Understanding Proofs

Eklavya Sharma

Abstract

Many people falsely believe that an explanation written in English that attempts to justify their opinions is a proof. Such an explanation may be useful for convincing most people, but technically, it need not be a valid proof. Proofs are useful because they are objective in nature and establish truth beyond any doubt. This article attempts to precisely define what a proof is.

I was a Teaching Assistant (TA) for the 'Design and Analysis of Algorithms' course at CSA, IISc in 2020. While grading homeworks, I was surprised to find out that a lot of students didn't know how to write proofs. Many of them wrote incorrect proofs but strongly believed that their proofs were correct. Hence, they believed that my grading was pedantic or arbitrary. When I talked to them, I realized that the problem was not that people didn't know *how* to prove; the problem was that many of them didn't know *what* exactly a proof is.

Learning to write proofs is tricky. Most students learn proof-writing by looking at examples of proofs. This is what I too initially did during my undergrad. Examples are good for learning proof techniques, like proof by contradiction, proof by induction, proof by cases, etc. But such examples can't be used to truly understand *what* a proof is. Unless people know what a proof is, it would be very difficult for them to know whether their attempted proof is correct or not.

1 Why proofs?

Often in math, we want to ascertain the truth of statements. Let's take a few statements as examples. Do you know which of these are true and which are false?

1. The product of any two odd numbers is odd.
2. Let $f : \mathbb{R} \to \mathbb{R}$ be a function. Let $a, b \in \mathbb{R}$ such that $a < b$, $f(a) < 0$, and $f(b) > 0$. Then $\exists c \in \mathbb{R}$ such that $a < c < b$ and $f(c) = 0$.
3. Every even natural number greater than 2 is the sum of two prime numbers.

Let's say you made up your mind about which of these are true and which are false. How would you convince others that your opinions are correct? In fact, how would *you* know that your opinions are correct? How can you be sure that there's no mistake in your reasoning?

Early in the history of mathematics, many mathematicians made claims about what they thought was true. But some of these claims were later shown to be false. I, too, have at times believed that something was true, only to later find a counterexample and a flaw in my reasoning. This made me very uncomfortable. How can I ever trust myself now? Eventually, mathematicians found a way out of this existential problem. They found a technique that can establish the truth of a statement in a way that leaves no room for doubt. This technique is called *proof*.

Somewhat confusingly, the word 'proof' already exists in the English language, and it's meaning is similar, but not exactly the same, as the word 'proof' in math. Many people falsely believe that an explanation for why they think a statement is true is a mathematical proof if the explanation is detailed enough and written using enough amount of mathematics. But that's not necessarily true. The word 'proof' has a very specific meaning in math.

As an analogy, if you want to explain someone an algorithm, one option is to explain the algorithm using English sentences. But that may be imprecise and ambiguous. On the other hand, if you give them code for the algorithm in a programming language, that would be completely unambiguous. The syntax of the programming language would force you to be unambiguous. Similarly, a proof is a way of establishing the truth of a statement using arguments written in a specific language. Writing arguments in such a language would force you to be unambiguous.

The ability to precisely state something is often taken for granted. A 10-year-old child can probably explain how to sort a list of numbers in ascending order. But when I was new to programming, writing a program to sort a list of numbers was non-trivial for me. Many of my friends had a similar experience. It was difficult not because we didn't know what loops and conditionals were. It was difficult because we weren't used to stating algorithms precisely. Before this exercise, the idea that I don't know how to precisely describe how to sort a list of numbers would have sounded absurd! In this sense, programming was enlightening for me. I think it's the same with writing proofs, i.e., learning to write proofs gives people the ability to precisely reason about the truth of mathematical statements.

If you write an algorithm in a programming language and the compiler successfully compiles you code, you can be sure that there are no syntax errors in your code. Similarly, there is a simple mechanical way to check whether a proof is correct.

In fact, there are computer programs, like Coq, that can validate your proofs if they're written in a certain proof language. If such a program says that your proof is correct, you can be absolutely sure that your proof is correct.

2 What is a proof?

A proof is a *sequence* of *statements*, where each statement is either an assumption or follows from previously-known facts using *rules of inference*. A *rule of inference* is a statement of the form $A_1, A_2, \ldots, A_n \vdash B$, which means that if A_1, A_2, …, A_n are known to be true, then we can infer that B is true.

Let's look at an example of a rule:

$$k \in \mathbb{Z},\ x = 2 \times k + 1 \ \vdash\ \text{isOdd}(x).$$

This rule says that if k is an integer and $x = 2 \times k + 1$, then we can infer that x is odd. We know that $3 \in \mathbb{Z}$ and $7 = 2 \times 3 + 1$. So using this rule, we can infer isOdd(7). In this rule, x and k are *parameters*, i.e., they are treated as placeholders which we can replace by other values. Here we replaced x by 7 and k by 3.

Note that once we assume $3 \in \mathbb{Z}$ and $7 = 2 \times 3 + 1$, we don't need to know what $\in$, $\mathbb{Z}$, $+$, $\times$, $=$, 2, 3, 1, 7 and isOdd mean. The inference procedure is simply symbolic substitution. This is why checking proofs is simple, we just need to ensure that the symbolic substitution was carried out correctly.

