How to math: an introduction to rigor

Roy Dong

October 29, 2020

The purpose of this document

This document is meant to be a guide for engineering graduate students interested in learning higher-level mathematics. Oftentimes, an undergraduate curriculum in engineering does not provide sufficient infrastructure to begin to analyze mathematics rigorously, and this infrastructure is often presumed in advanced math courses. This document will focus on the structure of formal logic and how it is instantiated in mathematical proofs, which will hopefully give the reader the necessary tools to look at a proof and ask: 'Is this proof right?' That being said, this document also focuses on relatively specialized topics on formal logic, so even someone with training in formal methods may be interested in a quick skim. Along the way, perhaps this document may also contain content to convince the skeptic that rigor has value.

This document will probably be most helpful to people who have interacted with mathematical concepts and developed an intuition, as well as had some experience programming, but have not done much in rigorous mathematics. I will be making analogies to programming throughout, and Fitch notation itself, as I present it, will be a 'programming way to think about proofs'.

That being said, the purpose of the document is to provide the reader with the capacity to make their mathematical proofs more rigorous, so I will be skimming a lot of the more technical constraints on Fitch notation and mainly preserve the concepts: ironically, there will be a lack of rigor as I discuss it.

What is rigor?

In many ways, mathematical proofs are like a programming language. There is a syntax and rules that are valid, i.e. accepted by the mathematical community as logical inferences which can be done. So, to say 'This proof is rigorous.' is essentially the same as saying 'This code compiles.' To make a hand-wavey argument is to say: 'There's pseudocode for it right now with some placeholder functions in the API, but if I wanted to, I know that I could finish it. All I have to do is put in the time. If you really want a pedantic proof, you can put in the time.'

Generally, mathematicians are fine with hand-wavey arguments when they are comfortable they can complete the code. When you hear a mathematician complain about a lack of rigor, they are complaining that it would be difficult to finish the code themselves. When the code doesn't compile, the truth of the assertion is threatened, as we'll see in examples below.

Mathematics, as a field, is quite varied. There are things that are intuitive after some training, such as linear algebra. Then there are things that are horribly unintuitive, such as Gödel's incompleteness theorem, the Vitali set, or the Löwenheim-Skolem theorem.

When doing mathematics in fields that have intuitive content, it often amounts to having an intuition, tracking down the fundamental properties that make that intuition true, and then writing that intuition down in a fashion that can be understood by other mathematicians. Hand-wavey-ness often freely abounds here.

When mathematics is done in unintuitive fields, formalization reigns supreme. By rigorously checking every step, mathematicians are able to trace down unexpected conclusions of their assumptions. Perhaps, eventually, mathematicians can develop an intuition even for these unintuitive things, so I'm not too committed to this distinction for the field at large. What is important is that this distinction is made at an individual level for a student.

Your first exposure to probability theory will probably come with little intuition, and rigor should be the life-raft you cling to until you can build your yacht of intuition. This is also one of the most painful parts of learning math for the first time: your instructor has intuition whereas you do not. He waves his hands and you can only stare at the motion dumbfounded and ask: 'What just happened?' This pain may be made worse if you are in a room full of mathematics undergraduates who also have intuition, due to their extended exposure. Many graduate students find it difficult the first time they take a higher-level math course because the expectations of them are not clear. Essentially, they have to simultaneously learn the content of the mathematics, e.g. real analysis, topology, measure theory, while learning the syntax of a mathematical proof. Without prior exposure to the latter, it's often very difficult to decouple the mathematical content itself from the structure of the proof. So, the rest of this document will focus on the structure of mathematical proofs, and cover very little math proper.

Fitch-style notation

I want to open with a caveat that I'm not certain what Fitch notation was designed to do. Was it meant to be a didactic tool? Was it meant to be a system that could be computed? Was it meant to just be a theoretical system that is equal in expressive power to other first-order logic systems?

I am not an expert in that area, and this section will only detail how I have used Fitch notation to my personal benefit. As such, I have likely made adjustments to Fitch's original material. Additionally, I have no idea if this is the best way to think about things; it is simply the technique I used that I found helpful. In particular, I will not be nit-picky about the first-order logic structure of statements. Logicians, avert your eyes; what follows may break your heart.

Also, for those who are very much so interested in learning this material in more depth, I learned everything I know about Fitch from Barker-Plummer, Barwise, and Etchemendy's Language, Proof, and Logic.

