Lecture 4: Proofs

In this lecture and the following one, we look at mathematical proofs.

What is a Proof?

A proof plays the same role in mathematics, more or less, as *source code* does in computer programming. A valid source code file is one that can be compiled or interpreted and run by a computer, without syntax errors. Ideally it even does what it is supposed to do, when you run it. But, source code exists to be written and read by humans — computers themselves would be more than happy if programs were just written in binary. Thus, among programs that compile and run correctly, the one that is more human-readable is the better one. If we had to rank some student submissions for a programming assignment:

1. One student submits a source code file that does not compile, due to syntax errors. They would get a low mark, most likely a fail.
2. One student submits a program that compiles, but does not do the expected thing when you run it. They might just about get a pass, possibly a 3rd class mark.
3. One student submits a program that compiles, does the correct thing when you run it, but the code is not elegant and hard to read. They might get a 2nd class mark.
4. One student submits a program that not only compiles and does the correct thing when you run it, but the code (and comments) are elegant and easy for the marker to read. This student might get a first-class mark.

If two students both submit code that does not compile, but one submits more elegant non-compiling code than the other, this does not matter quite as much: both these students are at risk of failing. But once your code compiles and runs correctly, the easier to read, the better. And so it is with mathematical proofs: you should aim to make your proofs

1. Correct, because if your work is not correct, it does not count as a proof at all.
2. Without compromising on 1., as easy for a human to understand as possible.

To be pedantic, there is no such thing as a 'correct proof' and an 'incorrect proof': the kind of thing that could or could not be a proof, depending on if it is correct or not, is called an *argument* and a *proof* is simply an argument that is correct. So the following two questions mean the same thing:

- Is this a valid argument? (or: 'a correct argument')
- Is this a proof?

It does not make sense to ask 'Is this a correct proof?' because if an argument is not correct, then it is not a proof.

So what *is* a proof?

A mathematical proof is a structured argument in a semi-formal language where every step meets two conditions:

1. Every step follows logically from the previous steps.
2. Every step is justified.

Unlike computer programs where it is usually clear what is or is not a syntax error (either the compiler accepts your code, or not), the ‘compiler’ for a mathematical argument is other mathematicians. One would hope that if you gave ten mathematicians the same argument, they would all come up with the same answer as to whether it is valid or not. But since mathematical arguments are not defined by a formal grammar like programming languages, we say they are written in a semi-formal language.

There are in fact fully formalised proof languages that can be checked by a computer, such as Coq[1] but these require more skill to use than ‘ordinary’ proofs and are beyond the scope of this unit.

Since mathematical arguments are both written and read by humans, a certain amount of common sense and imprecision often creeps in, and this is not a problem as long as it does not get in the way of the correctness of an argument. Proofs written by professional mathematicians will feature justifications like ‘it is obvious that’ or they will prove one case and say ‘the other three follow similarly’, and this is not a problem for the working mathematician as long as the other cases do actually follow! But when you are learning to write proofs for the first time, you should avoid these terms as far as possible, to avoid you claiming something is ‘obvious’ when it is in fact false.

Since proofs are written for humans to read, they will often contain text that is not part of the formal argument itself, but still helps the reader: just like a report might contain headings and subheadings, and a table of contents, a proof might contain a line ‘We prove this claim by induction. The base case is . . . The induction step is . . . ’ You should always include structure like this in your own proofs.

There is no difference in formality or correctness between an argument structured in English text, and an argument written in symbols. The following two examples are exactly the same as far as correctness goes:

- We know A is true, and we know that B would imply that A is false, therefore we conclude that B is false too using ‘modus tollens’.
- $A \wedge (B \to \neg A) \vDash \neg B \because$ (M.T.)

The difference between the two is a matter of style, and you can choose between them and even vary between both in the same proof as long as the conclusions you draw are correct (follow logically from previous steps) and justified. So please do not think that to look good, a proof has to have lots of symbols and almost no English text.

We will look at a way one could model proofs as a data structure a bit later on in these notes. These data structures could then be printed using English text or symbols, depending on your preference, just like arithmetic terms can be printed using infix or prefix or RPN.

Finally, note that while checking the correctness of an argument is in principle an algorithmic task (one could program a computer to do it, if the argument is in a formal enough language) just like checking if source code compiles is an algorithmic task in practice (we can run the compiler on it), writing both arguments and source code is at heart a *creative* task. Even leaving questions of elegance and style aside, given a high-leveel description of a mathematical or coding problem, finding the right argument steps or lines of code in the first place takes intuition and experience, and is not just a case of following a list of rules.

[1] https://coq.inria.fr

Odd and Even

Before we can give examples of proofs, we need something to do proofs on. We will look at arithmetic over the integers, and specifically the fact that all integers are either odd or even. We define this a bit more formally.

