Induction and Recursive Definition

Margaret M. Fleck

10 March 2011

These notes cover mathematical induction and recursive definition

1 Introduction to induction

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

Claim 1 *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 weird. 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 course. 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.

2 An Example

A simple proof by induction has the following outline:

> Claim: $P(n)$ is true for all positive integers n.
>
> Proof: We'll use induction on n.
>
> **Base**: We need to show that $P(1)$ is true.
>
> **Induction**: Suppose that $P(k)$ is true, for some positive integer k. We need to show that $P(k+1)$ is true.

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.

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

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 positive 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 $k+1$.

For our specific example, the induction proof might look as follows:

> Proof: We will show that $\Sigma_{i=1}^{n} i = \frac{n(n+1)}{2}$ for any positive integer n, using induction on n.
>
> **Base**: We need to show that the formula holds for $n = 1$. $\Sigma_{i=1}^{1} i = 1$. And also $\frac{1 \cdot 2}{2} = 1$. So the two are equal for $n = 1$.
>
> **Induction**: Suppose that $\Sigma_{i=1}^{k} i = \frac{k(k+1)}{2}$ for some positive integer k. We need to show that $\Sigma_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$.
>
> By the definition of summation notation, $\Sigma_{i=1}^{k+1} i = (\Sigma_{i=1}^{k} i) + (k+1)$
>
> Substituting in the formula from our inductive hypothesis, we get that $(\Sigma_{i=1}^{k} i) + (k+1) = (\frac{k(k+1)}{2}) + (k+1)$.
>
> But $(\frac{k(k+1)}{2}) + (k+1) = \frac{k(k+1)}{2} + \frac{2(k+1)}{2} = \frac{(k+2)(k+1)}{2} = \frac{(k+1)(k+2)}{2}$.
>
> So, combining these equations, we get that $\Sigma_{i=1}^{k+1} i = \frac{(k+1)(k+2)}{2}$ which is what we needed to show.

One way to think of a proof by induction is that it's a template for building direct proofs. If I give you the specific value $n = 47$, you could write a direct proof by starting with the base case and using the inductive step 46 times to work your way from the $n = 1$ case up to the $n = 47$ case.

3 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 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.

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 positive integers 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 not 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 that has a lower bound also has a smallest element. Standard axioms used to define the integers include either a well-ordering or an induction axiom.

These arguments don't depend on whether our starting point is 1 or some other integer, e.g. 0 or 2 or -47. All you need to do is ensure that your base case covers the first integer for which the claim is supposed to be true.

4 Building an inductive proof

In constructing an inductive proof, you've got two tasks. First, you need to set up this outline for your problem. This includes identifying a suitable proposition P and a suitable integer variable n.

Notice that $P(n)$ must be a statement, i.e. something that is either true or false. For example, it is **never** just a formula whose value is a number. Also, notice that $P(n)$ must depend on an integer n. This integer n is known as our *induction variable*. The assumption at the start of the inductive step ("$P(k)$ is true") is called the inductive hypothesis.

Your second task is to fill in the middle part of the induction step. That is, you must figure out how to relate a solution for a larger problem $P(k+1)$ to a solution for a small problem $P(k)$. Most students want to do this by starting with the small problem and adding something to it. For more complex situations, it's usually better to start with the larger problem and try to find an instance of the smaller problem inside it.

5 Another example of induction

Let's do another example:

Claim 2 *For every positive integer* $n \geq 4$, $2^n < n!$.

Remember that $n!$ ("n factorial") is $1 \cdot 2 \cdot 3 \cdot 4 \ldots n$. E.g. $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$.

First, as usual, try some specific integers and verify that the claim is true. Since the claim specifies $n \geq 4$, it's worth checking that 4 does work but the smaller integers don't.

