Programming Mathematical Proofs

Steven Ramsay

Contents

1 Proof as a form of programming

Proofs are semi-formal arguments that make a case for the truth of some precise statement. The key word as far as proofs are concerned is *convincing*: a proof is a way of communicating your belief that something is true in a way that will convince another party.

Some proofs are more convincing than others. In mathematics, the statements that we want to give proofs for only have certain shapes — they are built from a few, standard logical operators. Over the years, mathematicians have developed equally standard approaches to dealing with the logical operators in proofs. By following these standard approaches, the benefit is that you will end up with a proof that will be convincing. Proofs of this kind are typically blocks of English text with a scattering of mathematical formulas.

It can be very helpful to think of the individual steps of a proof as being operations that manipulate the *proof state*. The proof state is a pair consisting of a set of *assumptions* and the current *goal*. The goal is the formula you are currently trying to prove and the assumptions are the resources that you have available in order to try to prove it. Each step of the proof (i.e. each application of one of the rules of natural deduction, signalled using an appropriate form of words) changes the proof state by either adding to the set of assumptions or changing the goal. For example, if we have assumptions $A_1, \ldots, A_k$ and we are aiming to prove $A \Rightarrow B$, then the magic words "assume A", which is a form of words used to signal a use of implication introduction, lead us to a proof state in which the assumptions are $A_1, \ldots, A_k, A$ and the goal is B. When you are reading a proof, it is very helpful to picture the proof state in your mind after each step of the proof.

The following example is a proof of a very simple fact from number theory: if two natural numbers n and m add up to zero, then m is not of the form $k + 1$ for any possible k.

Lemma 1. *For all $n, m, k \in \mathbb{N}$: if $n + m = 0$ then $m \neq k + 1$.*

Proof. Let n, m and k be natural numbers. Assume $n+m = 0$. To see that $m \neq k+1$, we assume $m = k+1$ and try to obtain a contradiction. Combining these assumptions we get $n+(k+1) = 0$. By the associativity of addition this is the same as $(n + k) + 1 = 0$. Since $n + k$ is a natural number, we have $(n + k) + 1 \neq 0$. We have reached our contradiction. □

Suppose someone gives you a piece of straightline C code (no loops) for computing some complicated function $f(x, y)$ of a pair of integers variables x and y, and asks you to check that it works. To understand what the code is doing, most likely you will start at the beginning of the code, read each statement in turn and keep track of the values of x and y in your head[1] as they are manipulated by the code. By the end, you should be able to tell whether or not the final value returned is $f(x, y)$ (for the original values of x and y) or not. Of course, to do this, you need to understand the "rules" of the C language, i.e. how the syntax fits together and how each statement affects the current state, i.e. the current values of x and y.

The same process is required to understand a proof. You will go through, statement by statement, keeping track of the proof state until, by the end, it should be clear whether or not the proof works, i.e. in the final proof state, the goal is something that is already known. Of course, to do this you will need to know the "rules" of mathematical proof. Giving you those rules is the purpose of the following section, but I think it is useful to go through this example first anyway so you can see what I mean.

One thing that we need to make clear before that, however, is what the starting point for the proof is.

In "real life" mathematicians are trying to prove statements that nobody has ever proven before (and may turn out to be false). To give themselves the best chance to prove their statement, they will employ all the mathematical knowledge that they have at their disposal: countless theorems already proven by the mathematicians that came before them, or that they proved during their career. In other words, they start the proof with a set of assumptions that is enormous (but they know each is justified by a proof, somewhere).

When learning mathematics, however, we are usually proving quite simple statements that someone else has already proven a long time ago. Hence, it can be a bit confusing to the student to know which already proven statements they are allowed to assume and which they are required to prove first. This will not be a problem during *Types and Lambda Calculus* because (I guess) you don't know any basic theorems of this theory, so we will just agree that you can assume only those statements that we have proven so far during the course of the unit (and anything you know about sets and strings).