First, let's introduce the symbols that are going to be used. This is in Figure 1.

Symbol	Interpretation
\wedge	and
V	or
\rightarrow	implies
-	not
\forall	for all
Э	there exists
	contradiction

Figure 1: The basic symbols we will use.

Next, we will see that the fundamental building block of Fitch notation is the 'turnstile' symbol \vdash :

Assumptions	
Conclusions	

This symbol is absolute magic. Above the horizontal bar, there are assumptions. Below the horizontal line, as long as the left line still extends, you are in an environment where these assumptions can be treated as true.

 $\begin{array}{c} \text{Suppose } p. \\ \hline p \text{ is true here!} \\ p \text{ is still true here!} \\ p \text{ continues to be true here!} \end{array}$

p has ceased to be true.

What this symbol does is it lets you scope the domain of your assumption. In the assumption, you can also take free variables and bind them to values.

 ϵ is a free variable here! It doesn't refer to a value or mean anything!

 $\begin{array}{l} {\rm Pick} \ \epsilon > 0. \\ \hline \epsilon \ {\rm is \ treated \ as \ a \ constant \ in \ here! \ Just \ like \ an \ ordinary \ number!} \end{array}$

 ϵ is a free variable again.

You can nest these turnstiles too.

Suppose p .	
Suppose q .	
p is true here!	
q is also true here!	

q is not true here, but p is.

This notation is wonderful at making you consciously track the different variables in use, as well as the dependencies of certain values on others. So, like a programming language, you track which environment you're in, what variables are currently scoped, and then you have to make due with whatever's available to you at that point.

Additionally, in the system, you can always add another \vdash , throw any assumptions that you want in, make a new environment, and see what it leads to. The caveat is that you can't take those assumptions out of the environment without some work.

Now, let's get into the actual rules of inference in Fitch's system. Also, in the interest of space, we will often write $p \vdash q$ horizontally to indicate:



There's a peaceful symmetry in the setup of Fitch's system: every symbol has one rule by which it can be introduced, and one rule by which it can be eliminated. Let's discuss these rules and maybe do a few examples.

Here, we will use p, q, r, \ldots to denote propositions, which are simply atomic statements that are either true or false, e.g. 'William is Billy Jr.'s father.' or 'Phoenix is the capital of Arizona'. We will also use $P(x), Q(x), \ldots$ to denote sentences with one free variable, e.g. 'x > 5' or 'x is odd'. These sentences are neither true nor false until some value of x is specified.

Most of these rules should be relatively intuitive, but take some time to look at each and convince yourself that they are logically valid. Also, note that when I use \wedge as a symbol, it means 'and' within the proof system, which is not to be confused with when I literally type the word *and*.

- \rightarrow introduction: From $p \vdash q$, we can conclude $p \rightarrow q$. In other words, if we can assume p and then prove q, we can conclude p implies q.
- \rightarrow elimination: From $p \rightarrow q$ and p we can conclude q. This is known as 'modus ponens'.
- \wedge introduction: From p and q, we can conclude $p \wedge q$. This will seem weird until you recall the above distinction that although \wedge means 'and', this rule of inference still means something. \wedge is inside the programming language, whereas *and* is used as an English word.
- \wedge elimination: From $p \wedge q$, we can conclude p.
- \lor introduction: From p, we can conclude $p \lor q$. This may seem like a pointless maneuver, since it is strictly a loss of information, but sometimes it's useful.
- \lor elimination: From $p \lor q$ and $p \to r$ and $q \to r$, we can conclude r. This is commonly known as 'proof by cases': you either know p or q is true, but they both imply r so, regardless of which one is true, we can conclude r.
- \neg introduction: From $p \vdash \bot$, we can conclude $\neg p$. This is commonly known as 'proof by contradiction': if you suppose p and arrive at a logical inconsistency, you can assume $\neg p$. Some logicians will not allow this sort of inference, if you look at the discussion of the 'law of excluded middle', but that's besides the point. You, in your day-to-day life, will always be able to do a proof by contradiction.
- \neg elimination: From $\neg \neg p$, we can conclude p. Double negations cancel out.
- \perp introduction: From $p \land \neg p$, we can conclude \perp . A contradiction is defined as showing some proposition and its negation are both true.
- \perp elimination: From \perp we can conclude p. This principle is known as 'reductio ad absurdum', which translates to: 'From absurdity, everything follows.'

