COSC 91/191, Spring 2019
Lecture 8
April 10, 2019

Scribes: Almas Abdibayev and Christina Lu

1 General notes

You should be using MathTime Professional 2 for all your math typesetting needs, as the alternatives are quite ugly. Use the command `\input{mtpro2}` to activate it.

Most of today's lecture references Higham's *Handbook of Writing for the Mathematical Sciences*.

2 Theorems, lemmas, and corollaries

What is a theorem? To answer this question, you must first understand the difference between a theorem, lemma, and corollary. A corollary is something that follows easily from either a lemma or a theorem. A lemma is a "helping result." And on the main question here, mathematicians and computer scientists disagree. In math, a theorem is a major result of independent interest; in computer science, a theorem only has to be some significant result, or a result that closes out a section or chapter. In all three cases, you must signal the end of the proof with a QED box.

How do you state a theorem? Your theorem must be as self-contained as possible and be unambiguous about when your result applies. You may omit notation for a lemma if it is clear from the previous text. For example, suppose that the text has already established that A is an invertible matrix.

Lemma: *The matrix A has full rank.*

Theorem: *If A is an invertible matrix, then A has full rank.*

Of course, you should avoid notation entirely if possible.

Every invertible matrix has full rank.

How do you prove a theorem? Before proving a theorem, you must first state the technique you will be using (contradiction, induction, construction, etc.). For long proofs, you should give the reader a roadmap in the form of a topic paragraph, followed by guideposts during the proof, in order to orient them. Lemmas are helpful to factor out pieces.

Topic paragraph: We will show X and Y, which will imply Z. Topic sentence: We start by showing X. Topic sentence: Now we show Y. Topic sentence: Having shown X and Y, now we prove that they imply Z. Concluding paragraph or sentence: Thus, we see that X and Y, and hence Z.

3 Definitions

In order to make it distinct to the reader, you should use different fonts or colors when defining a term. You should also place definitions as close to their first usage as possible. When writing a definition, you should omit *and only if*.

A directed graph is *strongly connected* if there exist paths from each vertex to all other vertices.

4 Notation

According to Charles Leiserson: Using the right notation really helps. You'll feel it if you get it wrong.

Use math mode when using notation. For constant functions such as sin and min where they do not stand in for any other function, do not use italics but rather typeset them in Times New Roman. LaTeXprovides macros such as `\sin` and `\min` (also `\log`, `\lg`, and `\ln`). Other miscellany: don't use lowercase letters with uppercase subscripts. Don't use weird fonts just because you can; Tom suggests avoiding calligraphic fonts. Don't let the font be the only difference between symbols.

5 Writing math

Math is always part of a sentence. Treat it as such in your punctuation: `\[ F = ma \ . \]` When math is set off in a display and ends a sentence, add a space before the punctuation that ends the sentence; if the math is inline, no space is necessary. Tom also prefers `\[ \]` over `$$ $$` to delimit math displays.

Math operators are verbs, but don't use them as main verbs. In some cases, it may be necessary to use a *we* crutch.

Always write out *for all* and *there exists*; do not use the symbols $\forall$ and $\exists$ unless you are writing quantified logic.

Always start your sentences with words instead of math. If need be, describe what it is you are using. If you have a bullet-point list however, it is permissible to use math notation at the start of an item.

6 Miscellaneous

In this section, we will look at useful pieces of advice that don't share much thematically.

Though it is rare that we, as computer scientists, would use complex numbers, it is still important to be mindful of the following:

- For signal processing: don't use i as an index, since i denotes $\sqrt{-1}$.
- For electrical engineering: don't use j as an index, for a similar reason as above.

When defining a set, Tom prefers to use a colon (:) to denote *such that*, as opposed to a pipe (|), but it's not a hard and fast rule.

There are three different epsilons in LaTeX that you should be able to distinguish: `\epsilon` (ϵ), `\varepsilon` (ε), and `\in` ($\in$; for inclusion in a set)

Don't write `$k^{th}$`, because the *th* should not be raised. Use `$k$th` (kth) instead.

When dealing with multiple parentheses, opinions vary on whether you need to be mindful of relative heights between outer and inner parentheses. Tom and his coauthors decided not to deal with this in CLRS. You can certainly use `\big(`, `\Big(`, `\bigg(`, `\Bigg(` to control the heights manually. Within a formula, you need to take care of heights being proportional to each other:

$$(\frac{x}{y}) \text{ vs } \left(\frac{x}{y}\right)$$

To deal with the proper height alignment, you should use `\left` and `\right` before each respective parenthesis. When you need to match the height of only one side of either expression following `\left,\right` (e.g., `\left{`) you can use `\right.` or `\left.` (notice the dot which is an invisible delimiter) respectively.

Often in math we assume things. Be sure to add *that* after *assume*:

- ✓ assume that x is even
- ✗ assume x is even

Try not to put multiple equations into one line of a math display. Example:

$$\text{✗} \quad x + y = 5 \quad \text{and} \quad x - y = 3$$

$$\text{✓} \quad \begin{aligned} x + y &= 5\,, \\ x - y &= 3\,. \end{aligned}$$

There are multiple ways to break a line in a long equation. Opinions vary but both of these options work:

$$\textbf{Option 1:} \quad \begin{aligned} z &= a + b + c \\ &\quad + d + e + f \end{aligned}$$

$$\textbf{Option 2:} \quad \begin{aligned} z &= a + b + c + \\ &\quad d + e + f \end{aligned}$$

If you choose option 1, be sure to put an empty `\mbox{}` before the plus-sign that starts the second line. Plus ($+$) and minus ($-$) are both unary and binary operators, and the space that LaTeX leaves around them differs for these cases. Putting an empty `\mbox{}` before the operator fools LaTeX into thinking that it's a binary operator, rather than a unary operator.

Be careful when using $\subset$ and $\subseteq$ as these symbols denote different notions: $\subset$ denotes proper inclusion, whereas in $A \subseteq B$, it could be the case that $A = B$.

Justin Zobel, in *Writing for Computer Science*, recommends the following:

- Number your definitions. Tom notes that it only makes sense if you actually refer to them by this index later in the text.
- Avoid proofs by contradiction. Tom reminds us that sometimes it's better to use proofs by contradiction since they might be shorter.

If you're writing a sequence enclosed by angle brackets (e.g., $\langle 1, 2, 3 \rangle$) use `\langle` and `\rangle`. You can also size them using `\left` and `\right` as noted previously.

Give a type for a variable for maximum readability (e.g., *vertex* v), but don't go overboard.

Watch out for assigning a wrong type. For example, *The values are represented as a list of numbers* L. Here, it's ambiguous whether L is a number or a list. Instead, write *The values are represented as a list* L *of numbers.*

Try not to use stacked fractions within inline text; try to keep the text at the same height.

Avoid using multiple stacked indices. For example, assume that set $W = \{w_1, w_2, ..., w_n\}$ has been defined. Suppose you wanted to write a sum using elements of the set W as subscripts.

✗ $\sum_i f_{w_i}$

✓ $\sum_{w \in W} f_w$

There is a difference between using `\cdots` as opposed to `\ldots`. For a sequence of numbers, use `\ldots`. Most other times use `\cdots`.

Stick with Greek letters that people know.

Lyn Dupré, in *Bugs in Writing*, recommends to avoid using *equals* and to instead use *is equal to.* Tom disagrees.