In this claim, the proposition $P(n)$ is $2^n < n!$. So an outline of our inductive proof looks like:

> Proof: Suppose that n is an integer and $n \geq 4$. We'll prove that $2^n < n!$ using induction on n.
>
> Base: $n = 4$. [show that the formula works for $n = 4$]
>
> Induction: Suppose that the claim holds for $n = k$. That is, suppose that we have an integer $k \geq 4$ such that $2^k < k!$. We need to show that the claim holds for $n = k+1$, i.e. that $2^{k+1} < (k+1)!$.

Notice that our base case is for $n = 4$ because the claim was specified to hold only for integers ≥ 4.

Fleshing out the details of the algebra, we get the following full proof. When working with inequalities, it's especially important to write down your assumptions and what you want to conclude with. You can then work from both ends to fill in the gap in the middle of the proof.

> Proof: Suppose that n is an integer and $n \geq 4$. We'll prove that $2^n < n!$ using induction on n.
>
> Base: $n = 4$. In this case $2^n = 2^4 = 16$. Also $n! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$. Since $16 < 24$, the formula holds for $n = 4$.
>
> Induction: Suppose that the claim holds for $n = k$. That is, suppose that we have an integer $k \geq 4$ such that $2^k < k!$. We need to show that $2^{k+1} < (k+1)!$.
>
> $2^{k+1} = 2 \cdot 2^k$. By the inductive hypothesis, $2^k < k!$, so $2 \cdot 2^k < 2 \cdot k!$. Since $k \geq 4$, $2 < k+1$. So $2 \cdot k! < (k+1) \cdot k! = (k+1)!$.
>
> Putting these equations together, we find that $2^{k+1} < (k+1)!$, which is what we needed to show.

6 Some comments about style

Notice that the start of the proof tells you which variable in your formula (n in this case) is the induction variable. In this formula, the choice of induction variable is fairly obvious. But sometimes there's more than one integer floating around that might make a plausible choice for the induction variable. It's good style to always mention that you are doing a proof by induction and say what your induction variable is.

It's also good style to label your base and inductive steps.

Notice that the proof of the base case is very short. In fact, I've written about about twice as long as you'd normally see it. Almost all the time, the base case is trivial to prove and fairly obvious to both you and your reader. Often this step contains only some worked algebra and a check mark at the end. However, it's critical that you do check the base case. And, if your base case involves an equation, compute the results for both sides (not just one side) so you can verify they are equal.

The important part of the inductive step is ensuring that you assume $P(k)$ and use it to show $P(k+1)$. At the start, you must spell out your inductive hypothesis, i.e. what $P(k)$ is for your claim. Make sure that you use this information in your argument that $P(k+1)$ holds. If you don't, it's not an inductive proof and it's very likely that your proof is buggy.

At the start of the inductive step, it's also a good idea to say what you need to show, i.e. quote what $P(k+1)$ is.

These "style" issues are optional in theory, but actually critical for beginners writing inductive proofs. You will lose points if your proof isn't clear and easy to read. Following these style points (e.g. labelling your base and inductive steps) is a good way to ensure that it is, and that the logic of your proof is correct.

7 Another example

The previous examples applied induction to an algebraic formula. We can also apply induction to other sorts of statements, as long as they involve a suitable integer n.

Claim 3 *For any natural number n, $n^3 - n$ is divisible by 3.*

In this case, $P(n)$ is "$n^3 - n$ is divisible by 3."

Proof: By induction on n.

Base: Let $n = 0$. Then $n^3 - n = 0^3 - 0 = 0$ which is divisible by 3.

Induction: Suppose that $k^3 - k$ is divisible by 3, for some positive integer k. We need to show that $(k+1)^3 - (k+1)$ is divisible by 3.

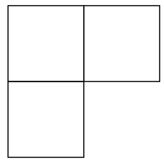
$(k+1)^3 - (k+1) = (k^3 + 3k^2 + 3k + 1) - (k+1) = (k^3 - k) + 3(k^2 + k)$

From the inductive hypothesis, $(k^3 - k)$ is divisible by 3. And $3(k^2 + k)$ is divisible by 3 since $(k^2 + k)$ is an integer. So their sum is divisible by 3. That is $(k+1)^3 - (k+1)$ is divisible by 3. □

The zero base case is technically enough to make the proof solid, but sometimes a zero base case doesn't provide good intuition or confidence. So you'll sometimes see an extra base case written out, e.g. $n = 1$ in this example, to help the author or reader see why the claim is plausible.