Definition 1. An integer a is

- **even**, if we can write $a = 2k$ for some integer k.
- **odd**, if we can write $a = 2k + 1$ for some integer k.

Every integer matches exactly one of these two cases (we will prove this later on based on more basic rules).

The wording 'for some integer' *introduces* a new variable. We can do this at any time in a proof that we like, but every time we do this, we must pick a new variable that does not exist in the proof yet (at least not in the current 'scope').

Rules of Arithmetic

To do proofs in arithmetic, we need a set of rules that we can use as steps in the proof. Since there are infinitely many integers, we cannot check all possibilities with a truth table, so we will have to do proofs with syntax rather than semantics. The rules here are all sound, but not complete (there are true statements that we cannot prove with these rules); this is not usually a problem in practice.

All the following rules apply for natural numbers, integers and real numbers unless otherwise stated.

- Commutative laws: $a + b = b + a$ and $a \times b = b \times a$ for any numbers a, b.
- Associative laws: $(a + b) + c = a + (b + c)$ for any numbers a, b, c, and the same for the $\times$ operation.
- Distributive law: $a \times (b + c) = a \times b + a \times c$ for any numbers a, b, c.
- Neutral elements: $a + 0 = a$ and $a \times 1 = a$ for any number a.
- Inverse elements: over the integers and reals but not the naturals, $a + (-a) = 0$ and, over the real numbers only, if $a \neq 0$ then $a \times 1/a = 1$. More precisely, for any number a we can find another number b such that $a + b = 0$ (namely, set $b = (-a)$) if we are working over the integers or reals, and if $a \neq 0$ then we can find a b such that $ab = 1$ if we are working over the reals.
- Multiplication and zero: $a \times 0 = 0$ for any number a. More interestingly, if $a \times b = 0$ then at least one of a, b must already be zero; another way of putting this is that if $a \neq 0$ and $b \neq 0$ then also $a \times b \neq 0$. This is the rule you need to deduce that the solutions to $(x - 2)(x - 3) = 0$ are exactly 2 and 3, for example. (Note, this rule is not true for numbers that 'wrap around' such as 64-bit unsigned integers on a computer, where for example $2^{63} \times 2$ becomes 0 again.)

From basic rules like these we can prove many others, such as that $(a + b) \times (a + b) = a \times a + 2 \times a \times b + b \times b$, which one obviously writes as $(a + b)^2 = a^2 + 2ab + b^2$.

In general, you can use any rule of arithmetic that you have learnt in school in a proof, as long as it is correct for the numbers you are using. For example, if you are doing a proof over the integers, then be careful of dividing: not only do you have to make sure you do not divide by zero, but the result might not be an integer anymore, and so might not be odd or even.

Proof Strategies, Part 1

What kind of steps can we take in a proof?

Suppose that we are trying to prove the claim ‘if a is an even integer and b is an odd integer, then ab is even’.

At any time in a proof, you have a set of statements that you know are true. This might be because they are part of what you are given at the start: in the example we know that a is even. It might be because we have already proved them earlier. It might also be because a statement is a generally known law of arithmetic or logic.

Many of the steps we take in a proof are applying rules to statements we know are true, resulting in new statements that we now also know are true. If we manage to add the statement we are trying to prove to the set of things we know are true, we are done.

Other steps are ‘unpacking’ and ’packing’ defintions. For a mathematician, ‘even’ and ‘odd’ are definitions with precise formal meanings as we showed above. Our intuition about how even and odd numbers work can guide us, but in a proof the only thing you are allowed to do with the information ‘a is even’ is unpack the definition of ‘even’. Generally a good proof strategy is to unpack the definitions you are given, calculate, then pack the result back into the definition you need. For our example proof, this means

1. Unpack the definitions of ‘a is even’ and ‘b is odd’.
2. Calculate with the results of the above.
3. At some point, pack what you get back into the definition ‘ab is even’.

Direct Proof

In a direct proof, you just go ahead and calculate. If you can do a direct proof of a statement, it is normally the best option.

Claim: if a is even and b is odd, then ab is even.
Proof:

1. We know that a is even. Unpack this to get that there is an integer k such that $a = 2k$.
2. We know that b is odd. Unpack this to get that there is an integer m such that $b = 2m + 1$ (note, k is already taken, so we need a new variable).
3. Calculate: $ab = (2k)(2m + 1) = 4km + 2k = 2(2km + k)$.
4. If we set $c = 2km + k$ then we have $ab = 2c$ where c is an integer. Pack this to get: ab is even.

It is fine to write out proofs as paragraphs rather than lists, so the following is just as good a proof (or perhaps even better):

Since a is even, there is an integer k such that $a = 2k$, and since b is odd, there is an integer m such that $b = 2m + 1$. Then $ab = (2k)(2m + 1) = 2(2km + k)$. Therefore, ab is even.