The example above, however, is in number theory, and you probably already have a good idea of what simple statements are true about natural numbers and which are not. So, for this example, I will be very clear about what we are allowing ourselves to assume at the start of the proof:

(A) for all $p \in \mathbb{N}$: $p + 1 \neq 0$

(B) for all p, q and r in $\mathbb{N}$: $p + (q + r) = (p + q) + r$

(C) for all $p, q \in \mathbb{N}$: $p + q \in \mathbb{N}$

[1] If you are not so experienced with C, or the code is very complicated, then it may be useful to write down the evolving values of x and y on a scrap of paper.

Proof step	Assumptions	Goal
	A, B, C	$\forall nmk \in \mathbb{N}.\ n+m=0 \Rightarrow m \neq k+1$
Let n, m and k be natural numbers.	A, B, C, $n, m, k \in \mathbb{N}$	$n+m=0 \Rightarrow m \neq k+1$
Assume $n+m=0$.	A, B, C, $n, m, k \in \mathbb{N}$, $n+m=0$	$m \neq k+1$
To see that $m \neq k+1$, we assume $m=k+1$ and try to obtain a contradition.	A, B, C, $n, m, k \in \mathbb{N}$, $n+m=0$, $m=k+1$	$\perp$
Combining these we get $n+(k+1)=0$	A, B, C, $n, m, k \in \mathbb{N}$, $n+m=0$, $m=k+1$, $n+(k+1)=0$	$\perp$
By the associativity of addition, this is the same as $(n+k)+1=0$,	A, B, C, $n, m, k \in \mathbb{N}$, $n+m=0$, $m=k+1$, $n+(k+1)=0$, $(n+k)+1=0$	$\perp$
Since $n+1$ is a natural number, we have $(n+k)+1 \neq 0$.	A, B, C, $n, m, k \in \mathbb{N}$, $n+m=0$, $m=k+1$, $n+(k+1)=0$, $(n+k)+1=0$, $(n+k)+1 \neq 0$	$\perp$
but this is a contradiction.	A, B, C, $n, m, k \in \mathbb{N}$, $n+m=0$, $m=k+1$, $n+(k+1)=0$, $(n+k)+1=0$, $(n+k)+1 \neq 0$, $\perp$	$\perp$

Table 1: Trace of the proof of $\forall nmk \in \mathbb{N}.\ n+m=0 \Rightarrow m \neq k+1$

Statement B is what is referred to in the proof as “associativity of addition”. Statements A and C are used in the last step, but are not referred to by any name. This probably means the author of the proof believes them to be so simple that they are not worth mentioning. Now let’s analyse the proof step by step starting in the proof state containing only these assumptions A, B and C. A trace of the proof is given in Table 1, whose final two columns shows the proof state after executing each of the statements.

A few things about the general shape of this proof are worth observing.

- First, the proof appears to prove the statement, because in the final proof state the goal is already contained in the assumptions. However, we can’t know that it is really a valid proof unless we check that each of the proof steps really transforms the proof state before-it into the proof state after-it, and this requires a knowledge of the rules.

- The set of assumptions grows larger over the course of the proof.

- The goal starts off quite large and complicated, but is progressively simplified over the first few proof steps and, once it is as simple as possible, it doesn’t change from that point on.

Backward Reasoning Steps 1–3 in which the goal gets simpler are called *backwards reasoning*. It is quite typical that a proof starts by performing backward reasoning. Backwards reasoning steps are of the form “To prove a goal of shape A it is enough to prove the simpler goal(s) $B_1, \ldots, B_n$”. In the case of the example, the particular backwards steps that are employed only generate a single simple goal. For example, step 2 uses the backwards reasoning rule for implication, which says “to prove A implies B, it is enough to assume A is true and then try to prove B”. Backwards reasoning is very easy because the backwards reasoning steps you are allowed to make mostly just depend upon the syntactic shape of the goal: is it a conjunction, an implication, a for all, etc. Often, but not always, backwards reasoning not only simplifies the goal, but also adds new assumptions (which is helpful for fowards reasoning).

Forwards Reasoning Steps 4–7 are *forwards reasoning* steps. Forwards reasoning consists of those proof steps that do not change the goal, but simply deduce more and more assumptions (facts) by combining existing ones. The majority of a non-trivial proof is spent on forwards reasoning. Forwards reasoning is a bit more difficult only because you can build up a large set of assumptions and spotting which ones can be fruitfully combined to deduce new ones *that are actually relevant to proving your goal* requires some insight.