I'll now give a few examples of proofs. Note that the way proofs are presented here looks very different from the way proofs are presented in most mathematical texts. I'll comment on this discrepancy after presenting the first example and explain why this difference is merely cosmetic.

3 Example 1: product of odd numbers is odd

We will prove that the product of two odd numbers is odd. To do this, we first need to express this statement precisely:

$$\text{isOdd}(a), \text{isOdd}(b) \vdash \text{isOdd}(ab).$$

In our proof, we will need many definitions and basic facts. For example, the definition of isOdd, the definition of 'logical and' ($\wedge$), the definition of $=$, and some basic facts about arithmetic operators $+$ and $\times$, like closure over integers, associativity, distributivity,

etc. We will capture these definitions and facts using inference rules, called *axioms*. Specifically, axioms are either definitions or they are statements that are so simple and obvious that we assume them to be true without proof. In this example, instead of stating beforehand all the axioms that we will use in the proof, we will introduce them when we need them.

Axioms for isOdd:

- `Odd1`: $x = 2k + 1, k \in \mathbb{Z} \vdash \text{isOdd}(x)$.
- `Odd2`: $\text{isOdd}(x) \vdash \exists k(x = 2k + 1 \wedge k \in \mathbb{Z})$.

Here `Odd1` and `Odd2` are the names of the two axioms.

Here $\wedge$ means 'logical and'. Note that in axiom `Odd2`, x is a parameter but k is not, since it is *bound to* the '$\exists$' symbol.

Next, we have an inference rule about $\exists$: Let $\phi(k)$ be an expression containing a variable k. Then from $\exists k \phi(k)$ we can infer $\phi(r)$, where r is a fresh variable, i.e., a variable that hasn't been used so far in the proof. We call this rule $\exists$-elimination.

Let's start the proof now.

Proof of $\text{isOdd}(a), \text{isOdd}(b) \vdash \text{isOdd}(ab)$**:**

1: $\text{isOdd}(a)$ *// given*

2: $\text{isOdd}(b)$ *// given*

These statements follow from the left half of the result that we want to prove, i.e., $\text{isOdd}(a), \text{isOdd}(b) \vdash \text{isOdd}(ab)$. The text after '`//`' are comments, i.e., not part of the statements.

3: $a = 2r + 1 \wedge r \in \mathbb{Z}$ *// from `Odd2`, 1, $\exists$-elimination*

4: $b = 2s + 1 \wedge s \in \mathbb{Z}$ *// from `Odd2`, 2, $\exists$-elimination*

Axioms for $\wedge$:

- $\wedge$E1: $A \wedge B \vdash A$.
- $\wedge$E2: $A \wedge B \vdash B$.

5: $a = 2r + 1$ *// 3, $\wedge$E1*

6: $r \in \mathbb{Z}$ *// 3, $\wedge$E2*

7: $b = 2s + 1$ *// 4, $\wedge$E1*

8: $s \in \mathbb{Z}$ *// 4, $\wedge$E2*

Axioms for $=$:

- `id=`: $a = a$.
- `repl=`: Let $\phi(x)$ be a predicate containing variable x. Then $a = b, \phi(a) \vdash \phi(b)$.

(Recall that rules of inference are statements of the form $A_1, A_2, \ldots, A_n \vdash B$. But when $n = 0$, we simply write the rule as B. `id=` is one such rule.)

Using `repl=`, we can prove symmetry of $=$, i.e., $a = b \vdash b = a$ (use $\phi(x) : x = a$). Let's name this result `symm=`. Using `repl=`, we can prove transitivity of $=$, i.e., $a = b, b = c \vdash a = c$ (use $b = c$ and $\phi(x) : a = x$; see Lemma 2 in Appendix A for details). Let's name this result `trn=`. Using `repl=` twice, we can prove that $a = b, c = d \vdash ac = bd$ (first use $\phi(x) : ac = xc$ and $a = b$, then use $\phi(x) : ac = bx$ and $c = d$). Let's name this result `mult=`. (See Lemma 3 in Appendix A for a similar proof.)

9: $ab = (2r + 1)(2s + 1)$ *// mult=, 5, 7*

Axioms about arithmetic:

- Distributivity of $\times$ over $+$: $a(b + c) = ab + ac$, $(a + b)c = ac + bc$.
- Multiplicative identity: $a \times 1 = a$, $1 \times a = a$.
- Associativity of $\times$: $(ab)c = a(bc)$.
- Associativity of $+$: $a + (b + c) = (a + b) + c$.

Using the above axioms for arithmetic with `repl=` and `symm=`, we can prove that for any r and s, we get $(2r+1)(2s+1) = 2(r(2s+1)+s)+1$ (see Theorem 4 in Appendix A for details).

10: $(2r + 1)(2s + 1) = 2(r(2s + 1) + s) + 1$. *// Theorem 4 in Appendix A*
11: $ab = 2(r(2s + 1) + s) + 1$. *// 9, 10, trn=*

Axioms for $\mathbb{Z}$ membership:

- $1 \in \mathbb{Z}$, $2 \in \mathbb{Z}$.
- $a \in \mathbb{Z}, b \in \mathbb{Z} \vdash a + b \in \mathbb{Z}$.
- $a \in \mathbb{Z}, b \in \mathbb{Z} \vdash ab \in \mathbb{Z}$.