There's two more symbols I want to get into, but first I want to focus on \perp elimination. This isn't just a quirk of Fitch notation, this generally happens in a lot of proof systems: if you accidentally assumed $p \land \neg p$, or the existence of some object that can't exist, you can literally prove anything. This is why the call for rigor is often important: a research paper can be gibberish if you accidentally assumed this somewhere.

Great, now we can move onto the quantifier rules. x will be seen as a variable, which can be free or bound, whereas c will be seen as a constant. Think of it like a fixed number.

• ∀ introduction: If we can pick an arbitrary item and prove something, we can conclude that it holds for all items. From

$$\begin{array}{c|c} \text{Pick any } x \text{ such that } P(x). \\ \hline Q(x). \end{array}$$

we can conclude $\forall x : P(x) \to Q(x)$.

- \forall elimination: From $\forall x : P(x)$ we can conclude P(c) for any constant c. This is often called 'instantiation'. What may be confusing is that, in practice, x and c may actually share the same symbol. For example, we may see 'There exists x such that x > 5', and in a later line of the proof, the sentence 'Pick x such that x > 5.' So, depending on context, a symbol like x will either have to be treated as a free variable, a bound variable, or a constant.
- \exists introduction: From P(c) we can conclude $\exists x : P(x)$.
- \exists elimination: I've decided to exclude this since it is a little hoary and not necessary for our development.

Next, to cement these concepts, I'll go over a few examples of this system in action.

Examples

First, I'll do some proofs entirely in first-order logic so you can get used to the rules. Then, I'll use a stylized version of this system to prove some simple mathematical facts.

First, let's prove modus tollens, i.e. from $\neg q \rightarrow \neg p$ and p we can conclude q. This will also require the assumption of the law of excluded middle: $q \lor \neg q$. First, let's write out our goal in this system:

$$\begin{array}{c} \neg q \to \neg p \\ p \\ q \lor \neg q \\ \hline ??? \\ q \end{array}$$

So, we can take our goal and express it as this skeleton of a proof. Now let's fill in the blanks. My first instinct is that I should use proof-by-cases on the law of excluded middle: one of the two cases is already clear, i.e. it's easy to show q is true in the case that q. Thus, let's use \lor elimination to update this skeleton, with one of the cases already done.

$$\neg q \rightarrow \neg p$$

$$p$$

$$q \lor \neg q$$

$$\left| \frac{q}{q} \right|$$

$$\left| \frac{\neg q}{q} \right|$$

$$q$$

What is true in the environment where '???' is written? We have $\neg q \rightarrow \neg p$ and p and $q \lor \neg q$ and $\neg q$. We know $\neg q$ is true in this environment, and an implication from it, so let's follow our nose:

$$\begin{vmatrix} \neg q \to \neg p \\ p \\ q \lor \neg q \end{vmatrix}$$
$$\begin{vmatrix} \frac{q}{q} \\ \neg q \\ \neg p \\ p \land \neg p \\ \bot \\ q \\ \end{vmatrix}$$

We arrive at $\neg p$, and can more or less sense that a reductio ad absurdum argument is looming, so we finish the proof. The steps are \rightarrow elimination, \land introduction, \perp introduction, \perp elimination.

Here's one more example, of one direction of De Morgan's laws. We want to show:

$$\begin{array}{|} \neg p \land \neg q \\ \hline \neg (p \lor q) \end{array}$$

We begin by unraveling how we would show this. We need to introduce a \neg :

$$\begin{array}{c|c}
\neg p \land \neg q \\
\hline p \lor q \\
\hline ??? \\
\bot \\
\neg (p \lor q)
\end{array}$$

The assumption in the innermost environment suggests we need a proof by cases:

$$\begin{array}{c|c} \neg p \land \neg q \\ \hline p \lor q \\ \hline p \\ \hline ??? \\ \bot \\ \hline q \\ \hline ??? \\ \bot \\ \hline \bot \\ \neg (p \lor q) \end{array}$$

Then the answer seems relatively apparent at this point:

$$\begin{array}{c|c} \neg p \land \neg q \\ \hline p \lor q \\ \hline p \lor q \\ \hline p \\ \neg p \\ p \land \neg p \\ \bot \\ \hline q \\ q \land \neg q \\ \downarrow \\ \downarrow \\ \neg (p \lor q) \\ \end{array}$$

Note that, whenever proofs are done in this abstract first-order logic sense, you can basically use that as a new rule of inference. So, feel free to just memorize De Morgan's laws and move on with your life if you find this system too arduous.