2 Rules of the game

In this section are listed all the rules that suffice for any mathematical proof. With one important exception, they are organised by logical connective. For each logical connective there is a list of the forwards and backwards rules for that connective: the backwards rules tell you how you can simplify a goal which is built from that connective at the outer level, the forwards rules tell you how you can deduce new assumptions from an existing one built from that connective at the outer level.

Each rule is given in form of a table entry of shape:

Assms	Goal

“blah blah”

Assms	Goal

This tells you the syntax of the proof rule, e.g. the words “blah blah”, and the semantics, i.e. how it changes the proof state. For a proof to be valid, the commands must hang together as a correct program, so you must ensure that you only use a rule when the proof state before has the correct shape. For example, the backwards reasoning rule for implication requires that the goal is literally an implication, i.e. has the shape $A \Rightarrow B$. It does not apply when the goal has shape, e.g. $\forall x \in X.\, A \Rightarrow B$, which is a for all (whose body is an implication), nor when the goal has hape $(A \Rightarrow B) \wedge C$, which is a conjunction (whose right conjunct is an implication). The exact form of words is not important, as long as it is clear from the context (i.e. the proof state) which rule are you trying to invoke.

2.1 Implication

$A \Rightarrow B$ A implies B If A then B. Suppose A, then B.

Backward Rule

To prove $A \Rightarrow B$ starting from some assumptions, it suffices to instead prove B with A added to the assumptions.

Assms	Goal
$\ldots$	$A \Rightarrow B$

“Assume A. We prove B as follows...”

“To show $A \Rightarrow B$, we assume A. ...”

“Assume A. ...”

“Suppose A is true. ...”

Assms	Goal
$\ldots, A$	B

Forwards Rule

If you already know $A \Rightarrow B$ and you already know A, then you know B.

Assms	Goal
$\ldots, A \Rightarrow B, A$	C

"From $A \Rightarrow B$ and A we conclude B."

"Applying $A \Rightarrow B$ to A we obtain B."

"We have B by *modus ponens*."

Assms	Goal
$\ldots, A \Rightarrow B, A, B$	C

2.2 Conjunction

$A \wedge B$ $\qquad$ A and B

Backwards Rule

To prove $A \wedge B$ from some assumptions, it suffices to give two separate proofs: one proof of A from the assumptions and one proof of B from the assumptions.

Assms	Goal
$\ldots$	$A \wedge B$

"We prove A and B separately. To prove A..."

"For A, we argue as follows ... For B, ..."

Assms	Goal
$\ldots$	A

Assms	Goal
$\ldots$	B

The proof splits into two at this point, so you have to keep in mind two states, although you can just work with one at a time.

Forwards Rule

If you know $A \wedge B$, then you can know A and you know B.

Assms	Goal
$\ldots, A \wedge B$	C

“” (too trivial to mention)

Assms	Goal
$\ldots, A \wedge B, A, B$	C

2.3 Disjunction

$A \vee B$ $\qquad$ A or B

Backwards Rules

To prove $A \vee B$ from some assumptions, it suffices to give a proof of A from those assumptions.

Assms	Goal
$\ldots$	$A \vee B$

“To see $A \vee B$, observe that A is true since...”

“We prove A.”

“It suffices to prove A.”

Assms	Goal
$\ldots$	A

To prove $A \vee B$ from some assumptions, it suffices to give a proof of B from those assumptions.

Assms	Goal
$\ldots$	$A \vee B$

“To see $A \vee B$, observe that B is true since...”

“We prove B.”

“It suffices to prove B.”

Assms	Goal
$\ldots$	B

Forwards Rule

If you have already know $A \vee B$ and, additionally, you can give the following two proofs:

- a proof of C starting from your assumptions and A

- a proof of C starting from your assumptions and B

Then you will also know C.

Assms	Goal
$\ldots, A \vee B$	C

"We proceed by cases on $A \vee B$.
Assume A ... Hence C.
Assume B ... Hence C."
"We analyse the two cases."
"We proceed by case analysis on $A \vee B$."