12: $r(2s + 1) + s \in \mathbb{Z}$. *// 6, 8, axioms for $\mathbb{Z}$ membership*
13: $\text{isOdd}(ab)$ *// 11, 12, Odd1*

This completes the proof of $\text{isOdd}(a), \text{isOdd}(b) \vdash \text{isOdd}(ab)$.

Note how we only used symbolic substitution in the proof. So if you believe in the axioms above and if you verify that the symbolic substitution was performed correctly, then the proof is irrefutable. All the axioms above are well-known facts (some can even be found in elementary-school textbooks), so almost everyone would believe them. Proofs written like this are, therefore, the gold standard in establishing truth.

But this isn't how proofs are written in most mathematical texts, and for good reason: writing proofs this way is cumbersome. A more common way of writing the above proof is as follows.

Lemma 1. *If a and b are odd numbers, then their product ab is also odd.*

Proof. Since a is odd, $a = 2r + 1$ for some $r \in \mathbb{Z}$. Similarly, $b = 2s + 1$ for some $s \in \mathbb{Z}$.

Therefore, $ab = (2r+1)(2s+1) = 2(r(2s+1)+s)+1$. Since $r(2s+1)+s \in \mathbb{Z}$, we get that ab is odd. □

A proof written in this way is called an *informal* proof. On the other hand, the proof above that used symbolic substitution is called a *formal* proof.

Note the similarity between the informal and formal proofs. In the informal version, we skipped some of the simple intermediate results, but the core idea is the same as that in the formal version. It is the formal proof that ultimately establishes truth; an informal proof is just a convenient tool to convince others of the existence of a formal proof without the hassle of writing down the full formal proof.

Although you will probably not need to write formal proofs, it's instructive to have a rough idea of the formal version in mind when writing informal proofs. If you're ever unsure of whether your informal proof is correct, try to see if you can get a formal proof. Once you convince yourself that a formal proof exists, you'll be confident about your proof's correctness.

Disclaimer: the axiom names above are probably non-standard. Some of the axioms above are actually not axioms in standard systems of logic, since they can be derived from even more primitive axioms. The words *informal* and *formal* as defined above are also not standard.

4 Example 2: De Morgan's theorem

TODO: introduce propositional logic, most of its axioms and proof by contradiction.

- A list of propositional logic axioms.
- A list of derived rules.

5 Example 3: $|E| \geq |V| - 1$ for a simple undirected connected graph (V, E)

TODO: axiomatize some graph-theoretic definitions and introduce proof by induction.

TODO: maybe add more examples from other areas, like number theory, linear algebra, probability.

TODO: talk about soundness and completeness. Link to more resources on logic for those who are interested.

Acknowledgements

I attended Prof. Shan Sundar Balasubramaniam's enlightening course on 'Logic in Computer Science' at BITS Pilani during my undergrad years. That's where I first learned about the true meaning of proofs.

A Miscellaneous Proofs

Lemma 2 (`trn=`). $a = b, b = c \vdash a = c$.

Proof.

1:	$a = b$	// *given*
2:	$b = c$	// *given*
3:	$a = c$	// *1, 2,* `repl=` *with* $\phi(x) : a = x$.

□

Lemma 3 (`add=`). $a = b, c = d \vdash a + c = b + d$.

Proof.

1:	$a = b$	// *given*
2:	$a + c = b + c$	// *1,* `repl=` *with* $\phi(x) : a + c = x + c$
3:	$c = d$	// *given*
4:	$a + c = b + d$	// *3, 2,* `repl=` *with* $\phi(x) : a + c = b + x$

□

Theorem 4. $(2r + 1)(2s + 1) = 2(r(2s + 1) + s) + 1$

Proof.

1:	$(2r + 1)(2s + 1) = (2r)(2s + 1) + 1 \times (2s + 1)$	// *distributivity of* $\times$ *over* $+$
2:	$(2r)(2s + 1) = 2(r(2s + 1))$	// *associativity of* $\times$
3:	$1 \times (2s + 1) = 2s + 1$	// *Multiplicative identity*
4:	$(2r)(2s + 1) + 1 \times (2s + 1) = 2(r(2s + 1)) + 2s + 1$	// *2, 3,* `add=`
5:	$(2r + 1)(2s + 1) = 2(r(2s + 1)) + 2s + 1$	// *1, 4,* `trn=`
6:	$2(r(2s + 1) + s) = 2(r(2s + 1)) + 2s$	// *distributivity of* $\times$ *over* $+$
7:	$2(r(2s + 1)) + 2s = 2(r(2s + 1) + s)$	// *6,* `symm=`
8:	$2(r(2s + 1)) + 2s + 1 = 2(r(2s + 1) + s) + 1$	// *7,* $1 = 1$, `add=`
9:	$(2r + 1)(2s + 1) = 2(r(2s + 1) + s) + 1$	// *5, 8,* `trn=`

□