Functions
Intro to Induction

Margaret M. Fleck

27 February 2009

This lecture finishes up functions and introduces mathematical induction (section 4.1 of Rosen).

1 Warmup using 2D points

Monday's lecture probably went by really fast since you were all worrying about the midterm. So, let's remember what it means to prove that a specific function is, say, onto.

Let $f : \mathbb{Z}^2 \rightarrow \mathbb{Z}$ be defined by $f(x, y) = x + y$. I claim that f is onto.

First, let's make sure we know how to read this definition. $f : \mathbb{Z}^2$ is shorthand for $\mathbb{Z} \times \mathbb{Z}$, which is the set of pairs of integers. So f maps a pair of integers to a single integer, which is the sum of the two coordinates.

To prove that f is onto, we need to pick some arbitrary element y in the co-domain. That is to say, y is an integer. Then we need to find a sample value in the domain that maps onto y, i.e. a "preimage" of y. At this point, it helps to fiddle around on our scratch paper, to come up with a suitable preimage. In this case, $(0, y)$ will work nicely. So our proof looks like:

> Proof: Let y be an element of $\mathbb{Z}$. Then $(0, y)$ is an element of $f : \mathbb{Z}^2$ and $f(0, y) = 0 + y = y$. Since this contruction will work for any choice of y, we've shown that f is onto.

2 Another proof involving composition

Last class, we saw a proof of the following fact:

Claim 1 *For any sets A, B, and C and for any functions $f : A \to B$ and $g : B \to C$, if f and g are injective, then $g \circ f$ is also injective.*

Let's prove a similar claim involving surjective:

Claim 2 *For any sets A, B, and C and for any functions $f : A \to B$ and $g : B \to C$, if f and g are surjective, then $g \circ f$ is also surjective.*

> Proof: Let A, B, and C be sets. Let $f : A \to B$ and $g : B \to C$ be functions. Suppose that f and g are surjective.
>
> We need to show that $g \circ f$ is surjective. That is, we need to show that for any element x in C, there is an element y in A such that $(g \circ f)(y) = x$.
>
> So, pick some element x in C. Since g is surjective, there is an element z in B such that $g(z) = x$. Since f is surjective, there is an element y in A such that $f(y) = z$.
>
> Substituting the value $f(y) = z$ into the equation $g(z) = x$, we get $g(f(y)) = x$. That is, $(g \circ f)(y) = x$. So y is the element of A we needed to find.

3 Proof by cases

Now, let's do a slightly trickier proof. First, a definition. Suppose that A and B are sets of numbers (e.g. the reals, the rationals, the integers). A function $f : A \to B$ is *increasing* if, for every x and y in A, $x \leq y$ implies that $f(x) \leq f(y)$. f is called *strictly increasing* if, for every x and y in A, $x < y$ implies that $f(x) < f(y)$.[1]

[1] In math, "strictly" is often used to exclude the possibility of equality.

Claim 3 *For any sets of number A and B, if f is any strictly increasing function from A to B, then f is one-to-one.*

A similar fact applies to strictly decreasing functions.

To prove this, we will restate one-to-one using the alternative, contrapositive version of its definition.

$$\forall x, y \in A, x \neq y \rightarrow f(x) \neq f(y)$$

Normally, this wouldn't be a helpful move because we've created negative facts when we used to have positive facts. But, in this case, it's a good approach.

> Proof: Let A and B be sets of numbers and let $f : A \rightarrow B$ be a strictly increasing function. Let x and y be distinct elements of A. We need to show that $f(x) \neq f(y)$.
>
> Since $x \neq y$, there's two possibilities.
>
> Case 1: $x < y$. Since f is strictly increasing, this implies that $f(x) < f(y)$. So $f(x) \neq f(y)$
>
> Case 2: $y < x$. Since f is strictly increasing, this implies that $f(y) < f(x)$. So $f(x) \neq f(y)$.
>
> In either case, we have that $f(x) \neq f(y)$, which is what we needed to show.

The phrase "distinct elements of A" is math jargon for $x \neq y$.

