# CODING STYLE

## Contents

## 1. List of style comments

These will be changed over time, but having some here now will hopefully encourage a consistent LaTeX style. We will call "code[1]" the contents of the source files.

(1) Keep all lines in all tex files to at most 80 characters.
(2) Do not use indentation in the tex file. Use syntax highlighting in your editor, instead of indentation, to visualize environments, etc.
(3) Use

```
\medskip\noindent
```

to start a new paragraph, and use

```
\noindent
```

to start a new paragraph just after an environment.
(4) Do not break the code for mathematical formulas across lines if possible. If the complete code complete with enclosing dollar signs does not fit on the line, then start with the first dollar sign on the first character of the next line. If it still does not fit, find a mathematically reasonable spot to break the code.
(5) Displayed math equations should be coded as follows

```
$$
...
...
$$
```

In other words, start with a double dollar sign on a line by itself and end similarly.
(6) *Do not use any macros.* Rationale: This makes it easier to read the tex file, and start editing an arbitrary part without having to learn innumerable macros. And it doesn't make it harder or more timeconsuming to write. Of course the disadvantage is that the same mathematical object may be TeXed differently in different places in the text, but this should be easy to spot.

---

This is a chapter of the Stacks Project, version 714e994, compiled on Oct 28, 2014.

[1] It is all Knuth's fault. See [Knu79].

(7) The theorem environments we use are: “theorem”, “proposition”, “lemma” (plain), “definition”, “example”, “exercise”, “situation” (definition), “remark”, “remarks” (remark). Of course there is also a “proof” environment.

(8) An environment “foo” should be coded as follows

```
\begin{foo}
...
...
\end{foo}
```

similarly to the way displayed equations are coded.

(9) Instead of a “corollary”, just use “lemma” environment since likely the result will be used to prove the next bigger theorem anyway.

(10) Directly following each lemma, proposition, or theorem is the proof of said lemma, proposition, or theorem. No nested proofs please.

(11) The files preamble.tex, chapters.tex and fdl.tex are special tex files. Apart from these, each tex file has the following structure

```
\input{preamble}
\begin{document}
\title{Title}
\maketitle
\tableofcontents
...
...
\input{chapters}
\bibliography{my}
\bibliographystyle{amsalpha}
\end{document}
```

(12) Try to add labels to lemmas, propositions, theorems, and even remarks, exercise, and other environments. If labelling a lemma use something like

```
\begin{lemma}
\label{lemma-bar}
...
\end{lemma}
```

Similarly for all other environments. In other words, the label of a environment named “foo” starts with “foo-”. In addition to this please make all labels consist only of lower case letters, digits, and the symbol “-”.

(13) Never refer to “the lemma above” (or proposition, etc). Instead use:

```
Lemma \ref{lemma-bar} above
```

This means that later moving lemmas around is basically harmless.

(14) Cross-file referencing. To reference a lemma labeled “lemma-bar” in the file foo.tex which has title “Foo”, please use the following code

```
Foo, Lemma \ref{foo-lemma-bar}
```

If this does not work, then take a look at the file preamble.tex to find the correct expression to use. This will produce the “Foo, Lemma <link>” in the output file so it will be clear that the link points out of the file.

(15) If at all possible avoid forward references in proof environments. (It should be possible to write an automated test for this.)

(16) Do not start any sentence with a mathematical symbol.

(17) Do not have a sentence of the type "This follows from the following" just before a lemma, proposition, or theorem. Every sentence ends with a period.
(18) State all hypotheses in each lemma, proposition, theorem. This makes it easier for readers to see if a given lemma, proposition, or theorem applies to their particular problem.
(19) Keep proofs short; less than 1 page in pdf or dvi. You can always achieve this by splitting out the proof in lemmas etc.
(20) In a defining property foobar use

```
{\it foobar}
```

in the code inside the definition environment. Similarly if the definition occurs in the text of the document. This will make it easier for the reader to see what it is that is being defined.
(21) Put any definition that will be used outside the section it is in, in its own definition environment. Temporary definitions may be made in the text. A tricky case is that of mathematical constructions (which are often definitions involving interrelated lemmas). Maybe a good solution is to have them in their own short section so users can refer to the section instead of a definition.
(22) Do not number equations unless they are actually being referenced somewhere in the text. We can always add labels later.
(23) In statements of lemmas, propositions and theorems and in proofs keep the sentences short. For example, instead of "Let $R$ be a ring and let $M$ be an $R$-module." write "Let $R$ be a ring. Let $M$ be an $R$-module.". Rationale: This makes it easier to parse the trickier parts of proofs and statements.
(24) Use the

```
\section
```

command to make sections, but try to avoid using subsections and subsubsections.
(25) Avoid using complicated latex constructions.

## 2. Other chapters

Preliminaries

(1) Introduction
(2) Conventions
(3) Set Theory
(4) Categories
(5) Topology
(6) Sheaves on Spaces
(7) Sites and Sheaves
(8) Stacks
(9) Fields
(10) Commutative Algebra
(11) Brauer Groups
(12) Homological Algebra
(13) Derived Categories
(14) Simplicial Methods
(15) More on Algebra
(16) Smoothing Ring Maps
(17) Sheaves of Modules
(18) Modules on Sites
(19) Injectives
(20) Cohomology of Sheaves
(21) Cohomology on Sites
(22) Differential Graded Algebra
(23) Divided Power Algebra
(24) Hypercoverings

Schemes

(25) Schemes
(26) Constructions of Schemes
(27) Properties of Schemes
(28) Morphisms of Schemes

## References

[Knu79] Donald Ervin Knuth, *Tau Epsilon Chi, a system for technical text*, American Mathematical Society, Providence, R.I., 1979, Revised version of Stanford Computer Science report number STAN-CS-78-675.