8 A geometrical example

Let's see another example of the basic induction outline, this time on a geometrical application. *Tiling* some area of space with a certain type of puzzle piece means that you fit the puzzle pieces onto that area of space exactly, with no overlaps or missing areas. A right triomino is a 2-by-2 square minus one of the four squares.



I claim that

Claim 4 *For any positive integer n, a $2^n \times 2^n$ checkerboard with any one square removed can be tiled using right triominoes.*

Proof: by induction on n.

Base: Suppose $n = 1$. Then our $2^n \times 2^n$ checkerboard with one square removed is exactly one right triomino.

Induction: Suppose that the claim is true for some integer k. That is a $2^k \times 2^k$ checkerboard with any one square removed can be tiled using right triominoes.

Suppose we have a $2^{k+1} \times 2^{k+1}$ checkerboard C with any one square removed. We can divide C into four $2^k \times 2^k$ sub-checkerboards P, Q, R, and S. One of these sub-checkerboards is already missing a square. Suppose without loss of generality that this one is S. Place a single right triomino in the middle of C so it covers one square on each of P, Q, and R.

Now look at the areas remaining to be covered. In each of the sub-checkerboards, exactly one square is missing (S) or already covered (P, Q, and R). So, by our inductive hypothesis, each of these sub-checkerboards minus one square can be tiled with right triominoes. Combining these four tilings with the triomino we put in the middle, we get a tiling for the whole of the larger checkerboard C. This is what we needed to construct.

9 A general result

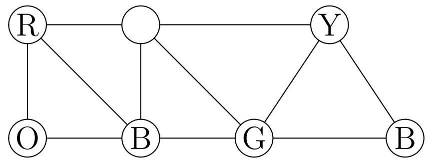
We can also use induction to prove a useful general fact about graph colorability:

Claim 5 *If all vertices in a graph G have degree $\leq D$, then G can be colored with $D + 1$ colors.*

The objects involved in this claim are graphs. To apply induction to objects like graphs, we organize our objects by their size. Each step in the induction process will show that the claim holds for all objects of a particular (integer) size. For graphs, the "size" would typically be either the number of vertices or the number of edges. For this proof, it's most convenient to use the number of vertices.

> Proof: by induction on the number of vertices in G.
>
> Base: The graph with just one vertex has maximum degree 0 and can be colored with one color.
>
> Induction: Suppose that any graph with k vertices and maximum vertex degree $\leq D$ can be colored with $D + 1$ colors.
>
> Let G be a graph with $k + 1$ vertices and maximum vertex degree D. Remove some vertex v (and its edges) from G to create a smaller graph G'.
>
> G' has k vertices. Also, the maximum vertex degree of G' is no larger than D, because removing a vertex can't increase the degree. So, by the inductive hypothesis, G' can be colored with $D + 1$ colors.
>
> Because v has at most D neighbors, its neighbors are only using D of the available colors, leaving a spare color that we can assign to v. The coloring of G' can be extended to a coloring of G with $D + 1$ colors.

For example, here's a graph with vertex maximum degree 4, in which we've colored all but one vertex. We've used five colors so far. But, since the empty vertex has only four neighbors, one of these five colors is free to complete the coloring. This example also illustrates how loose this upper bound is: the theorem guarantees that the chromatic number is ≤ 5 but the true chromatic number for this graph is only 3.



