formats for a few commands; use whichever you prefer. These packages are often bundled with LaTeX implementations. The manual for these two packages, called grfguide.pdf, can be found on the web; it is a very useful guide to inserting graphics into LaTeX documents, and to using color.

To use either of these packages, you must first insert

```
\usepackage{graphics}
```

or

```
\usepackage{graphicx}
```

into your file prior to `\begin{document}`.

If you want to import a computer generated figure into the .tex file, the figure needs to be in a format that TeX can work with. For most implementations of TeX, the best format to use is EPS (encapsulated postscript, ending in .eps). For TeXShop, TeXWorks and any other implementation that uses PDF files instead of DVI files, the correct format for graphics is PDF. It is also possible to use JPEG files in TeXShop, and perhaps other implementations. The method of inserting these different types of files is the same; the format of the graphics file simply has to work with your implementation of TeX.

To insert an .eps file (or a .pdf file in TeXShop), use the format

```
\begin{figure}[ht]
\centering
\includegraphics{<Path+filename>}
\caption{<Caption>}
\label{<Figure Label>}
\end{figure}
```

In Windows, the path+filename has the form

```
c:/mysubdir/mypic.eps
```

Notice that we use / instead of \ in TeX. In a Mac or Unix the path+filename has the form

```
/Users/username/myfolder/mypic.pdf
```

The caption can be left blank if you do not wish to insert one (though you still need to write {} if you leave the caption blank). You should not write the figure number in the caption, since that is done automatically by LaTeX. To refer to the figure elsewhere in the text, use the format

```
Figure~\ref{<Figure Label>}
```

There are two common ways to create mathematical graphics: via mathematical software such as Mathematica, Maple or Sage, or vector drawing programs such as Adobe Illustrator or Inkscape. Both these types of programs store graphics in their own proprietary formats, but when you are done making the graphics, they can be exported to EPS or PDF formats. Do not use bitmap graphics programs such as Adobe Photoshop; vector graphics programs are much better suited to mathematical drawing (and indeed are based upon mathematical ideas such as splines). A nice program for creating simple graphics is GeoGebra, which is not as powerful as Illustrator or Inkscape, but it has some nice features that are useful for demostrating various geometrical ideas.

One tricky matter when creating graphics is to have the right amount of white space surrounding the graphic, which means a little bit of white space, but not too much. EPS files have a "bounding box" that determines how much white space is inserted; the bounding box is usually created automatically with as little white space as possible, though that can be modified in vector graphics programs by inserting an invisible rectangle of the right size around the graphic. For PDF files, which are often created with the graphic on an entire blank page, the white space can be cropped using an appropriate program (such as Preview on the Mac).

Here are some important formatting points to note when you use figures in a mathematics text:

- Each figure must be referred to by its figure number somewhere in the text. If a figure is not referred to, it will not necessarily be read.
- Each figure should be inserted **after** it is first referred to.
- Captions should be left blank, or should be minimal. Any explanation of a figure should be done in the text, not in the caption.

13 Bibliography

The best way to make a bibiography in LaTeX is to use the package called amsrefs.sty that solves this problem. This package, and a manual for the package, can be downloaded from
http://math.bard.edu/bloch/bardtex.htm. Place the package in the same folder in your computer as your .tex file. To use this package, you must first insert

```
\usepackage{amsrefs}
```

into your file prior to `\begin{document}`.

The bibliography is put at the very end of the document, and looks like

```
\newpage
\begin{bibdiv}
\begin{biblist}[\normalsize]

\addcontentsline{toc}{section}{Bibliography}
\markboth{Bibliography}{Bibliography}

\bib{G-K-P}{book}{
author = {Graham, Ronald},
author = {Knuth, Donald},
author = {Patashnik, Oren},
title = {Concrete Mathematics},
edition = {2},
publisher = {Addison-Wesley},
address = {Reading, MA},
date = {1994}
}

\bib{BA1}{article}{
author = {Banchoff, Thomas},
title = {Critical points and curvature for embedded polyhedra},
journal = {J. Diff. Geom.},
volume = {1},
date = {1967},
pages = {245--256}
}

\bib{SLOA}{report}{
author = {Sloane, Neal J. A.},
title = {The On-Line Encyclopedia of Integer Sequences},
```

```
    eprint = {http://www.research.att.com/~njas/sequences},
    }

    \bib{WINT}{article}{
    author = {Wintgen, P.},
    title = {Normal cycle and integral curvature for polyhedra in
    Riemannian manifolds},
    book = {
      series = {Colloq. Math. Soc. J\'anos Bolyai},
      volume = {31},
      title = {Differential Geometry},
      editor = {So\'os, Gy.},
      editor = {Szenthe, J.},
      publisher = {North-Holland},
      address = {Amsterdam},
      date = {1982}
    },
    pages = {805--816}
    }

\end{biblist}
\end{bibdiv}
```

Each new bibiographic item starts with the

```
\bib{<Label>}{<Type of Item>}{
```

where the Label is the label for the bibliographic item, and the Type of Item is one of `article`, `book`, `misc`, `report`, `thesis`. Some of the fields, such as `author` and `title` are necessary in every bibliographic entry, whereas others such as `series` and `editor` are optional. Note that if there are multiple authors, each one is listed separately, always with last name first. The order of the fields inside each bibliographic entry does not matter. Do not forget the commas between the fields. Also, unlike regular text, in the bibliographic entries TeX is not bothered by a period that is not at the end of a sentence, and you do not need to do anything about such periods.

To refer to an item in the bibliography, use the formats

```
\cite{<Label>} or \cite{<Label>}*{<Location>}
```

where `Location` is something such as "Chapter 3," or "Theorem 1.2.3."

Sample Project Outline

```
\documentclass[11pt, oneside, reqno]{book}
\usepackage{amssymb, amsthm, amsmath, amsfonts}
\usepackage{bardjuniorsem}
\usepackage{graphics}
\usepackage{amsrefs}

\begin{document}

\titlepgjrsem{<Title of Project>}{<Your Name>}
    {<Month>}{<Year>}

\abstr

<Text of abstract>

\tableofcontents
\newpage

\section{<Title of First Section>}
\label{secA1}

<Text>

\section{<Title of Second Section>}
\label{secAB}

<Text>

\begin{figure}[ht]
\centering
\includegraphics{<Path+filename>}
\caption{<Caption>}
\label{<Figure Label>}
\end{figure}

\newpage
\begin{bibdiv}
\begin{biblist}[\normalsize]
```

```
\addcontentsline{toc}{section}{Bibliography}
\markboth{Bibliography}{Bibliography}

\bib{<Label>}{book}{
author = {<Last Name, First Name>},
title = {<Title>},
publisher = {<Publisher>},
address = {<City>},
date = {<Year>}
}

\bib{<Label>}{article}{
author = {<Last Name, First Name>},
title = {<Title>},
journal = {<Journal Name>},
volume = {<Volume Number>},
date = {<Year>}
pages = {<Starting Page--Ending Page>}
}

\end{biblist}
\end{bibdiv}

\end{document}
```