

Proofwriting Checklist

In the Guide to Proofs, we gave a number of examples of good and not-so-great proofs, along with what aspects of them we liked or disliked. If you haven't yet done so, take a few minutes to go and read over it.

In this handout, we've distilled five major points about proofwriting that we will specifically be looking for when grading your assignments. They are as follows:

- Clearly articulate your start and end points.*
- Every sentence in your proofs should be load-bearing.*
- All variables should be scoped and introduced properly.*
- Make specific claims about specific variables.*
- Write in complete sentences and complete paragraphs.*

We strongly recommend that you work through this checklist on every proof that you write. Doing so will help you improve your proofwriting and possibly smoke out some underlying logic errors. Also, ***we will be applying this checklist to the proofs that you submit*** as part of how we grade, so catching these sorts of issues before you submit will help us give you even better targeted feedback on your proofs.

The remainder of this handout goes into more detail about what each of these rules mean.

Clearly Articulate Your Start and End Points

When you're writing a proof, you're laying out an argument that explains why a certain result is true. Most proofs have a number of intermediate steps that build up toward a larger result. When writing a proof, it's important to make sure that the reader has a clear sense of where it is that you're going and how you're going to arrive there. Otherwise, your proofs will be extremely hard to read, since while the reader might follow each *individual* step, they might have no idea where you're going with things. Think about how you might write an argumentative essay – if you just list a series of facts without giving some idea of where you're ultimately going, your readers are going to have a heck of a time trying to make sense of what you're doing!

Let's illustrate this with an example. Consider the following proof:

Proof: Consider an arbitrary $x \in A$. Since $x \in A$ and $A \subseteq B$, we see that $x \in B$. And, since $x \in B$ and $B \subseteq C$, we see that $x \in C$. Therefore $x \in C$, as required. ■

Here's a question for you – what exactly is this proof trying to accomplish? It's hard to say, since we don't know that A , B , and C are, it seems like the statements $A \subseteq B$ and $B \subseteq C$ come out of nowhere, and the conclusion doesn't say exactly why any of this matters.

The above proof was written for the following theorem:

Theorem: If $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

With knowledge of the theorem in mind, the proof makes more sense. We know that $A \subseteq B$ and that $B \subseteq C$ by assumption, and we're looking at elements of A and trying to get them as elements of C because we're trying to prove something about the subset relation. But that still shifts a lot of work to the person reading the proof. A better proof would provide more guidance about where everything comes from and where everything is going. Here's what that might look like:

Proof: Let A , B , and C be arbitrary sets where $A \subseteq B$ and $B \subseteq C$. We will prove that $A \subseteq C$. To do so, choose an arbitrary $x \in A$. We will prove that $x \in C$.

Since $x \in A$ and $A \subseteq B$, we see that $x \in B$. And, since $x \in B$ and $B \subseteq C$, we see that $x \in C$. Therefore $x \in C$, as required. ■

Compare this proof to the one before it. Even if you had no idea what the theorem was when going into this proof, you could still see exactly what's being done – what's being assumed, what's being proved, how the logic flows, etc. There's no more mystery about why $A \subseteq B$ and $B \subseteq C$ are true: we can see that they're true by assumption.

There's a number of reasons why it's worthwhile to set up your proofs this way. First, when you're still working through the problem and trying to figure out why exactly the result is true, this step forces you to write out exactly what it is that you're assuming and what you need to prove. That makes it much easier to figure out what directions you should consider. It also forces you to articulate very precisely what it is that you need to establish. If you look at the overall theorem to prove here, it might seem, well, kinda *obvious*. Like, “well, *of course* if A is a subset of B and B is a subset of C , then A is a subset of C – that's just what subset *means*!” But if you start unpacking the definitions and articulating where specifically you're going to start and end, it becomes much easier to see what you need to do.

Every Sentence in Your Proofs Should Be Load Bearing

When you're writing a proof, you are trying to convey a mathematical argument, and each step in what you write should advance your argument. As a general rule, every statement in a proof should do one of the following things:

- **Set up a goal.** As mentioned in the preceding pages, your proof should start off with an introduction that clearly articulates a start and end point. In larger proofs, you might find yourself needing to prove an auxiliary result that you'll use to build up to the larger result, and when you do that, you'll similarly want to set up what it is that you're trying to prove.
- **Introduce a new variable.** Sometimes, in the course of a proof, you'll need to introduce new variables. If you're proving something universally-quantified, you might want to say something like "let x be an arbitrary bananafish," and if you're proving something existentially-quantified you might want to say something like "since n is even, we know there is an integer k such that $n = 2k$."
- **Combine preceding results into something new.** Any sentence that doesn't set up a new goal or introduce a new variable should make progress toward the result by taking some number of preceding statements and deriving some new, *mathematically rigorous* result from those preceding statements. For example, you might say something like "since $n = 2k$, we see that $n^2 = 2(2k)^2$ " or "since $A \subseteq B$ and $x \in A$, we learn that $x \in B$."