Furthermore, if $p \vdash q$ and $q \vdash p$, then they are equivalent and can be used interchangeably.

If you're interested in practicing this system more, here's a few exercises to try and show. These are also just general logical inferences that are good to have. (Except the last one. That one's just for fun.) Some of them might require the law of excluded middle.

$$\begin{vmatrix} \neg (p \lor q) \\ \neg p \land \neg q \end{vmatrix}$$
$$\begin{vmatrix} p \to q \\ q \lor \neg p \\ p \to q \end{vmatrix}$$
$$(p \to q) \to r$$
$$p \to (q \to r)$$

Here are two rules of inference we can use with quantifiers, which are generally very useful. There's a few more tools required to show them which I'll omit as they aren't necessary for the main point.

$$\begin{vmatrix} \exists x : \neg P(x) \\ \neg \forall x : P(x) \end{vmatrix}$$
$$\begin{vmatrix} \neg \forall x : P(x) \\ \exists x : \neg P(x) \end{vmatrix}$$

Now, I'll go into examples with math terms instead of propositions. The key thing to try and look at is not the content, but the structure of the proof. I will also immediately become less rigorous about only using syntactic consequences, but this balancing quantity of semantic analysis is necessary to prove all but the most basic facts.

Show that $f(x) = x^2$ is continuous.

Before we begin, we'll have to recall some definitions. So, f is continuous at a point x if for all $\epsilon > 0$ there exists a $\delta > 0$ such that for all y, we have $|x - y| < \delta$ implies $|f(x) - f(y)| < \epsilon$. We say f is continuous if it is continuous at all points x in its domain. So, we want to show:

$$\forall x : \forall \epsilon > 0 : \exists \delta > 0 : \forall y : (|x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon)$$

Let's put this into a Fitch-esque framework.

Pick an arbitrary x.
Pick an arbitrary
$$\epsilon > 0$$
.
Take $\delta = ???$.
Pick any y.
Pick any y.
 $|x - y| < \delta$
 $???$
 $|f(x) - f(y)| < \epsilon$

Just by unpacking the quantifiers, we now know what we need to show. The next part will require some mathematics proper, rather than just Fitch-like manipulations. But here's the proof, built up the way I would do it:



Now we just have to find the right value of δ . We can see that we simply need $(2|x|+\delta)\delta < \epsilon$. An easy way to ensure this is to pick δ such that $2|x|\delta < \epsilon/2$ and $\delta^2 < \epsilon/2$. Take $\delta = \frac{\min(\sqrt{\epsilon/2}, \epsilon/4|x|)}{2}$.



Keeping these environments diagrammatically drawn, and also tracking the scoping of the variables is great; for example, it greatly clears up confusion about continuity versus uniform continuity. As an exercise, show that f(x) = 2x is uniformly continuous, that is:

$$\forall \epsilon > 0 : \exists \delta > 0 : \forall x : \forall y : |x - y| < \delta \to |f(x) - f(y)| < \epsilon$$

Note that now, if you're careful about scoping, the diagrammatic form enforces that your choice of δ cannot depend on x, as x is not scoped at the time δ must be chosen.

Another exercise is to show $f(x) = x^2$ is not uniformly continuous. To start you off, you can propagate the negations we use our rules of inference to find equivalent statements.

$$\begin{aligned} \neg \forall \epsilon > 0 : \exists \delta > 0 : \forall x : \forall y : |x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon \\ \exists \epsilon > 0 : \neg \exists \delta > 0 : \forall x : \forall y : |x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon \\ \exists \epsilon > 0 : \forall \delta > 0 : \neg \forall x : \forall y : |x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon \\ \exists \epsilon > 0 : \forall \delta > 0 : \exists x : \exists y : \neg (|x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon) \\ \exists \epsilon > 0 : \forall \delta > 0 : \exists x : \exists y : |x - y| < \delta \rightarrow |f(x) - f(y)| < \epsilon) \end{aligned}$$

One piece of advice I frequently overhear is: 'When you don't know how to prove something, just step back and ask yourself what you're trying to show.' This advice is very good, but also can sound tautologically trivial. Additionally, without much experience, it can be impossible to really know how to do this. Fitch calculus should provide a guideline on how to start a proof when you don't know what to do. You can sketch the skeleton out of what you need to show, and which smaller steps you need to take. This can be done almost mechanically. So, this advice was actually a concrete instruction on what to do, not some abstract idea or approach to consider.