When we got partway into the proof, we had the fact $x \neq y$ which isn't easy to work with. But the trichotomy axiom for real number states that for any x and y, we have exactly three possibilities: $x = y$, $x < y$, or $y < x$. The constraint that $x \neq y$ eliminates one of these possibilities.

We then used a technique called "proof by cases". In proof by cases, you establish a list of several possibilities, one of which must be true. You then try each possibility in turn, assuming it's true and showing that your desired conclusion follows. If the conclusion follows no matter which possibility you've chosen, it must be true in general. This technique isn't hard. The

main point to be careful about is making sure your cases do actually cover all the possibilities.

4 Without loss of generality

In this example, the proofs for the two cases are very, very similar. So we can fold the two cases together. Here's one approach, which I don't recommend doing in the early part of this course but which will serve you well later on:

> Proof: Let A and B be sets of numbers and let $f : A \to B$ be a strictly increasing function. Let x and y be distinct elements of A. We need to show that $f(x) \neq f(y)$.
>
> Since $x \neq y$, there's two possibilities.
>
> Case 1: $x < y$. Since f is strictly increasing, this implies that $f(x) < f(y)$. So $f(x) \neq f(y)$
>
> Case 2: $y < x$. Similar to case 1.
>
> In either case, we have that $f(x) \neq f(y)$, which is what we needed to show.

This method only works if you, and your reader, both agree that it's obvious that the two cases are very similar and the proof will really be similar. Dangerous assumption right now. And we've only saved a small amount of writing, which isn't worth the risk of losing points if the TA doesn't think it was obvious.

But this simplificiation is very useful in more complicated situations (e.g. in CS 273) where you have may have lots of cases and the proofs really are very similar and the proof for each case is long.

Here's another way to simplify our proof:

> Proof: Let A and B be sets of numbers and let $f : A \to B$ be a strictly increasing function. Let x and y be distinct elements of A. We need to show that $f(x) \neq f(y)$.

We know that $x \neq y$, so either $x < y$ or $y < x$. Without loss of generality, assume that $x < y$.

Since f is strictly increasing, $x < y$ implies that $f(x) < f(y)$. So $f(x) \neq f(y)$, which is what we needed to show.

The phrase "without loss of generality" means that we are adding an additional assumption to our proof but we claim it isn't really adding any more information. In this case, x and y are both arbitrary elements of A and the condition $f(x) \neq f(y)$ is symmetrical in its two inputs (i.e. it means the same thing as $f(y) \neq f(x)$. So we haven't yet made any real distinction in properties between x and y. It doesn't actually matter which of the two numbers is called x and which is called y. So if we happen to have chosen our names in the wrong order so that $y < x$, we can just rename our two numbers in the opposite way. And then we'll have $x < y$.

Again, this is a potentially dangerous proof technique that you'll probably want to see a few times before you use it yourself. It's actually used fairly often to simplify proofs, so it has an abbreviation: WOLOG or WLOG.

5 Introduction to induction

At the start of the term, we saw the following formula for computing the sum of the first n integers:

Claim 4 *For any positive integer n, $\Sigma_{i=1}^{n}\ i = \frac{n(n+1)}{2}$.*

At that point, we didn't prove this formula correct, because this is most easily done using a new proof technique: induction.

Mathematical induction is a technique for showing that a statement $P(n)$ is true for all natural numbers n, or for some infinite subset of the natural numbers (e.g. all positive even integers). It's a nice way to produce quick, easy-to-read proofs for a variety of fact that would be awkward to prove with the techniques you've seen so far. It is particularly well suited to analyzing the performance of recursive algorithms. Most of you have seen a few of

these in previous programming classes and you'll see a lot more of them as you finish the rest of the major.

Induction is very handy, but it may strike you as a bit wierd. It may take you some time to get used to it. So, we've got two tasks which are a bit independent:

- Learn how to write an inductive proof.
- Understand why inductive proofs are legitimate.

Everyone needs to be able to write an inductive proof to pass this class. However, some of you might finish the term feeling a bit shaky on whether you believe induction works correctly. That's ok. You can still use the technique, relying on the fact that we say it's ok. You'll see induction (and its friend recursion) in later terms and will gradually come to feel more confident about its validity.