Assms	Goal
$\ldots, A \vee B, A$	C

Assms	Goal
$\ldots, A \vee B, B$	C

This forces the proof to split into two, and you're not done until you have completed both of them.

2.4 Negation

$\neg A$ not A A is false A is absurd

Backwards Rule

To prove $\neg A$ starting from some assumptions, it suffices to derive a contradiction (give a proof of false) starting from the same assumptions with A added.

Assms	Goal
$\ldots$	$\neg A$

"We assume A and try to obtain a contradiction."

"To show $\neg A$, we assume A..."

"Assume A."

Assms	Goal
$\ldots, A$	$\bot$

Forwards Rule

If you know $\neg A$ and you also know A, then you know absurdity.

Assms	Goal
$\ldots, \neg A, A$	C

"From A and $\neg A$ we obtain a contradiction."

Assms	Goal
$\ldots, \neg A, A, \bot$	C

2.5 If and only if

$$\boxed{\Leftrightarrow \equiv \text{iff}}$$

Backwards Rule

An iff can be thought of as two implications, one in each direction. So to prove an iff it suffices to prove each separately.

Assms	Goal
$\ldots$	$A \Leftrightarrow B$

"In the forwards direction...

... in the backwards direction..."

"We prove each direction separately..."

Assms	Goal
$\ldots$	$A \Rightarrow B$

Assms	Goal
$\ldots$	$B \Rightarrow A$

Forwards Rule

Conversely, to use one, first just extract the two directions.

Assms	Goal
$\ldots, A \Leftrightarrow B$	C

"" (too trivial to mention)

Assms	Goal
$\ldots, A \Leftrightarrow B, A \Rightarrow B, B \Rightarrow A$	C

2.6 False

$\perp$ false absurdity

Forwards Rule

If you know absurdity, then you know everything.

Assms	Goal
$\ldots, \bot$	C

"Hence we obtain the desired result."

"*Ex falso quodlibet*"

"A follows trivially"

Assms	Goal
$\ldots, \bot, A$	C

Note, this A can be anything you like, including C.

2.7 Universal quantification

$\forall x \in X.\, A$	$\forall x : X.\, A$	for all x in X, A	A holds of all x in X

Backwards Rule

To give a proof of $\forall x \in X.\ A$ from some assumptions, it suffices to give a proof of A from the same assumptions with $x \in X$ added, as long as you have not made any prior assumptions about the name x (x doesn't appear in any of your existing assumptions).

Assms	Goal
$\ldots$	$\forall x \in X.A$

with x not occurring free in the assms

"Let $x \in X$. We show A."

"Let x be an arbitrary member of X. We show A."

"Suppose x is in X. ... therefore A."

Assms	Goal
$\ldots, x \in X$	A

Forwards Rule

If you have know $\forall x \in X.\, A$ and you know $t \in X$, then you know A with all occurrences of x replaced by t.

Assms	Goal
$\ldots\ \forall x \in X.A\, t \in X$	C

"It follows that A holds of t"

Assms	Goal
$\ldots,\ \forall x \in X.A,\ t \in X,\ A[t/x]$	C

2.8 Existential Quantification

$\exists x \in X.\, A$	$\exists x : X.\, A$	there exists x in X such that A	A holds of some x in X

Backwards Rule

To prove $\exists x : X.\, A$ from some assumptions, it suffices to find a witness t and give proofs that $t \in X$ and of A with every occurence of x replaced by t, starting from the same assumptions.

Assms	Goal
$\ldots$	$\exists x \in X.\, A$

"We show that A holds of t."

"We take t as witness. To see $A[t/x]$..."

"We show that t is such an x."

Assms	Goal
$\ldots$	$t \in X$

Assms	Goal
$\ldots$	$A[t/x]$

Here, $A[t/x]$ means A with all free occurrences of x replaced by t.

Forwards Rule

Assms	Goal
$\ldots\ \exists x \in X.A$	C

with x not occurring free in the assms

"Let x be the witness..."