If you find yourself reading over a sentence that doesn't accomplish any of these goals, it is likely unnecessary and should be eliminated. This is a great way to reduce the size of your proofs and to make sure that you're being rigorous.

This is a particularly useful check to apply to a proof after you've first finished writing it, since often times in the course of solving a problem and writing up a first proof draft you'll go in a direction that ultimately ends up not being necessary, or write out some high-level lines of reasoning that you then make more rigorous later on. If you look over the Guide to Proofs and look at the proof that $\log_2 3$ is irrational, you'll see an example of an initial draft of a proof, some analysis as to why it feels a bit disorganized, and then some suggestions for cleanup. Most of the cleanup there can be summarized by looking at the three above classes of statements and eliminating anything that doesn't fit.

All Variables Should Be Scoped and Introduced Properly

In programming languages like C, C++, and Java, you're required to declare variables before you use them. The type of the variable lets the reader (and the compiler!) know what sort of thing the variable can hold and what it represents. If you try to use a variable you haven't declared, or if you try to treat a variable of one type as though it had a different type, you get a compiler error because there's something amiss with what you've done.

Variables in mathematical proofs obey a similar sort of convention. When writing proofs, it's important that you clearly and precisely articulate what each variable stands for and, additionally, where it comes from. When you use a variable in a proof, you should explicitly articulate

- the name of the variable,
- what value it represents, and
- where it comes from.

Those last two points are critical in writing proofs. Every variable that you use should be of one of the following types:

- **An arbitrarily-chosen value.** A variable like this doesn't represent some specific number, set, or quantity, but rather an arbitrarily-chosen value. Variables like these often arise in the context of proving universally-quantified statements. For example, if you want to prove the claim "for any natural number n , if n is even, then n^2 is even," you might introduce a variable n like this:

Let n be an arbitrary even natural number.

Consider an even natural number n .

Let n be an even natural number.

Here, we're indicating that the variable is named n , its value is some even natural number, and that it's chosen arbitrarily.

- **An existentially instantiated value.** Sometimes, you know that some quantity must exist, but you don't know what it is. For example, if you know that n is an even natural number, you know that n must be twice some other natural number, and so you might give it a name, as shown here:

Since n is even, there is some integer k such that $n = 2k$.

It's important to note that this number k is *not* chosen arbitrarily. That would imply that any choice of k would work here, but that's not true: there's only one choice of k you can pick where $n = 2k$. Rather, k is called an **existentially instantiated** variable, because we know that *there exists* some value with some property and we're introducing the variable k as a way of saying what that value is.

- **An explicitly chosen value.** Sometimes, you'll introduce a variable simply as a simpler way of referring to some other quantity. For example, we might say something like this:

Let $m = 2k^2$.

Or, we could say something like this:

Consider the set $D = \{ x \in S \mid x \notin f(x) \}$.

Here, we're just giving a name to an existing quantity, which functions like a constant in a language like C, C++, or Java.

When you write up a proof (or, more generally, when you're reading something mathematical), it's important to make sure that you can look at each variable and clearly tell whether that variable is arbitrarily chosen, existentially instantiated, or explicitly chosen. Just like variables in C, C++, or Java, this helps you clearly indicate what your variables mean, what they store, and where they're coming from.

One particular caveat to watch out for: some variables in mathematics are true placeholders that don't actually stand for anything. For example, in set-builder notation, we use placeholder variables to denote the name of some unknown quantity:

$$\{ n \in \mathbb{N} \mid n \text{ is even and } n^2 > 48 \}$$

In this context, n does not represent a value. It's just a placeholder so that we can write the expression " n is even and $n^2 > 48$ " in a way that's clear and easy to follow. It's an error to try to reference the number n out of this context.

To see how these rules come into play, let's look at one possible proof of this result:

For any sets A , B , and C , if $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

Here's a not-so-great proof of this result:

Proof: Let A , B , and C be arbitrary sets where $A \subseteq B$ and $B \subseteq C$. This means that for any choice of x , if $x \in A$, then $x \in B$. Similarly, for any choice of x , if $x \in B$, then $x \in C$. We need to prove that $A \subseteq C$, which means that we need to prove that for any choice of x , if $x \in A$, then $x \in C$.

To show this, consider any $x \in A$. Since $x \in A$ and we know that any $x \in A$ must also be an element of B , we see that $x \in B$. Similarly, since $x \in B$ and we know that any $x \in B$ must also be an element of C , we see that $x \in C$, which is what we needed to show. ■

Let's focus on a few of the sentences here. For starters, let's look at this sentence from the opening paragraph:

This $[A \subseteq B]$ means that for any choice of x , if $x \in A$, then $x \in B$.

