Style Guide

Introduction

CS22 is many students' first exposure to proofs. The course staff therefore wants to instill in those students – and all students! – a sense of what a good proof looks like. In this document, we will lay out our expectations for written work in CS22 and explain how feedback will be provided. We will give a few pointers for writing good proofs and direct you to other guides that go into more detail.

Why Style Matters

CS22 is not a course with proofs: it is a course *about* proofs, and the fact that you have proven your answer is much more important than the fact that you have the answer. **Again, your proof is more important than your answer.** This may seem counterintuitive – a proof seems like the 'work' of, say, a hard math problem; if you have the answer you clearly understand how to do the work, right?

It may help to think of CS22 like a debate class. Having opinions in a debate class doesn't earn you anything: *your arguments do.* Why is your answer right? Why should I believe you? When writing a homework, pretend you are debating someone who has just read the problem for the first time and thinks there is no way to answer it. You even have an advantage that you don't usually get: your answer isn't an opinion, it is a fact. Your proof (conveniently) will prove that.

With that in mind, your proof style is extremely important. You don't win a debate by being confusing or overly verbose or by not explaining yourself. You win by being clear, concise, and thorough. Take pride in your arguments and explain them to us, and we will be more likely to believe them – and if we believe your argument then it's very difficult for us to take a single point away from you.

How Style Affects Your Grade

Proof writing is an art, not a science, and we do not expect students to follow some particular design recipe. Like any art, it must be learned, and feedback is an important part of that. The tips below explain what we are considering in assessing your proof-writing abilities. Style is not a large part of your grade, and for a good reason: it is not meant to penalize you, it is meant to give you feedback. Take the comments to heart, not the points. *It is generally not likely that you will be able to argue for more points in the style category.*

Our textbook (Mathematics for Computer Science by Eric Lehman, F Tom Leighton, Albert R Meyer) gives some excellent tips. We reproduce them here:

State your game plan. A good proof begins by explaining the general line of reasoning, for example, "We use case analysis" or "We argue by contradiction."

Keep a linear flow. Sometimes proofs are written like mathematical mosaics, with juicy tidbits of independent reasoning sprinkled throughout. This is not good. The steps of an argument should follow one another in an intelligible order.

A proof is an essay, not a calculation. Many students initially write proofs the way they compute integrals. The result is a long sequence of expressions without explanation, making it very hard to follow. This is bad. A good proof usually looks like an essay with some equations thrown in. Use complete sentences.

Avoid excessive symbolism. Your reader is probably good at understanding words, but much less skilled at reading arcane mathematical symbols. So use words where you reasonably can.

Revise and simplify. Your readers will be grateful.

Introduce notation thoughtfully. Sometimes an argument can be greatly simplified by introducing a variable, devising a special notation, or defining a new term. But do this sparingly since you're requiring the reader to remember all that new stuff. And remember to actually define the meanings of new variables, terms, or notations; don't just start using them!

Structure long proofs. Long programs are usually broken into a hierarchy of smaller procedures. Long proofs are much the same. Facts needed in your proof that are easily stated, but not readily proved are best pulled out and proved in preliminary lemmas. Also, if you are repeating essentially the same argument over and over, try to capture that argument in a general lemma, which you can cite repeatedly instead.

Be wary of the "obvious." When familiar or truly obvious facts are needed in a proof, it's OK to label them as such and to not prove them. But remember that what's obvious to you, may not be – and typically is not – obvious to your reader. Most especially, don't use phrases like "clearly" or "obviously" in an attempt to bully the reader into accepting something you're having trouble proving. Also, go on the alert whenever you see one of these phrases in someone else's proof.

Finish. At some point in a proof, you'll have established all the essential facts you need. Resist the temptation to quit and leave the reader to draw the "obvious" conclusion. Instead, tie everything together yourself and explain why the original claim follows.

We also offer these tips courtesy of the CS22 staff:

- Read your proof aloud to yourself. Does it make sense logically and grammatically? Does it flow smoothly? If you stumble on words aloud, consider revising for clarity. **Writing a second draft is an excellent way to achieve this.**

- Read your proof *verbatim* to someone else who knows the subject (or imagine doing so). Do they understand it without additional explanation? If not, consider adding those explanations into the proof.

- Read proofs! Lots of them! Read our solution proofs, read our example proofs (under docs on the website), read the proofs in the textbook. This will teach you good habits because you will begin to recognize things that are ineffective or difficult to read.

- Pay attention to the length of your proofs. We set no hard cutoffs on length, but if your proofs are consistently much longer than the textbook's proofs or our homework solutions, you should revise them to minimize redundancy. There may also be a more-direct or elegant argument you haven't considered.

- Come to clinic! The TAs will happily talk to you about their thoughts on what makes up a good proof. Everyone has different ideas and perspectives, so coming to multiple hours to get feedback will be quite rewarding.

FAQ

Why does style matter if I did everything right?
If you have a really cool story, that doesn't mean people will buy your book. Communicating your ideas is important! We want to make sure you know how to do it effectively. If you start focusing on your communication skills on paper, we guarantee you will notice a sharp rise in your grades on written assignments (yes, even in STEM).

Where can I find some really beautiful proofs?
Look up *The Book*, which contains some of the most elegant proofs mathematics has yet created. While you're at it, read about Paul Erdoš̈ (pronounced like "AIR-dish"), an eccentric and interesting mathematician who inspired the creation of The Book. Finally, reddit has a math community (reddit.com/r/math) which, while sometimes pretentious, is a good place to watch mathematical minds communicate frankly.

How can I improve my proofs?
If you've already followed the pointers above, but still want to improve your proof writing, the best way to do so is to practice. A lot. This means reading papers (find research papers on topics that interest you and work through them!), reading through the textbook, and finding interesting problems to work on and prove. Go to problem sessions or to clinics, where you will be able to get additional problems!