Assms	Goal
$\ldots\ \exists x \in X.A\, x \in X\, A$	C

3 Playing the game

There are generally many ways that one can complete a proof because, in any given proof state, several forwards and backwards rules may be applicable. In many cases, it requires trial and error until you find the right strategy for a proof of a given statement. However:

An excellent rule of thumb is to:

1. Apply backwards reasoning until the goal is as simple as possible (doesn't involve any more logical connectives) and all the delicious extra assumptions have been extracted.
2. Rewrite the goal using any non-inductive/recursive definitions that you have assumed (see Theories and Equality in Section 4). This may give you even more opportunities to apply backwards reasoning steps.
3. Step back and consider what you now know to be true (your assumptions). Then apply forwards reasoning to deduce more and more of them until you have finally deduced the goal you are looking to prove.

This strategy will not work in all cases, but I would always use it as my first attempt at a proof. In this unit, it will serve you very well for almost every proof.

3.1 Examples

The following is an example of pure logic. Let's suppose we do not know any facts about logic already (i.e. our starting assumptions are empty).

Lemma 2. *A implies $\neg\neg A$*

Proof. Assume A. We wish to show $\neg\neg A$, so we assume $\neg A$ and try to obtain a contradiction. From A and $\neg A$ we obtain the desired contradiction. □

Make sure you can follow which rule is being used in each step and how the proof state evolves. The first step in becoming proficient at proving is to at least be able to check that someone else's proof is valid. The following table gives a listing of the proof state at the end of each step in the proof.

Proof step	Assumptions	Goal
		$A \Rightarrow \neg\neg A$
"Assume A"	A	$\neg\neg A$
"... so we assume $\neg A$ and ... contradiction."	A, $\neg A$	$\bot$
"From A and $\neg A$ we obtain..."	A, $\neg A$, $\bot$	$\bot$

In the following example, we collect quite a few assumptions from our initial backward reasoning. In such cases it can sometimes be helpful to label them as we go for when we want to combine them in new ways during forward reasoning.

Lemma 3. *if A implies B then $\neg B$ implies $\neg A$*

Proof. Assume $A \Rightarrow B$ (A1) and assume $\neg B$ (A2). We aim to prove $\neg A$, so we will assume A (A3) and try to obtain a contradiction. From (A1) and (A3), we obtain B (A4). From (A4) and (A2) we obtain the desired contradiction. □

This is what is going on in my head when I read this proof back to myself. Let's again suppose that my starting knowledge is zero. Make sure that you can identify which backwards or forwards rule was used in each case.

Proof step	Assumptions	Goal
		$(A \Rightarrow B) \Rightarrow \neg B \Rightarrow A$
"Assume $A \Rightarrow B$"	A1:$A \Rightarrow B$	$\neg B \Rightarrow \neg A$
"assume $\neg B$"	A1:$A \Rightarrow B$, A2:$\neg B$	$\neg A$
"assume A ... try ... contradiction."	A1:$A \Rightarrow B$, A2:$\neg B$, A3:A	$\bot$
"From (A1) and (A3), we obtain B."	A1:$A \Rightarrow B$, A2:$\neg B$, A3:A, A4:B	$\bot$
"From ... obtain ... contradiction."	A1:$A \Rightarrow B$, A2:$\neg B$, A3:A, A4:B, $\bot$	$\bot$

One important reason that it is helpful to have the proof state in your head is that, because the proof steps are just certain phrases in English, sometimes the proof state is needed for disambiguation. For example, it may only be possible to understand whether the word "assume" heralds a use of implication introduction or a use of negation introduction, based on goal at the point at which the word was used. Here's another example.

Lemma 4. $A \Rightarrow (B \Rightarrow C)$ *implies* $A \wedge B \Rightarrow C$

Proof. Assume $A \Rightarrow (B \Rightarrow C)$ (*). Assume A and B. From (*) and A, $B \Rightarrow C$ follows. From this and B, C follows. □

Proof step	Assumptions	Goal
		$(A \Rightarrow (B \Rightarrow C)) \Rightarrow (A \wedge B) \Rightarrow C$
"Assume $A \Rightarrow (B \Rightarrow C)$ (*)"	*:$A \Rightarrow (B \Rightarrow C)$	$A \wedge B \Rightarrow C$
"Assume A and B"	*:$A \Rightarrow (B \Rightarrow C)$, A, B	C
"From (*) and A, $B \Rightarrow C$ follows."	*:$A \Rightarrow (B \Rightarrow C)$, A, B, $B \Rightarrow C$	C
"From this and B, C follows."	*:$A \Rightarrow (B \Rightarrow C)$, A, B, $B \Rightarrow C$, C	C

In this example I omitted the step between assuming $A \wedge B$ and deducing A and B separately because there is never any reason not to immediately deduce the two conjuncts separately.