Claim: if n is even, then n^2 is even.
Proof: since n is even, there is some k such that $n = 2k$. Then $n^2 = (2k)^2 = 2(2k^2)$, so n^2 is even.

Indirect Proof

Both the above proofs were proofs of implications $P \to Q$. In a direct proof, we start by knowing P is true, we calculate, and we get to knowing Q is true.

Indirect proofs use that $P \to Q$ is equivalent to its contrapositive $\neg Q \to \neg P$, so we can start by assuming $\neg Q$ and calculate, and if we get to $\neg P$ then we have also proved $P \to Q$. We would usually take this strategy if $\neg Q$ looks like an easier statement to start calculating with.

Claim: if n^2 is odd, then n is odd.
Proof: here we do not want to start with n^2 as the main thing you can do with a squared number is take the square root, but that might not be an integer. But if we start with n and square it, that remains an integer. So, assume that n is even (not odd, we are proving the contrapositive so we have to negate the statement we are after). Then $n = 2k$ for some k. Then $n^2 = (2k^2) = 2(2k^2)$ which is even again. So, if n is even then n^2 is even, and by indirect proof it follows that if n^2 is not even then n is not even either — and 'not even' is the same as 'odd'.

Claim: if ab is even, then a is even or b is even.
Proof: the contrapositive here is 'if a is odd and b is odd, then ab is odd'. Indeed, if a is odd then write $a = 2k+1$ and if b is odd then write $b = 2m+1$, then $ab = (2k+1)(2m+1) = 2(2km+k+m)+1$ which is odd again.

Proof by Contradiction

The idea here is that if we know, for some statement S, that $S \to F$ is true, then S must itself be false, since $T \to F$ would not be true but $F \to F$ is true.

So, if we want to prove that P is true, then we can try and prove that $\neg P$ is false, which is an equivalent statement. The way a proof by contradiction works is we assume $\neg P$, calculate, and if we reach a contradiction then $\neg P$ must have been false all along.

Claim: $\sqrt{2}$ is irrational.
Proof: first, we unpack the claim. A rational number is a number q that can be written as a fraction a/b where a, b are integers, and $b \neq 0$. If these a, b have a common factor, then we can cancel it on both sides of the fraction, so every rational number can be written as a fraction a/b where a, b have no factor (greater than 1) in common. An irrational number is a number that is not a rational number.

Assume that $\sqrt{2}$ is a rational number, that is $\sqrt{2} = A/B$ with A, B integers and $B \neq 0$. Simplify this fraction by removing all common factors to get a/b.

Squaring both sides implies that $2 = a^2/b^2$ which we can rewrite as $2b^2 = a^2$ since $b \neq 0$. Since $2b^2 = a^2$ is an equation over the integers, this means that a^2 must be even.

We know from earlier that if a is odd then a^2 is odd, so by the contrapositive, if a^2 is even then a is even too and we can write $a = 2k$.

Plugging this back in, we get $2b^2 = (2k)^2 = 4k^2$ and we can cancel a 2 on both sides to get $b^2 = 2k^2$. This means that b^2 must be even too, and therefore b must be even.

But this is impossible. We have shown that for any integers a, b with $b \neq 0$ such that $(a/b)^2 = 2$, both a and b must be even. However, we know that every fraction a/b can be simplified until a, b have no common factors, in particular they do not have a factor 2 in common. This is a contradiction.

Therefore, the assumption that we can find integers A, B with $A/B = \sqrt{2}$ in the first place must have been wrong.

Claim: if ab is odd, then a is odd or b is odd.
Proof: the claim we are proving has the structure $P \to Q$. We want to show that $\neg(P \to Q)$ leads to a contradiction, in which case $P \to Q$ itself must be true. The term $\neg(P \to Q)$ is equivalent to $P \wedge \neg Q$, so we want to assume that P: 'ab is odd' is true and Q: 'a is odd or b is odd' is false, and find a contradiction.

Suppose that ab is odd, but a and b are both even. Then $a = 2k$ and $b = 2m$ for some integers k, m. But then $ab = 4km = 2(2km)$ is even too, which contradicts the assumption that ab is odd. Therefore, if ab is odd, then a and b cannot both be even.

(In fact, if ab is odd, then neither a nor b can be even, but that is not what we were trying to prove here.)

That was quite a mouthful! As a general rule, direct and indirect proofs are about equally hard to read, but proofs by contradiction can take extra effort to understand. So, it is good style not to do a proof by contradiction when you could equally well do a direct or indirect proof. The first example ($\sqrt{2}$ is irrational) would be much harder to do without contradiction, so that is a good use of this technique, but the second example could be done more simply as an indirect proof, with the same key calculation that $(2k)(2m)$ must be even again.

