MAU22200 Lecture 2

John Stalker

Trinity College Dublin

13 September 2021

Lecture notes

There are lecture notes posted to the webpage and Blackboard
Start reading Chapter 1 (Introduction) It's long, but you don't need to read it all now.
Its main purpose is to introduce the cast of characters for the drama, especially the main characters: normed vector spaces, metric spaces and topological spaces. We'll see more of each in later chapters.
It has a few other purposes:

- to give some back story on the main characters
- to introduce some background on sets and functions, not part of general topology but needed for it
- to fix some terminology and notation
- to give you lots of examples of proofs, and a few techniques

Reading and writing proofs

- Mathematical writing is dense. Reading and writing maths will be slower than other types of reading or writing.
- It's never written, and shouldn't always be read, in linear order.
- You can't memorise each step of every proof.
- There are three sorts of steps:
  - Steps which are purely mechanical: Details are generally omitted in proofs because they can be filled in mechanically.
  - Steps which are nearly mechanical: There are a few types which quickly become familiar.
  - Steps which require actual thought: These are fairly rare. You need to identify and remember these. Many proofs have no such steps.

What's purely mechanical?

There are algorithms to verify some types of statements.

1. Arithmetic statements, e.g.
   $2^{2^5} + 1 = 4294967297 = 641 \cdot 6700417$.
2. The statement that one set of linear equations do or don't imply another, e.g. $x + y + 2z = 5$ and $x + 2y + 3z = 5$ imply $2x + 3y + 5z = 8$.
3. The statement that one set of polynomial equations over $\mathbf{C}$ do or don't imply another, e.g. $x + y + z = 0$ and $x^2 - xy - xz + y^2 - yz + z^2 = 0$ imply $x^3 - 3xyz + y^3 + z^3 = 0$ and $xy + xz + yz = 0$.
4. The statement that one elementary function is the derivative (or antiderivative) of another, e.g. $\frac{d}{dx} \tan(\log x) = \frac{\sec^2(\log(x))}{x}$ or $\int \frac{\sec^2(\log(x))}{x} dx = \tan(\log x)$.
5. The statement that an elementary function has no elementary anti-derivative.

What's purely mechanical? (continued)

6. Logical statements using only Boolean operations, i.e. and, or, not, implies, if and only if, etc.
7. The corresponding statements about sets, e.g. if $A \subseteq B$, $B \subseteq C$ and $C \subseteq A$ then $A = B = C$. To convert this to purely Boolean form rewrite $A \subseteq B$ as $x \in A \Rightarrow x \in B$ and $A = B$ as $x \in A \Leftrightarrow x \in B$, etc.
8. Statements proved by simple operations on quantifiers, e.g. that the negation of "For all $\epsilon > 0$ there is a $\delta > 0$ such that if $0 < |x - w| < \delta$ then $|f(x) - z| < \epsilon$" is "There is an $\epsilon > 0$ such that for each $\delta > 0$ there is an x such that $0 < |x - w| < \delta$ and $|f(x) - z| \geq \epsilon$"

Details are normally omitted if you can fill them in yourself. At this point you should be able to cope with **??**, **??**, **??**, **??**, **??** and **??** but not **??** or **??**.

What's nearly mechanical?

Certain steps are more or less routine. At any point in a proof are a small number of these available. Which ones are depends on the form of what you're trying to prove. For example, if you want to prove that $A \subseteq C$ for some given subsets A and C then these are some things you could try:

- Assume $x \in A$ and try to prove $x \in C$.
- Use the fact that if $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$.
  - Try to find a B for which you know that $A \subseteq B$ and $B \subseteq C$.
  - Try to find a B for which you know that $A \subseteq B$ and for which proving $B \subseteq C$ looks easier than $A \subseteq C$.
  - Try to find a B for which you know that $B \subseteq C$ and for which proving $A \subseteq B$ looks easier than $A \subseteq C$.
- Assume that $A \nsubseteq C$, i.e. that there is an $x \in C$ such that $x \notin A$, and try to derive a contradiction.
- Use the fact that $\varnothing \subseteq C$ for all C and try to prove that $A = \varnothing$.