Here's one last example that demonstrates some of the quantifier rules. I leave it up to you to write out the trace.

Lemma 5. *Suppose x does not occur free in B. Then $(\exists x \in X.\, A) \Rightarrow B$ iff $\forall x \in X.\, A \Rightarrow B$.*

Proof. Assume x does not occur free in B. We proceed to prove both directions separately:

- In the forward direction, suppose $(\exists x \in X.\, A) \Rightarrow B$ (*). Then let $x \in X$ and assume A. Note that, obviously, $A = A[x/x]$ and we know that $x \in X$, so we can conclude $\exists x \in X.\, A$. It follows from (*) that, therefore, B.

- In the backward direction, suppose $\forall x \in X.\, A \Rightarrow B$ (*). Then suppose $\exists x \in X.\, A$. Let the witness be some $y \in X$ (so we know $A[y/x]$). Then it follows from (*) that $A[y/x] \Rightarrow B[y/x]$. Since we have the former already, we obtain the latter $B[y/x]$. But, we started by assuming that x does not occur in B, so $B[y/x]$ is exactly B.

□

Here is the trace of the backward direction part of the proof, the inital goal is a bit long so let us shorten it to $G := (\forall x \in X.\, A \Rightarrow B) \Rightarrow ((\exists x \in X.\, A) \Rightarrow B)$.

Proof step	Assumptions	Goal
	$x \notin \mathsf{FV}(B)$	G
"suppose ... (*)"	$x \notin \mathsf{FV}(B)$, $\forall x \in X.\, A \Rightarrow B$	$(\exists x \in X.\, A) \Rightarrow B$
"Then suppose ..."	$x \notin \mathsf{FV}(B)$, $\forall x \in X.\, A \Rightarrow B$, $\exists x \in X.\, A$	B
"Let the witness be..."	$x \notin \mathsf{FV}(B)$, $\forall x \in X.\, A \Rightarrow B$, $\exists x \in X.\, A$, $y \in X$, $A[y/x]$	B
"The it follows from (*)"	$x \notin \mathsf{FV}(B), \ldots, A[y/x], A[y/x] \Rightarrow B[y/x]$	B
"Since we have the former..."	$x \notin \mathsf{FV}(B), \ldots, A[y/x], A[y/x] \Rightarrow B[y/x], B[y/x]$	B

3.2 Bending the rules

There are many, many ways that people take short-cuts when writing proofs. Sometimes it is just laziness, but other times it makes the difference between a proof that is a reasonable size and a proof that is large beyond comprehension. Generally, it's ok to leave steps implicit as long as you believe that your reader can fill in the gaps: it should be possible, in principal, to derive all the explicit steps that you missed out without having to do too much search.

Sometimes backwards reasoning steps are left implicit. For example, when proving a goal of the form $\forall x \in X.\, A \Rightarrow B$, often the proof will launch straight into proving B and, as part of that proof, will use A and $x \in X$ as if they are already assumed.

Conversely, when forwards reasoning from an assumptions of shape $\forall x \in X.\, A \Rightarrow B$ and $A[t/x]$ and $t \in X$, often the proof will immediately deduce $B[t/x]$, performing the forwards rule for forall and the forwards rule for implication in one combined step.

I will try not to leave backwards steps implicit, because they are crucial to understanding where the proof is headed, which is important if you are just starting out. I will, however, often combine forwards steps together if I think it remains clear what is happening.

