Notes on mathematical induction

Mathematical induction is a technique used to prove things about, say, the set of all non-negative integers.

1. Formulation

- (The principle of mathematical induction, first version) *Suppose that $P(n)$ is an assertion about the non-negative integer n. If (a) $P(0)$ is true; and (b) you can prove $P(n+1)$ under the assumption that $P(n)$ is true; then $P(n)$ is true for all non-negative integers n.*

The basic idea here is very simple. The result is true for 0 by (a). Since true for 0, true for 1 by (b); since true for 1, true for 2; etc.

There is a second version in which the logic seems a bit more complicated, but which is in practice much more useful.

- (The principle of mathematical induction, second version) *Suppose that $P(n)$ is an assertion about the non-negative integer n. If (a) $P(0)$ is true; and (b) you can prove $P(n)$ under the assumption that $P(m)$ is true for all $m < n$; then $P(n)$ is true for all non-negative integers n.*

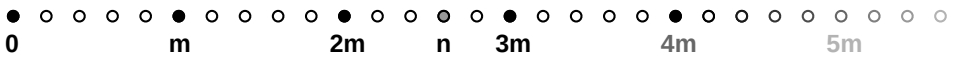
The result is true for 0 by (a). Since true for 0, true for 1 by (b); since true for 0 and 1, true for 2; etc. since true for 0, 1, and 2, true for 3; etc.

Let's look at an example of how it works.

- *Suppose n and m to be non-negative integers with $m > 0$. There exist q and r with $0 \leq r < m$ such that*

$$n = qm + r \ .$$

Let's first try to see what this says and why it's true, at least intuitively. Lay the non-negative integers out on a line, singling out the multiples of m:



These span the whole range of integers, and somewhere in there must lie a. So, at least intuitively, we have for some q

$$qm \leq a < (q+1)m \ .$$

But if $qm \leq a < (q+1)m$ then

$$0 \leq a - qm < m$$

and that's the remainder. So now we understand the result, and we want to prove it from more basic principles.

We first prove $P(0)$. This is easy: $0 = 0.m + 0$.

Now we assume $P(n_*)$ true for all $n_* < n$ (second form of mathematical induction). We set $n_* = n - m$. By the induction hypothesis

$$n_* = q_* m + r_*$$

but then

$$n = n_* + m = (q+1)m + r_* \ .$$

But wait! There's a minor problem we forgot about. It might happen that $n < m$ and then a_* is negative, so the induction assumption doesn't apply. That's OK, since if $n < m$ then $q = 0$ and $r = n$.

Exercise 1. *There is another possible proof that uses the first form of mathematical induction. Find it. Hint: if you know how to the quotient and remainder for n, what are they for $n+1$? There are two cases to be considered.*

The hardest part of proving something by mathematical induction is often deciding what the induction hypothesis $P(n)$ is to be. It has to be powerful enough to enable the next step, weak enough so as to be provable. This is a fine balance. Let's do another example, proving that the addition algorithm in base B is correct. Let's first recall how it works.

Suppose you want to add $a = a_{n-1} : \ldots : a_0$ and $b = b_{n-1} : \ldots : b_0$. (1) Set the initial carry $\varepsilon := 0$. (2) for $k = 0$ to $n- = 1$: $\varepsilon + a_k + b_k$ and represent it in base B as $c : d_k|$. Set the new carry $\varepsilon := c$. (3) Set d_n equal to the last carry. (4) At the end the sum is $d_n : \ldots : d_0$.

The rule gets more complicated as n increases, and basically you are just doing the same thing over and over n times, so induction on n is the natural idea. So the initial guess for the proposition is this: *If this algorithm is followed, then $d_n : \ldots : d_0$ is the base B expression for the sum.* What this means, neither more nor less, is that (a) the sum is equal to $d_nB^n + d_{n-1}B^{n-1} + \cdots + d_0$ and (b) $0 \leq d_k < B$ for all k. This is what we have to prove.

The case $n = 1$ is not too instructive. We add a_0 and b_0 where $0 \leq a_0 < B$ and $0 \leq b_0 < B$. We get a sum c with $0 \leq c < 2B$. Since c lies in this range, it as at most a two digit expression in base B, say $\varepsilon : d_0$. That's the answer.

For $n > 1$ In using the induction hypothesis, we have to think about the relationship between adding n digit numbers and adding $n+1$ digit numbers. That's easy—on the way to doing an $n+1$ digit addition we perform an n digit addition. So in adding $a_n : \ldots : a_0$ to $b_n : \ldots\ b_0$ we can assume that the addition of $a_{n-1} : \ldots : a_0$ to $b_{n-1} : \ldots\ b_0$ is OK. At this point we have the carry from this early part, ε. And we are next going to add $\varepsilon + a_n + b_n$.

At this point we realize there is a problem. We know that the old carry is either 0 or 1, but we haven't said anything about this, and we need to know it for the next step, for sure. *We must add that to the induction hypothesis.* It is certainly true for the case $n = 1$. which is OK, and gives a new carry also either 0 or 1. This proves the assertion about the carry, it proves that $0 \leq d_n < B$, and an extra thought or two gives us the claim that we actually have the sum as well (we are talking here about the n-th digit, so have to write down $a_nB^n + b_nB^n + \varepsilon B^n$.)

2. A mild variation

Sometimes it is not sufficient just to prove that the proposition is true for a single lowest value. For example, the Fibonacci numbers F_n are defined by the conditions: (1) $F_1 = F_2 = 1$, and (2) $F_{n+1} = F_n + F_{n-1}$. There is formula for the n-th Fibonacci number:

$$F_n = \frac{\Phi^n - (-1)^n/\Phi^n}{\sqrt{5}} \ .$$

The natural way to prove it is by induction, since the definition is by induction. But we need to show the formula to be true for F_1 and F_2 in order to get off the ground. In general, sometimes it is necessary to prove that something is valid for the first k possible values where $k > 1$, thena pply the second form of induction. This is entirely acceptable, since we start off by knowing $P(1), \ldots, P(k)$. The induction step tells us that $P(k+1)$ is true. But then we can also see that $P(k+2)$ is true. Etc. So the underlying reasoning behind induction still holds

3. Induction and recursion

Mathematical induction is a proof technique very strongly related to **recursion** in computation. Suppose you want to calculate the base 60 expression for some large number, say $1,000,451$. You divide by 60 and get

$$1,000,451 = 16674 \cdot 60 + 11 \ .$$

This tells you that the last digit in the expression is .. And now we divide the quotient 16674 by 60 again, etc. If we get for this quotient the expression $a_{n-1} : \ldots \ : a_1$ then the final answer we want is

$$a_{n-1} : \ldots \ : a_1 : 11$$

In other words, we find the base 60 expression for N by finding the base 60 expression for $\lfloor N/60 \rfloor$ and tacking it onto the left of the remainder after division by 60. We do this as long as the quotient doesn't vanish. This is recursion: the process has to stop because we are looking at smaller and smaller quotients at each step, until we get to 0, which can be handled directly. The logic is that of the second version of mathematical induction: we can do it for N because we can do it for $\lfloor N/60 \rfloor$, which is less than N. And we know explicitly what to do for $N = 0$. **The logical basis of recursion is mathematical induction.**