10 Computing colorings

Unfortunately, the result from the previous section only gives us an upper bound on the chromatic number for a graph. Some graphs with a few high-degree vertices do, in fact, have low chromatic number. For example, the central vertex in the wheel graph W_n has degree n. However, even when n is very large, W_n can be colored with only either three or four colors (depending on whether n is even or odd) by alternating colors around the rim of the wheel

In general, finding exact answers to graph coloring problems is very slow. Exact algorithms take exponential time, i.e. time proportional to 2^n, where n is the number of edges or vertices. There are a couple specific versions of the theoretical problem. I could give you a graph and ask you for its chromatic number. Or I could give you a graph and an integer k and ask whether k colors is enough to be able to color G. These problems are all "NP-complete" or "NP-hard." That is, they apparently require exponential time to compute, even though in some cases it's easy to check that their output is correct.[1] So, except for small examples and special cases, exact solution of coloring problems isn't practical.

However, for most practical applications, we can get a reasonable good

[1] They are exponential if the "P" and "NP" classes of algorithms are actually different, which is one of the big outstanding problems in computer science.

coloring using a "greedy" method. In this method, we take our graph G and remove vertices one-by-one, creating a series of smaller and smaller graphs. The goal is to ensure that each vertex has a low degree when removed. So we remove low-degree vertices first in hopes that this will simplify the graph structure around vertices with high degree.

We start by coloring the smallest graph and add the vertices back one-by-one, each time extending the coloring to the new vertex. If each vertex has degree $\leq d$ at the point when it's added to the graph, then we can complete the whole coloring with $d+1$ colors. This algorithm is quite efficient. Its output might use more colors than the optimal coloring, but it apparently works quite well for problems such as register allocation.

11 Strong induction

The inductive proofs you've seen so far have had 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.

Think about building facts incrementally up from the base case to $P(k)$. Induction proves $P(k)$ by first proving $P(i)$ for every i from 1 up through $k-1$. So, by the time we've proved $P(k)$, we've also proved all these other statements. For some proofs, it's very helpful to use the fact that P is true for all these smaller values, in addition to the fact that it's true for k. This method is called "strong" induction.

A proof by strong induction looks like this:

> 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 $P(n)$ is true for $n = 1, 2, \ldots, k$. We need to show that $P(k+1)$ is true.

The only new feature about this proof is that, superficially, we are assuming slightly more in the hypothesis of the inductive step. The difference is actually only superficial, and the two proof techniques are equivalent. However, this difference does make some proofs much easier to write.

12 Postage example

Strong induction is useful when the result for $n = k+1$ depends on the result for some smaller value of n, but it's not the immediately previous value (k). Here's a classic example:

Claim 6 *Every amount of postage that is at least 12 cents can be made from 4-cent and 5-cent stamps.*

For example, 12 cents uses three 4-cent stamps. 13 cents of postage uses two 4-cent stamps plus a 5-cent stamp. 14 uses one 4-cent stamp plus two 5-cent stamps. If you experiment with small values, you quickly realize that the formula for making k cents of postage depends on the one for making $k-4$ cents of postage. That is, you take the stamps for $k-4$ cents and add another 4-cent stamp. We can make this into an inductive proof as follows:

> Proof: by induction on the amount of postage.
>
> Base: If the postage is 12 cents, we can make it with three 4-cent stamps. If the postage is 13 cents, we can make it with two 4-cent stamps. plus a 5-cent stamp. If it is 14, we use one 4-cent stamp plus two 5-cent stamps. If it is 15, we use three 5-cent stamps.
>
> Induction: Suppose that we have show how to construct postage for every value from 12 up through k. We need to show how to construct $k+1$ cents of postage. Since we've already proved base cases up through 15 cents, we'll assume that $k+1 \geq 16$.
>
> Since $k+1 \geq 16$, $(k+1)-4 \geq 12$. So by the inductive hypothesis, we can construct postage for $(k+1)-4$ cents using m 4-cent stamps and n 5-cent stamps, for some natural numbers m and n. In other words $(k+1)-4) = 4m+5n$.

> But then $k + 1 = 4(m + 1) + 5n$. So we can construct $k + 1$ cents of postage using $m + 1$ 4-cent stamps and n 5-cent stamps, which is what we needed to show.

Notice that we needed to directly prove four base cases, since we needed to reach back four integers in our inductive step. It's not always obvious how many base cases are needed until you work out the details of your inductive step.