4 Theories and equality

We have talked about the generic, purely logical aspects of proof but, in reality, most proofs argue something in a specific domain (or, in the logical jargon, a specific theory). When you are working with a particular theory, e.g. the theory of arithmetic over the natural numbers, the theory gives you more to work with: for example, the language of formulas is enlarged to include new constants, function symbols and relations like 0, + and < respectively. Along with those symbols come axioms or definitions, and you can make use of these as extra assumptions in your proofs.

Most theories make use of some notion of equality over the inhabitants of the domain. The rules we can use for reasoning about equality are very straightforward:

Assms	Goal
$\dots$	C

“” (too trivial to mention)

Assms	Goal
$\dots, s = s$	C

At any time, you can assume $s = s$, for any term s, which is obviously true. On the other hand, if you know an equation, you can use it to rewrite the goal.

Assms	Goal
$\dots, s = t$	$C[s/x]$

“$C[s/x]$ is just $C[t/x]$”

“$C[s/x]$ is therefore the same as $C[t/x]$”

Assms	Goal
$\dots, s = t$	$C[t/x]$

The use of $C[s/x]$ here is just a way of saying that you have some proposition which contains an occurrence of s and the corresponding $C[t/x]$ indicates replacing that occurrence of s by t. It also makes sense to rewrite the assumptions:

Assms	Goal
$\ldots, s = t, A[s/x]$	C

"$C[s/x]$ is just $C[t/x]$"

"$C[s/x]$ is therefore the same as $C[t/x]$"

Assms	Goal
$\ldots, s = t, A[s/x], A[t/x]$	C

The following example uses the following recursive definition of mutliplication of natural numbers:

$$\begin{array}{rcl} 0 * m & = & 0 \\ (n+1) * m & = & n * m + m \end{array}$$

If an equational definition of some function f is valid in a given theory, then we can just take the equations as extra assumptions when reasoning about f.

As usual, when we see a statement, such as the equations above, that contains free variables (here m and n), then the convention is to regard them as universally quantified, i.e:

$$\forall m \in \mathbb{N}.\ 0 * m = 0 \qquad \text{and} \qquad \forall mn \in \mathbb{N}.\ (n+1) * m = n * m + m$$

For this reason, it is extremely important to know your definitions well, because you will always be using them when you come to writing proofs. For the purpose of this example, we also allow ourselves to assume the commutativity of multiplication:

$$\forall pq \in \mathbb{N}.\ p * q = q * p$$

Lemma 6. *for all $n, m \in \mathbb{N}$: if $n = 0$ or $m = 0$ it follows that $n * m = 0$*

Proof. Let $n, m \in \mathbb{N}$ and assume $n = 0 \vee m = 0$. We show that $n * m = 0$ by case analysis.

- When $n = 0$ then the goal $n * m = 0$ is $0 * m = 0$ which is true by the definition of multiplication.
- When $m = 0$ then the goal $n * m = 0$ is $n * 0 = 0$ which, by the commutativity of mutliplication, is $0 * n = 0$ and this also true by definition.

□

Some theories even come with their own proof rules for reasoning about the datatypes involved. Induction over the natural numbers is an excellent example:

Assms	Goal
$\dots$	$\forall n \in \mathbb{N}.\, A$

"We prove the result by induction on $x \in \mathbb{N}$.

When $x = 0$ ... hence A is true of 0.

When $x = k + 1$, assume A holds of k

... A is true of $k + 1$."

"By induction on $n \in \mathbb{N}$. We analyse the cases..."

Assms	Goal
$\dots$	$A[0/n]$

Assms	Goal
$\dots, A[k/n]$	$A[k+1/n]$

So, in the sense above, natural number induction is an alternative introduction rule (backward reasoning) for those formulas of shape $\forall x : X.\, A$ in the specific case that X is $\mathbb{N}$.

5 Non-constructive reasoning

This is the last "rule" in our arsenal as far as pure logic is concerned (if we exclude equality) and it takes us from intuititionistic to classical logic.

Assms	Goal
$\dots$	C

"By excluded middle either A or $\neg A$"

"Either A or $\neg A$, so..."

"$A \vee \neg A$ by LEM"

"By the sacred principal of *tertium non datur*..."