Here's a more complicated example, which is a mock prelim problem I frequently used when I was a graduate student. It is taken from some part of Stephen Boyd and Lieven Vandenberghe's *Convex Optimization*.

A symmetric matrix $X \in \mathbb{R}^{n \times n}$ is a *Euclidean distance matrix* if there exist vectors p_1, \ldots, p_n of arbitrary dimension such that $x_{ij} = \|p_i - p_j\|_2^2$ for $i, j \in \{1, \ldots, n\}$.

Show that X is a Euclidean distance matrix if and only if there exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

Hint: Consider Y with elements $y_{ij} = p_i^{\mathsf{T}} p_j$. This is the *Gram matrix* associated with vectors p_1, \ldots, p_n .

We have to show an 'if and only if', which will require proofs of both directions. Additionally, we are given a definition, which we can use to replace any instance of 'X is a Euclidean distance matrix'. We set up the Fitch form:

There exist vectors p_1, \ldots, p_n such that $x_{ij} = \|p_i - p_j\|_2^2$ for $i, j \in \{1, \ldots, n\}$. ??? There exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$. There exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$. ??? There exist vectors p_1, \ldots, p_n such that $x_{ij} = \|p_i - p_j\|_2^2$ for $i, j \in \{1, \ldots, n\}$.

Note that X is scoped throughout the global environment, i.e. it is treated as a fixed value everywhere. Let's do one of them at a time. First:

There exist vectors p_1, \ldots, p_n such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$.

There exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

We have an existential quantifier in both the header and the footer, so let's just do the \exists introduction and elimination rules in those locations.

There exist vectors
$$p_1, \ldots, p_n$$
 such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$.
Let p_1, \ldots, p_n be vectors such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$.
Take $Y = ???$.
We have shown that $V > 0$ in $\mathbb{D}^{n \times n}$ such that $x_i = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$.

We have shown that $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

There exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

As mentioned before, the same symbol, in this case p_i , will be used as a bound variable and a constant, depending on the environment and context. In the line 'There exist vectors p_1, \ldots, p_n such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in$ $\{1, \ldots, n\}$.' it is a bound variable. (It corresponds to $\exists x : P(x)$.) In 'Let p_1, \ldots, p_n be vectors such that $x_{ij} = \|p_i - p_j\|_2^2$ for $i, j \in \{1, \ldots, n\}$.' it is a constant. (It corresponds to P(c) for some constant c. At this point the p_i can just be treated as fixed values.) This slight overloading of notation abounds in mathematics, and it's good to identify it until you get very comfortable scoping things, in which case it will become second nature.

The hint actually tells us what to take for Y, so let's do that.

There exist vectors p_1, \ldots, p_n such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$. Let p_1, \ldots, p_n be vectors such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$. Take $Y = P^{\intercal}P$. TODO: We need to show $Y \succeq 0$ and $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$. We have shown that $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

There exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

It's very clear what we need to show now, and this direction of the proof is easy to complete.

There exist vectors p_1, \ldots, p_n such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$. Let p_1, \ldots, p_n be vectors such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$.

Take $Y = P^{\intercal}P$.

Note that $Y \succeq 0$ since it can be written $Y = P^{\intercal}P$. Furthermore, note that $x_{ij} = \|p_i - p_j\|_2^2 = p_i^{\intercal}p_i + p_j^{\intercal}p_j + 2p_i^{\intercal}p_j = y_{ii} + y_{jj} + 2y_{ij}$. The first equality is by assumption on the p_i . The second equality is expanding the norm. The third equality is by our chosen definition of Y. We have shown that $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

There exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \ldots, n\}$.

Now let's do the other direction, which is a bit harder, but not too much worse. Again, let's start by using \exists introduction and elimination as needed.

There exists some $Y \succeq 0$ in $\mathbb{R}^{n \times n}$ such that $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \dots, n\}$. Pick $Y \in \mathbb{R}^{n \times n}$ such that $Y \succeq 0$ and $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$ for $i, j \in \{1, \dots, n\}$. Let $p_i = ???$.

$$??? Thus, x_{ij} = ||p_i - p_j||_2^2 \text{ for } i, j \in \{1, \dots, n\}.$$

There exist vectors p_1, \ldots, p_n such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$.