Case Distinction

Often, in a proof, we want to say something about 'all x'. If we can split all x into several cases, such that every x fits in at least one (or exactly one) of the cases, then we can do the proof in each case individually.

Claim: $n(n+1)$ is always even.
Proof: consider the two cases when n is even and n is odd, since every integer fits one of these cases.

- When n is even, then $n = 2k$ for some k, and $n(n+1) = (2k)(2k+1) = 2(2k^2 + k)$ which is even again.
- When n is odd, then $n = 2k+1$ for some k, and $n(n+1) = (2k+1)(2k+2) = 2(2k^2 + 3k + 1)$ which is even.

Therefore, whether n is odd or even (and there are no other cases), $n(n+1)$ is always even.

Claim: if $a + b$ is even, then either both a, b are even, or both odd.
Proof: Assume that $a + b$ is even, then $a + b = 2k$ for some k. Consider two cases:

- a is even, that is $a = 2m$ for some m. Then $b = (a+b) - a = 2k - 2m = 2(k-m)$ which is even too.
- a is odd, that is $a = 2m+1$ for some m. Then $b = (a+b) - a = 2k - (2m+1) = 2(k-m) - 1 = 2(k-m-1) + 1$. which is odd too.

In both cases (again every a fits at least one case — in fact exactly one), a and b are both the same parity (both even, or both odd).

Euclid's Theorem

Now, we prove that integers are either odd or even according to the earlier defintion, from more basic principles.

Definition 2 (Euclid's theorem)**.** For two integers a, b with $b > 0$, there is exactly one pair of integers q, r such that $0 \leq r < b$ and

$$a = q \times b + r$$

We call q the quotient and r the remainder of dividing a by b with remainder.

For example, if $a = 17, b = 3$ then a/b with remainder is 'quotient 5 remainder 2' since $17 = 5 \times 3 + 2$.

First, let us prove that integers are either odd or even as defined before. Assuming this theorem holds (we will prove part of it today and the rest later), take $b = 2$. Then for any integer a, there is an integer q and an integer $r \in \{0, 1\}$ such that $a = 2q + r$, that is either $a = 2q$ or $a = 2q + 1$. The two cases are mutually exclusive, since if we had both $a = 2q$ and $a = 2q' + 1$ then $(q, 0)$ and $(q', 1)$ would both be pairs satisfying the theorem, but the theorem says there is exactly one such pair, which would be a contradiction.

The proof of Euclid's theorem itself comes in two parts: first, for any integers a, b with $b > 0$ we have at most one pair of numbers (q, r) satisfying the theorem, and secondly, that we have at least one such pair. We will do the first part today and the second in a later lecture, as we need a new technique for it.

A proof that *at most one* object X exists with some property is called a *uniqueness* proof, and there are two main ways to do it: either start assuming that X, X' both have the property and deduce that $X = X'$, or start assuming that X, X' are two different objects with the property, and show that this leads to a contradiction. A proof that *at least one* object exists with some property is called an *existence* proof, and the best way to do such a proof (if you can) is to give some algorithm that creates the object you need.

Claim. For any integers a, b with $b > 0$, there is at most one pair of integers (q, r) with $a = qb + r$ and $r \geq 0$ and $r < b$.
Proof. Suppose we had two such pairs (q, r) and (q', r'), which might or might not be the same. Then we know all of

- a, b, q, r, q', r' are all integers.
- $b > 0$
- $a = qb + r$
- $a = q'b + r'$
- $0 \leq r < b$
- $0 \leq r' < b$

We subtract the two equations for a and rearrange to get

$$b(q - q') = r' - r$$

We now make a case distinction on $r' - r$.

If $r' - r = 0$ then $b(q - q') = 0$, but we know $b \neq 0$, so we must have $q - q' = 0$. Then $q = q'$ and $r = r'$ so the two pairs are the same, as the theorem demands.

If $r' - r > 0$ then we show a contradiction. There are three subcases:

- If $q - q' = 0$ then $r' - r = 0$ too which is a contradiction to $r' - r > 0$.
- If $q - q' < 0$ then $b(q - q') < 0$ as $b > 0$, and this is a contradiction to $r' - r > 0$.
- If $q - q' > 0$ then $q - q'$ is at least 1, as it is an integer. Therefore, $b(q - q') \geq b$. But, since $r < b$ and $r' \geq 0$, then $r' - r < b$ which is a contradiction.

Either way, $r' - r > 0$ leads to a contradiction.

If $r' - r < 0$ then we do the same argument as above, but with the roles of r and r', and q and q' swapped.

This proves the uniqueness part of the theorem, and so we know that no integer can be both odd and even at the same time.