Assms	Goal
$\dots, A \vee \neg A$	C

This rule is needed in order to prove the converse of two of the previous lemmas. For example:

Lemma 7. *$\neg\neg P$ implies P*

Proof. Assume $\neg\neg P$. By excluded middle, $P \vee \neg P$. We proceed by case analysis to conclude P. In the first case we assume P and then the result is true by assumption. In the second case we assume $\neg P$ and then from this and our original assumption we obtain a contradiction. Therefore, P follows trivially. □

Proof by contradiction

The law of the excluded middle is quite clumsy to use in a proof. However, it is equivalent to the principle of *proof by contradiction* or *reductio ad absurdum*. A proof by contradiction proves the goal A by assuming $\neg A$ and attempting to obtain a contradiction.

Assms	Goal
$\ldots$	C

"We prove the result by contradiction,

so assume $\neg C$..."

"Assume $\neg C$, ...

hence we have obtained a contradiction.

We conclude C by *reductio ad absurdum*."

Assms	Goal
$\ldots, \neg C$	$\bot$

You might think this rule is quite reminiscent of the backwards rule for negation. Indeed it is! The only difference is that the backwards reasoning rule for negation only applies when the goal has shape $\neg A$. So we can't use this rule to prove an arbitrary C directly (unless C happens to be of shape $\neg A$).

We can, however, use the negation rule to prove $\neg\neg A$, and we know from Lemmas 2 and 6 that $\neg\neg A$ is logically equivalent to A (in the sense that they each imply the other). What is interesting about this is that Lemma 6 is only true by virtue of excluded middle. If we didn't have that rule, then it would be impossible to conclude that A and $\neg\neg A$ are logically equivalent. There is a sect of mathematicians called *intuitionists* (part of a larger branch called *constructivists*) that deny the validity of excluded middle and therefore all that follows from it (such as Lemma 6). To intuitionists, proof by contradiction is a not a principle but an abomination. I am personally quite relaxed about it. Anyway, we will talk about intuitionism more later in the unit, because it is very closely related to type theory.

One, somewhat more practical, objection to proof by contradiction is that it is not suggested by any particular logical operator. If you consider all the rules we have looked at until this point, your choice of rule is restricted by the shape of the goal or the shape of the assumptions. Proof by contradiction, however, can be used at any time. This complicates matters a bit when writing proofs because you have more choices. There is no syntactic clue in any of the formulas in the proof state, you just have to decide to do it based on your intuition. It is wise to accept that writing proofs, like writing programs, sometimes requires a bit of trial and error. Here is an example; you might like to write out the evolution of the proof state as practice:

Lemma 8. *$\neg Q \Rightarrow \neg P$ implies $P \Rightarrow Q$*

Proof. Assume $\neg Q \Rightarrow \neg P$ (A) and assume P. We prove Q by contradiction, so assume $\neg Q$. It follows from (A) that, additionally, $\neg P$. Hence, from P and $\neg P$ we have obtained a contradiction. □

This implication that we have now proven gives us another well known proof principle called *proof by contrapositive*: in order to prove $P \Rightarrow Q$, it is enough to prove $\neg Q \Rightarrow \neg P$.

Extra proof principles, like the contrapositive and natual number induction, give you yet more alternative approaches to how to structure the proof.

When faced with a goal like $\forall x : \mathbb{N}.\ P \Rightarrow Q$ you have several possible first steps. To dispense with the $\forall$ you could attempt a proof by induction or you could simply assume x as an arbitrary element. To prove $P \Rightarrow Q$ you could either assume P and try to show Q or, alternatively, you could try to prove the contrapositive. Maybe you would prefer to assume $\exists x : \mathbb{N}. \neg(P \Rightarrow Q)$ instead and derive a contradiction, concluding by *reductio ad absurdum*? Sometimes all of these approaches will work, though some may be shorter or more clear than others. This, a careful use of phrases like “clearly”, and a sense of when to split a large proof into several lemmas, are what make writing proofs a kind of art, and you will only get better at choosing the best approach with practice. Since I do not assume you have much experience at writing proofs, throughout the first half of the unit I will always signal to you in a question which of the commonly used alternative strategies (e.g. induction, contradiction or contrapositive) is a good choice.