What, exactly, is the variable x here? It's not an arbitrarily-chosen x , since we didn't say something like "choose an arbitrary x ." Instead, it's a placeholder: it says that *if we find some* x where $x \in A$, then we can conclude that $x \in B$. All that we've done here is set up some possible confusion for later on in the case where we do define some variable named x .

Think back to Rule Three. Every sentence in a proof should set up a goal, introduce a variable, or combine results together into something new. This sentence doesn't set up a goal. It doesn't introduce a new variable. In a sense it *kinda* combines results together into something new, but really, it's not doing that. It's just restating the definition of what a subset is. As a result, this sentence probably fails Rule Three and should be cut.

This sentence actually does cause problems later in the proof, specifically in these sentences:

To show this, consider any $x \in A$. Since $x \in A$ and we know that any $x \in A$ must also be an element of B , we see that $x \in B$.

In the first sentence, we introduce a new variable x , which is chosen as an arbitrary element of the set A (which is fine by both Rule Three and Rule Four). You can imagine that the reader is going to look at this and say “okay, I’m going to pick some specific thing x .” In the next sentence, though, the proof talks about “any $x \in A$.” Now the reader is going to be confused: “hold on, are you talking about the x that you just asked me to pick in the preceding sentence, or are you talking about some other thing called x ?”

Think of it this way: the following code wouldn’t be legal in C, C++, or Java:

```
int x = 137;
int x = 42; // Error!
```

The issue here is that x is already defined on the first line, so the second line is a variable redefinition error. If you want to talk about x going forward, just use its name, not its type:

```
int x = 137;
x = 42; // Okay!
```

The same is true of proofs. Phrases like “any x ,” “every x ,” or “any choice of x ” suggest that you’re introducing some new variable, rather than referring to an existing variable.

A better way to rewrite the above sentences would be to write something like this:

Before	After
To show this, consider any $x \in A$. Since $x \in A$ and we know that any $x \in A$ must also be an element of B , we see that $x \in B$.	To show this, consider any $x \in A$. Since $x \in A$ and $A \subseteq B$, we see that $x \in B$.

Make Specific Claims About Specific Variables

When you're first learning to write proofs, it's common to want to write proofs that make big claims about how things work *in general* rather than pinning down the specifics. For example, consider this not-so-great proof that if $A \subseteq B$ and $B \subseteq C$, then $A \subseteq C$.

Proof: Let A , B , and C be arbitrary sets where $A \subseteq B$ and $B \subseteq C$. We will prove that $A \subseteq C$.

Since $A \subseteq B$, we see that every element of A is an element of B . Similarly, since $B \subseteq C$, we see that every element of B is an element of C . Therefore, every element of A is an element of C , so by definition $A \subseteq C$, as required. ■

The intuition underlying this proof is good, but the way this is written is far too high-level. Specifically, remember that the definition of the statement $A \subseteq C$ is the following:

For every x , if $x \in A$, then $x \in C$.

In order to prove this claim by calling back to the definition, you'd need to show that if you chose an arbitrary element $x \in A$ that you'd find $x \in C$. The proof given above does not do this. The idea behind it – that anything in A is in B and anything in B is in C – is totally correct, but that's not how you'd phrase it in a proof. In proofwriting, if you want to make a claim that something is true in the general case, do so by using arbitrary choices or a proof by contradiction. For example:

Rewrite this...	... like this
Since $A \subseteq B$, every element of A is an element of B .	Consider any element $x \in A$. Since $A \subseteq B$ and $x \in A$, we see that $x \in B$.
The function f maps different inputs to different outputs.	Consider any arbitrary x and y where $x \neq y$. Then $f(x) \neq f(y)$.

When you're reading over your proofs, take a minute to check whether you are making *specific, precise claims* about *named variables* or *broad, general claims* about *all objects of a certain type*. If you find yourself doing the latter, rewrite it to use the former. This will both clarify your reasoning and make it significantly harder to make mistakes. Plus, if you find that you *can't* pin down precisely what you mean about something, it might indicate that there's some concept you're having trouble with.

Write In Complete Sentences and Complete Paragraphs

Although proofs exist to convey mathematical arguments, the expectation is that they should be written in grammatically-correct English sentences and in paragraph form.

A good test we recommend applying to your proofs is what we call the *mugga mugga test*. Take your proof and try reading it out loud, replacing all the mathematical content with the phrase “mugga mugga.” If what comes back is grammatically correct, then you’re on the right track! On the other hand, if what you write is hard to read aloud, or just plain doesn’t sound right, it means that you might need to go back and correct things.

