Formulating and Writing Proofs

Santiago Cañez

Proofs are the bread and butter of modern mathematics; they are the key to how new mathematics is created, and the manner in which new mathematics is communicated. And yet, becoming comfortable with writing proofs can be quite a hurdle to overcome, in particular since this is so different than what is done in lower-level courses. But, like anything else, proof-writing a skill which can be learned. Here I summarize some things to consider when it comes to writing proofs, especially if you're only now being exposed to this aspect of mathematics.

First, you should understand why you're being asked to write proofs in the first place. Yes, mathematicians care about rigor and yes they care about precision, but the real reason why proofs play a big role in higher-level mathematics is that they force you to come to terms with the underlying concepts and the relations between them. No topic is ever truly understood well until you can manipulate the abstract concepts with ease and recognize how they fit together into a coherent framework, and being able to formulate a proof goes a long way to towards achieving this.

Second, you should understand that "proof-writing" is actually not the difficult part. Indeed, writing a proof is easy—there are some basic structural guidelines to consider, but mainly all you're doing is writing down your thoughts. The challenging part is in coming up with the proof in the first place and in reasoning through how you get from point A to point B. Recognize that *this* is where your focus should be, and that once you've figured out how a proof should work, the actual writing of the formal proof is for the most part straightforward.

Finally, you have to come to terms with the fact that proof-writing is not a formulaic process, where after you follow this step and then that step you're done. This is in stark contrast to the computational problems you've done in previous courses, and you should not expect that a similar algorithmic approach is how proof-writing will be learned. Proof writing (or, more importantly, formulating) requires much more ingenuity and creativity that computational problems, so you should get used to the idea that understanding concepts and *how they relate to one another* is what is most important.

Here, then, are some specific things to keep in mind:

Know definitions. Definitions are absolutely crucial; *everything* you do in proof-based mathematics is built up from definitions, and so you'll have no hope of making any kind of progress until you know those definitions yourself. Know them inside-and-out: What idea is the definition trying to get across? What's the intuition behind why the definition is formulated as it is? What happens if we modify this or that aspect of the definition—does the meaning change as a result?

In more computational courses you can often get by without knowing precise definitions—for instance, you can learn how to compute integrals in calculus without knowing how an integral is actually defined, or you can learn how to apply series convergence tests without knowing how convergence of series is actually defined. This will NOT be possible in higher-level courses, so you must absolutely understand definitions.

Review examples. You have to be familiar with examples. Given a definition, you should know examples of things which satisfy it AND things which don't; given a theorem, you should know

examples of situations in which it is applicable and situations where it isn't, and so on. Going through example after example helps to build intuition and helps to get a sense for the connections between different topics, which are essential to understand.

In addition, you should look at as many examples of proofs as you can get your hands on. Reading through proofs help to get a sense for the types of arguments which tend to show up, the types of techniques which tend to show up, etc. A big part of learning how to write proofs comes from experience, and when just starting out such experience comes—yes from trying to write proofs yourself—but also from looking at proofs written out by others.

Rephrasing statements. Given a statement, whether it be in a definition or theorem, you should be comfortable with understanding how to phrase it in a different way. As an example, it is crucial to understand the notion of contrapositive—which gives a way of rephrasing an implication—since most every proof you ever write will either be carried out directly or by contrapositive. (Yes, contradiction, induction, etc. are also important to understand, but I would say that direct proofs and proofs by contrapositive come first in the order of importance.) Get in the habit of trying to phrase some given statement in many different ways in order to become more comfortable with the underlying concepts.

Working backwards. The formal written proofs you'll find in books or posted solutions are usually phrased as "here is what we are assuming, and here is how we get from what we are assuming to what it is we want to prove." However, this is NOT how the person writing the proof actually came up with it in the first place. Improving your proof-writing skills requires becoming comfortable with working "backwards": start with what it is you want to show, and think about what kinds of things you would need to know in order to reach that conclusion; then think about what you would need to know in order to reach *those* things, and so on. Before you ever get to writing a proof, work out via some scratch work what the argument should look like, by working backwards.

Try something. No one, at first, can look at a statement to be proven and know immediately what needs to be done and what the proof should look like. After years and years of experience it will become easier to do so for material with which you're familiar, but do not expect that this is what should happen when you're first starting out. It is quite common to be presented with a problem and not know what to do—the point is that you have to get into the habit of *trying* something. No progress will ever be made if you don't try. If you're trying to prove some general fact, work through some specific concrete examples first to see what about those examples might work in the general setting as well.

Along these lines, you should expect to fail and be wrong often in your first attempts. Proof-based problems are not ones which are meant to take a few minutes to finish, but rather are ones you're meant to think and think about, possibly for days. If you get something wrong, understand why it was wrong and think about whether there is some way around it or if a whole new approach is required.

Time constraints. Working under the time constraint of an exam can be stressful, especially if you're one who normally takes a while to figure out a proof. To help with exam problems you need to train yourself to be able to recognize in a short time span what the underlying point behind a given problem should be. To help with this, when working through practice problems, try the following:

- During a first run through, spend a minute or so on each problem thinking about what solving it would entail. The goal here is not actually work on solving the problem, but rather to recognize what would be important in doing so. Write down any theorems or definitions which might be relevant; identity which proof techniques might be fruitful; write down some different ways of rephrasing what it is you're trying to prove (in particular, write down the contrapositive if the statement you're given is an implication); and think back to examples you've seen elsewhere to see if a technique used there might be relevant now.
- After this first run through, go back and now actually try to solve the problems, putting to use the things you thought about in the first run through.

Again, don't expect to be able to each problem in just a few minutes at the first—the goal is to be able to quickly pick up on what each problem is getting at. Over time, and with experience, it should become easier to get a sense for what to do when presented with a new problem.