6 An Example

A simple proof by induction has the following outline:

> Proof: We will show $P(n)$ is true for all n, using induction on n.
>
> **Base**: We need to show that $P(1)$ is true.
>
> **Induction**: Suppose that $P(k)$ is true, for some integer k. We need to show that $P(k+1)$ is true.

For our formula example, our proposition $P(n)$ is $\Sigma_{i=1}^{n}\ i = \frac{n(n+1)}{2}$.

Notice two things about $P(n)$. First, it is a statement, i.e. something that is either true or false. For example, it is **never** just a formula whose value is a number. Second, this statement depends on an integer n. This integer n is known as our *induction variable*.

The part of the proof labelled "induction" is a conditional statement. We assume that $P(k)$ is true. This assumption is called the *inductive hypothesis*. We use this assumption to show that $P(k+1)$ is true.

Substituting in the definition of P for our example, we get the following outline:

> Proof: We will show that $\Sigma_{i=1}^{n}\ i = \frac{n(n+1)}{2}$ for any integer n, using induction on n.
>
> **Base**: We need to show that the formula holds for $n = 1$, i.e. $\Sigma_{i=1}^{1}\ i = \frac{1 \cdot 2}{2}$.
>
> **Induction**: Suppose that $\Sigma_{i=1}^{k}\ i = \frac{k(k+1)}{2}$ for some positive integer k. We need to show that $\Sigma_{i=1}^{k+1}\ i = \frac{(k+1)(k+2)}{2}$.

So, in the part marked "induction", we're assuming the formula works for some k and we'll use that formula for k to work out the formula for $k+1$. Let's not worry about how to fill in the details for the moment. (I'll do that in a few minutes.) Let's just look at logic behind the outline.

7 Why is this legit?

There are several ways to think about mathematical induction, and understand why it's a legitimate proof technique. Different people prefer different motivations at this point, so I'll offer several.

A proof by induction of that $P(k)$ is true for all positive integers [2] k involves showing that $P(1)$ is true (base case) and that $P(k) \rightarrow P(k+1)$ (inductive step).

> **Domino Theory:** Imagine an infinite line of dominoes. The base step pushes the first one over. The inductive step claims that one domino falling down will push over the next domino in the line. So dominos will start to fall from the beginning all the way down the line. This process continues forever, because the line is infinitely long. However, if you focus on any specific domino, it falls after some specific finite delay.

[2] Or all natural numbers, or indeed any set of integers starting at some specific lower bound.

Recursion fairy: The recursion fairy is the mathematician's version of a programming assistant. Suppose you tell her how to do the proof for $P(1)$ and also why $P(k)$ implies $P(k+1)$. Then suppose you pick any integer (e.g. 1034) then she can take this recipe and use it to fill in all the details of a normal direct proof that P holds for this particular integer. That is, she takes $P(1)$, then uses the inductive step to get from $P(1)$ to $P(2)$, and so on up to $P(1034)$.

Smallest counter-example: Let's assume we've established that $P(1)$ is true and also that $P(k)$ implies $P(k+1)$. Let's prove that $P(j)$ is true for all j, by contradiction.

That is, we suppose that $P(1)$ is true, also that $P(k)$ implies $P(k+1)$, but there is a counter-example to our claim that $P(j)$ is true for all j. That is, suppose that $P(m)$ was not true for some integer m.

Now, let's look at the set of all counter-examples. We know that all the counter-examples are larger than 1, because our induction proof established explicitly that $P(1)$ was true. Suppose that the smallest counter-example is s. So $P(s)$ is true. We know that $s > 1$, since $P(1)$ was true. Since s was supposed to be the smallest counter-example, $s-1$ must not be a counter-example, i.e. $P(s-1)$ is true.

But now we know that $P(s-1)$ is true but $P(s)$ is not true. This directly contradicts our assumption that $P(k)$ implies $P(k+1)$ for any k.

The smallest counter-example explanation is a formal proof that induction works, given how we've defined the integers. If you dig into the mathematics, you'll find that it depends on the integers having what's called the "well-ordering" property: any subset has a smallest element. Standard axioms used to define the integers include either a well-ordering or an induction axiom.