We did the other direction, so we have a strong instinct on what the p_i should be. Let's do this.

 $\begin{array}{c|c} \hline \text{There exists some } Y \succeq 0 \text{ in } \mathbb{R}^{n \times n} \text{ such that } x_{ij} = y_{ii} + y_{jj} - 2y_{ij} \text{ for } i, j \in \{1, \ldots, n\}. \\ \hline \text{Pick } Y \in \mathbb{R}^{n \times n} \text{ such that } Y \succeq 0 \text{ and } x_{ij} = y_{ii} + y_{jj} - 2y_{ij} \text{ for } i, j \in \{1, \ldots, n\}. \\ \hline \text{Take the SVD of } Y = U\Sigma U^{\intercal}. \text{ Let } P = \Sigma^{1/2} U^{\intercal}. \\ \hline \text{Let } p_i \text{ be the ith column of } P. \\ \hline \text{We have that } x_{ij} = y_{ii} + y_{jj} - 2y_{ij} = p_i^{\intercal} p_i + p_j^{\intercal} p_j + 2p_i^{\intercal} p_j = \|p_i - p_j\|_2^2. \\ \hline \text{The first equality is by our assumption on } Y. \\ \hline \text{The second equality is by our definition of } p_i. (Note that <math>y_{ij} = p_i^{\intercal} p_j$ by definition.) \\ \hline \text{The third equality is by expanding the norm.} \\ \hline \text{Thus, } x_{ij} = \|p_i - p_j\|_2^2 \text{ for } i, j \in \{1, \ldots, n\}. \end{array}

There exist vectors p_1, \ldots, p_n such that $x_{ij} = ||p_i - p_j||_2^2$ for $i, j \in \{1, \ldots, n\}$.

When I gave this as a mock question, students did not cleanly take apart what they needed to show. More commonly, students would play around with the given facts and definitions in a rather haphazard fashion. It wouldn't be clear at any given moment which direction of the proof they were pursuing. Oftentimes, this also led to mistakes.

For example, when attempting to show that 'if' direction, students would often try to refer to this p_i before they were appropriately scoped. The reason this was problematic is that, by referring to the p_i prior an explicit definition, they are already implicitly assuming X is a Euclidean distance matrix, which is the very thing they were trying to prove. You only know that Y is positive semi-definite, and satisfies the equality $x_{ij} = y_{ii} + y_{jj} - 2y_{ij}$. Anything else you want to be true of Y, such as $Y = P^{\mathsf{T}}P$ for some P of significance, has to be argued. Put another way, the chain of equalities $x_{ij} = y_{ii} + y_{jj} - 2y_{ij} = p_i^{\mathsf{T}}p_i + p_j^{\mathsf{T}}p_j + 2p_i^{\mathsf{T}}p_j = ||p_i - p_j||_2^2$ shows up in both proofs, but these equalities hold for different reasons depending on the direction of the proof you are doing.

Again, we hopefully see the strength of this diagrammatic approach to ensuring we scope our assumptions correctly, define environments formally, and we don't overstep ourselves in using assertions that aren't true in our given environment.

Closing remarks

This basically outlines how I interacted with math during my first few years of grad school. I hope this document can be of some help to students who struggle with figuring out how to make a proof technically sound.

As a closing comment, I'll note that logic itself distinguishes between syntactic consequence and semantic consequence. The Fitch manipulations have focused on syntactic consequences: we make arguments based on a set of mechanical rules on how symbols can be moved around. When intuition is weak, it's best to lean on this syntactic reasoning: only use given definitions and rules of inference. (Here, of course, in practice, the syntactic constraints will generally be much weaker than those a computer program or logician would require, but it's best to try to be as formal as possible to ensure proper reasoning at first.) As intuition builds, one can begin to reason about things semantically, e.g. 'this and that are consequences because of properties of the objects themselves.' In many ways, this is when a mathematician comes into fruition: when they can manipulate both symbols and concepts with an adept rationale.

After a point, you might begin to question the magic of the fact that syntactic manipulations, which are mechanistic in nature, somehow move in lockstep with semantic manipulations, facts which are true because of properties of the abstract objects themselves. The fact that these two drastically different systems somehow yield the same conclusions is the result of some arcane witchcraft, which is the topic of study in metamathematics courses. If this piques your interest at all, that is something worth looking into.