An example of following your nose

If $\varphi\colon X \to Y$ is a function then the preimage of $B \subseteq Y$ under φ is

$$\varphi^*(B) = \{x \in X : \varphi(x) \in B\}.$$

Most people write φ^{-1} in place of φ^*. This is a bad idea for subtle reasons.
There's a theorem that if $\varphi\colon X \to Y$ and $\psi\colon T \to X$ then $(\varphi \circ \psi)^* = \psi^* \circ \varphi^*$. How do we prove this? To prove functions are equal, usually we show that they take the same value at each point. The points here are sets, so we want to show that $(\varphi \circ \psi)^*(W) = (\psi^* \circ \varphi^*)(W)$. To prove sets are equal, we usually show that every element of one is an element of the other. So we want to show that $t \in (\varphi \circ \psi)^*(W)$ iff $t \in (\psi^* \circ \varphi^*)(W)$.
Unwrapping definitions can lead to long and confusing expressions, but simple definitions are usually fine. Like the definition of composition.

An example of following your nose (continued)

$(\psi^* \circ \varphi^*)(W) = \psi^*(\varphi^*(W))$. If you've just seen a definition and now you need to prove a statement where it appears then you will have to unwrap it. The definition of the preimage, applied to ψ and $\varphi^*(W)$ tells us that $t \in \psi^*(\varphi^*(W))$ iff $\psi(t) \in \varphi^*(W)$.
Applying it again tells us that $\psi(t) \in \varphi^*(W)$ iff $\varphi(\psi(t)) \in W$.
We can now use the definition of composition again:
$\varphi(\psi(t)) \in W$ iff $(\varphi \circ \psi)(t) \in W$. And of the preimage:
$(\varphi \circ \psi)(t) \in W$ iff $t \in (\varphi \circ \psi)^*(W)$.
Now we have a proof. $t \in (\psi^* \circ \varphi^*)(W) \Leftrightarrow t \in \psi^*(\varphi^*(W))$
$\Leftrightarrow \psi(t) \in \varphi^*(W) \Leftrightarrow \varphi(\psi(t)) \in W \Leftrightarrow (\varphi \circ \psi)(t) \in W$
$\Leftrightarrow t \in (\varphi \circ \psi)^*(W)$ for all t so $(\varphi \circ \psi)^*(W) = (\psi^* \circ \varphi^*)(W)$ for all W and hence $(\varphi \circ \psi)^* = (\psi^* \circ \varphi^*)$.
This is part of a lemma in the notes.

Gowers and Ganesalingam

You could imagine teaching a computer to find proofs the way computers play chess. At each stage there are a few possible moves. Create a new branch in your tree for each of these. If the number of branches gets too large, prune the ones which don't seem to be going anywhere.

People have done this. Tim Gowers and Mohan Ganesalingam even wrote a program which can prove the basic theorems of general topology *with no branching and no backtracking*! In other words, it doesn't try various branches, it just picks the "most obvious" option at each stage and doesn't ever revisit that choice. The rules for what's "most obvious" aren't obvious. They were arrived at by trial and error on many examples. You'll also learn what's "most obvious" by trial and error on many examples.

Reading and writing non-linearly

Proofs are normally printed in an order suited to verification. That's not the order they're written in.

If I'm proving $A \subseteq C$ by using the first strategy above then I'd write "Suppose $x \in A$. ... so $x \in C$. We've just seen that all $x \in A$ belong to C, so $A \subseteq C$." Then I'd try to fill in the middle.

If I were using one of the other strategies I might write "Suppose $x \in C$ and $x \notin A$. ... but this is impossible, so our assumption that there is an $x \in C$ which does not belong to A is untenable, and therefore $A \subseteq C$." Again, I'd then try to fill in the middle.

The word "suppose" has the same logical meaning in both cases, but the intent is different. You don't really find out what's going on until later. Possibly much later.

You'll need to read some proofs multiple times to understand the structure. Read it once to identify the top level structure, so you can split it into pieces. Read it again to understand the structure of each piece. Repeat as needed.

Identifying the key ideas

Most of any proof is either purely mechanical or nearly mechanical. You want to identify the few bits, if any, which aren't. Those are the key ideas of the proof. If you remember those you can easily reconstruct the rest.

This module, more than any other, is the one where you learn to read and write proofs. Most of that is learning how to do the nearly mechanical bits. If you learn than than you can recognise the key ideas in other people's proofs and can reconstruct a full proof from them. And you can write your own proofs more easily.