As an example, here’s a not-so-great proof that if n is even, then n^2 is even:

Proof: If n is even, $n = 2k$. $n^2 = 4k^2$ and can be written as $2(2k^2)$. $2k^2 \in \mathbb{Z}$, so n^2 is even. ■

Let’s apply the mugga mugga test to this proof, one sentence at a time. Here’s the first sentence:

Original	Mugga Mugga Version
If n is even, $n = 2k$.	If n is even, <i>mugga mugga</i> .

The mugga-muggaified version of this sentence isn’t grammatically correct – it has no subject and no verb. The reason for this is that the subject of the original sentence is n and the verb is “equals,” but since we’ve written out the equality using the equals sign, it got mugga-muggified in the updated version of the sentence.

More generally:

Tip: Avoid writing sentences where mathematical notation must be treated as a verb.

So what should we do instead? Let’s begin with what you *shouldn’t* do. Don’t rewrite the sentence like this in order to pass the mugga mugga test:

If n is even, n equals $2k$.

This technically passes the mugga mugga test, but it’s doing so by taking a clear mathematical statement ($n = 2k$) and rendering the unambiguous, precise mathematical symbol = in English. The whole reason for having mathematical symbols in the first place is so that we can be precise with our notation, and this is a step in the wrong direction.

Instead, consider rewriting the sentence in a way that introduces a new subject and a new verb. There are many ways that we can do this. Here are a few options to choose from:

New Version	Mugga Mugga Version
If n is even, then we can write $n = 2k$.	If n is even, then we can write <i>mugga mugga</i> .
Since n is even, we see that there is some integer k such that $n = 2k$.	Since n is even, we see that there is some integer k such that <i>mugga mugga</i> .
Because n is even, it can be expressed as $n = 2k$ for some integer k .	Because n is even, it can be expressed as <i>mugga mugga</i> for some integer k .

Notice how in each sentence we've introduced an explicit subject and verb in a way that passes the mugga mugga test.

Let's look at this second sentence:

Original	Mugga Mugga Version
$n^2 = 4k^2$ and can be written as $2(2k^2)$.	<i>Mugga mugga</i> and can be written as <i>mugga mugga</i> .

Again, we're failing the mugga mugga test because the subject and verb of the sentence are expressed in mathematical notation. We'd be better off rewriting this sentence in one of the following ways:

Original	Mugga Mugga Version
We can rewrite the expression $n^2 = 4k^2$ as $n^2 = 2(2k^2)$.	We can rewrite the expression <i>mugga mugga</i> as <i>mugga mugga</i> .
Rewriting $4k^2$ as $2(2k^2)$, we see that $n^2 = 2(2k^2)$.	Rewriting <i>mugga mugga</i> as <i>mugga mugga</i> , we see that <i>mugga mugga</i> .

A common theme in the mugga mugga test is that you should avoid using mathematical notation as the verb in a sentence. Similarly, you should avoid using mathematical notation or shorthands to abbreviate parts of sentences. There are a number of mathematical shorthands that have been developed over the years, primarily for use on blackboards where writing out longhand can take a while. For example, the word "therefore" is often abbreviated \therefore , and the word "because" is often abbreviated \because . These shorthands are just that – they're shorthands – and should not be used in mathematical proofs except if you're trying to write something up quickly and on a board. For example, please, please, please don't write the following:

$\therefore n$ is even, $n = 2k$ for some integer k , $\therefore n^2 = 4k^2 = 2(2k^2)$, $\therefore n^2$ is even $\because n^2 = 2m$ for $m = 2k^2$.

This one really, really, really fails the mugga mugga test:

Original	Mugga Mugga Version
$\therefore n$ is even, $n = 2k$ for some integer k , $\therefore n^2 = 4k^2 = 2(2k^2)$, $\therefore n^2$ is even $\because n^2 = 2m$ for $m = 2k^2$.	<i>Mugga mugga</i> n is even, <i>mugga mugga</i> for some integer k , <i>mugga mugga mugga mugga</i> , <i>mugga mugga</i> n^2 is even <i>mugga mugga mugga mugga</i> for <i>mugga mugga</i> .

This almost reads like a parody of a terrible math lecture. So please don't write proofs like this. ☺

The mugga mugga test won't catch all instances where you're not writing in complete sentences, but it's a great starting point.

Just as you're expected to write in complete sentences, you're expected to write in complete *paragraphs*. This means that your proofs should not consist of bulleted or numbered lists of statements. For example, please don't write proofs like these:

- Let n be an even integer.
- Since n is even, we can write $n = 2k$ for some integer k .
- Then $n^2 = 4k^2$.
- So $n^2 = 2(2k^2)$.
- Let $m = 2k^2$.
- Then $n^2 = 2m$.
- Therefore n^2 is even.

Although we can see what this proof is saying, this just isn't the format that's expected and so you shouldn't structure things this way.