13 Nim

In the parlour game Nim, there are two players and two piles of matches. At each turn, a player removes some (non-zero) number of matches from one of the piles. The player who removes the last match wins.[2]

Claim 7 *If the two piles contain the same number of matches at the start of the game, then the second player can always win.*

Here's a winning strategy for the second player. Suppose your opponent removes m matches from one pile. In your next move, you remove m matches from the other pile, thus evening up the piles. Let's prove that this strategy works.

> Proof by induction on the number of matches (n) in each pile.
>
> Base: If both piles contain 1 match, the first player has only one possible move: remove the last match from one pile. The second player can then remove the last match from the other pile and thereby win.
>
> Induction: Suppose that the second player can win any game that starts with two piles of n matches, where n is any value from 1 through k. We need to show that this is true if $n = k + 1$.
>
> So, suppose that both piles contain $k + 1$ matches. A legal move by the first player involves removing j matches from one pile,

[2] Or, in some variations, loses. There seem to be several variations of this game.

where $1 \leq j \leq k+1$. The piles then contain $k+1$ matches and $k+1-j$ matches.

The second player can now remove j matches from the other pile. This leaves us with two piles of $k+1-j$ matches. If $j = k+1$, then the second player wins. If $j < k+1$, then we're now effectively at the start of a game with $k+1-j$ matches in each pile. Since $j \geq 1$, $k+1-j \leq k$. So, by the induction hypothesis, we know that the second player can finish the rest of the game with a win.

The induction step in this proof uses the fact that our claim $P(n)$ is true for a smaller value of n. But since we can't control how many matches the first player removes, we don't know how far back we have look in the sequence of earlier results $P(1) \ldots P(k)$. Our previous proof about postage can be rewritten so as to avoid strong induction. It's less clear how to rewrite proofs like this Nim example.

14 Prime factorization

Early in this course, we saw the "Fundamental Theorem of Arithmetic," which states that every positive integer n, $n \geq 2$, can be expressed as the product of one or more prime numbers. Let's prove that this is true.

Recall that a number n is prime if its only positive factors are one and n. n is composite if it's not prime. Since a factor of a number must be no larger than the number itself, this means that a composite number n always has a factor larger than 1 but smaller than n. This, in turn, means that we can write n as ab, where a and b are both larger than 1 but smaller than n.[3]

Proof by induction on n.

Base: 2 can be written as the product of a single prime number, 2.

Induction: Suppose that every integer between 2 and k can be written as the product of one or more primes. We need to show

[3] We'll leave the details of proving this as an exercise for the reader.

that $k+1$ can be written as a product of primes. There are two cases:

Case 1: $k+1$ is prime. Then it is the product of one prime, i.e. itself.

Case 2: $k+1$ is composite. Then $k+1$ can be written as ab, where a and b are integers such that a and b lie in the range $[2, k]$. By the induction hypothesis, a can be written as a product of primes $p_1 p_2 \ldots p_i$ and b can be written as a product of primes $q_1 q_2 \ldots q_j$. So then $k+1$ can be written as the product of primes $p_1 p_2 \ldots p_i q_1 q_2 \ldots q_j$.

In both cases $k+1$ can be written as a product of primes, which is what we needed to show.

Again, the inductive step needed to reach back some number of steps in our sequence of results, but we couldn't control how far back we needed to go.

15 Recursive definitions

Recursive function definitions in mathematics are basically similar to recursive procedures in programming languages. A recursive definition defines an object in terms of smaller objects of the same type. Because this process has to end at some point, we need to include explicit definitions for the smallest objects. So a recursive definition always has two parts:

- Base case or cases
- Recursive formula/step

Recursive definitions are sometimes called inductive definitions or (especially for numerical functions) recurrence relations. Folks who call them "recurrence relations" typically use the term "initial condition" to refer to the base case.

For example, recall that the factorial function $n!$ is defined by $n! = n \times (n-1) \times \ldots \times 2 \times 1$. We can define the factorial function $n!$ recursively:

- $0! = 1$
- $n! = n \cdot (n-1)!$

The recursive definition gets rid of the annoyingly informal "..." and, therefore, tends to be easier to manipulate in formal proofs or in computer programs.

Notice that the base and inductive parts of these definitions aren't explicitly labelled. This is very common for recursive definitions. You're just expected to figure out those labels for yourself.

Here's another recursive definition of a familiar function:

- $g(1) = 1$
- $g(n) = g(n-1) + n$

This is just another way of defining the summation $\sum_{i=1}^{n} n$. This particular recursive function happens to have a nice closed form: $\frac{n(n+1)}{2}$. Some recursively-defined functions have a nice closed form and some don't, and it's hard to tell which by casual inspection of the recursive definition.

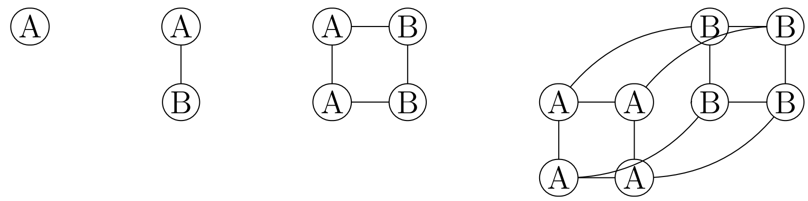
Notice that both the base case and the inductive equation must be present to have a complete definition. For example, if we leave off the base case in the definition of g, there are quite a lot of different functions that would satisfy the inductive condition.

16 Hypercubes

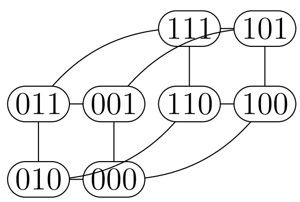
An n-cube or a hypercube Q_n is the graph of the corners and edges of an n-dimensional cube. It is defined recursively as follows (for any $n \in \mathbb{N}$):

1. Q_0 is a single vertex with no edges
2. Q_n consists of two copies of Q_{n-1} with edges joining corresponding vertices.

That is, each node v_i in one copy of Q_{n-1} is joined by an edge to its clone copy v_i' in the second copy of Q_{n-1}. Q_0, Q_1, Q_2, and Q_3 look as follows. The node labels distinguish the two copies of $Qn - 1$



The hypercube defines a binary coordinate system. To build it, we label nodes with binary numbers, where each binary digit corresponds to the value of one coordinate. The edges connect nodes that differ in exactly one coordinate.



Q^n has 2^n nodes. To compute the number of edges, we set up the following recursive definition for the number of edges $E(n)$ in the Q_n:

1. $E(0) = 0$
2. $E(n) = 2E(n-1) + 2^{n-1}$

The 2^{n-1} term is the number of nodes in each copy of Q^{n-1}, i.e. the number of edges required to join corresponding nodes. We'll leave it as an exercise to find a closed form for this recurrence.

17 More interesting sorts of recursion

The true power of recursive definition is revealed when the result for n depends on the results for more than one smaller value, as in the strong induction examples. For example, the famous Fibonacci numbers are defined by:

- $F_0 = 0$, $F_1 = 1$
- $\forall i \in \mathbb{N}, i \geq 2, F_i = F_{i-1} + F_{i-2}$

So $F_2 = 1$, $F_3 = 2$, $F_4 = 3$, $F_5 = 5$, $F_6 = 8$, $F_7 = 13$, $F_8 = 21$, $F_9 = 34$.

Recursion is also often the best approach when the value for n depends on the value for $n-1$ in a way that's more complicated than simple sums or products. For example, we could define a function f by:

- $f(0) = 3$
- $f(n) = 2f(n-1) + 3$

That is, $f(0) = 3$, $f(1) = 9$, $f(2) = 21$, $f(3) = 45$, etc. It's not instantly obvious how to rewrite this definition using a sum with "...".

These more complicated recursive definitions frequently do not have a closed form. Nor is it easy to write them (even informally) using "...".

18 Proofs with recursive definitions

Recursive definitions are ideally suited to inductive proofs. The main outline of the proof often mirrors the structure of the definition.

For example, let's prove the following claim about the Fibonacci numbers:

Claim 8 *For any $n \geq 0$, F_{3n} is even.*

Let's check some concrete values: $F_0 = 0$, $F_3 = 2$, $F_6 = 8$, $F_9 = 34$. All are even. Claim looks good. So, let's build an inductive proof:

Proof: by induction on n.

Base: $F_0 = 0$, which is even.

Induction: Suppose that F_{3k} is even. We need to show that that $F_{3(k+1)}$ is even.

$F_{3(k+1)} = F_{3k+3} = F_{3k+2} + F_{3k+1}$

But $F_{3k+2} = F_{3k+1} + F_{3k}$. So, substituting into the above equation, we get:

$F_{3(k+1)} = (F_{3k+1} + F_{3k}) + F_{3k+1} = 2F_{3k+1} + F_{3k}$

By the inductive hypothesis F_{3k} is even. $2F_{3k+1}$ is even because it's 2 times an integer. So their sum must be even. So $F_{3(k+1)}$ is even, which is what we needed to show.

Some people feel a bit uncertain if the base case is a special case like zero. It's ok to also include a second base case. For this proof, you would check the case for $n = 1$ i.e. verify that F_3 is even. The extra base case isn't necessary for a complete proof, but it doesn't cause any harm and may help the reader.

Another example, again with the Fibonacci numbers:

Claim 9 *For any* $n \geq 1$, $\Sigma_{i=1}^{n} (F_i)^2 = (F_n)(F_{n+1})$.

For example $\Sigma_{i=1}^{4} (F_i)^2 = 1 + 1 + 4 + 9 = 15 = 3 \cdot 5 = F_4 F_5$.

I have no intuitions about such equations. So let's hope induction will work in a simple way and give it a try:

Proof: by induction on n.

Base: If $n = 1$, then we have $\Sigma_{i=1}^{1} (F_i)^2 = (F_1)^2 = 1 = 1 \cdot 1 = F_1 F_2$. So this checks out.

Induction: Suppose that $\Sigma_{i=1}^{k} (F_i)^2 = (F_k)(F_{k+1})$.

$\Sigma_{i=1}^{k+1} (F_i)^2 = (\Sigma_{i=1}^{k} (F_i)^2) + (F_{k+1})^2$.

By the inductive hypothesis, this is $F_k F_{k+1} + (F_{k+1})^2$.

But $F_k F_{k+1} + (F_{k+1})^2 = F_k F_{k+1} + F_{k+1} F_{k+1} = F_{k+1}(F_k + F_{k+1}) = F_{k+1} F_{k+2}$. (The last step is using the definition of the Fibonacci numbers.)

So $\Sigma_{i=1}^{k+1} (F_i)^2 = F_{k+1} F_{k+2}$, which is what we needed to show.

19 Variation in notation

Certain details of the induction outline vary, depending on the individual preferences of the author and the specific claim being proved. Some folks prefer to assume the statement is true for k and prove it's true for $k+1$. Other assume it's true for $k-1$ and prove it's true for k. You can pick either one, though one of the two choices may yield a slightly simpler proof for some specific problem.

Some authors prefer to write strong induction hypotheses all the time, even when a weak hypothesis would be sufficient. This saves mental effort, because you don't have to figure out in advance whether a strong hypothesis was really required. However, for some problems, a strong hypothesis may be more complicated to state than a weak one. Many beginners make technical mistakes writing strong hypotheses, even if they can correct write weak ones. You'll probably use the strong form more often as you become more fluent with mathematical writing.

Authors writing for more experienced audiences may abbreviate the outline somewhat, e.g. packing an entirely short proof into one paragraph without labelling the base and inductive steps separately. However, being careful about the outline is important when you are still getting used to the technique.