1.6 Proof technique 2 — Mathematical induction

Of the basic mathematical tools useful in algorithmics, perhaps none is more important than *mathematical induction*. Not only does it allow us to prove interesting statements about the correctness and efficiency of algorithms, but we shall see in Section 1.6.4 that it can even be used to *determine* the actual statements that need to be proved.

Before the technique is discussed, a digression on the nature of scientific discovery is in order. There are two contrasting fundamental approaches in science: *induction* and *deduction*. According to the *Concise Oxford Dictionary*, induction consists of "inferring of general law from particular instances", whereas a deduction is an "inference from general to particular". We shall see that even though induction can yield false conclusions, it is not to be sneezed at. Deduction, on the other hand, is always valid provided it is applied properly.

We cannot in general trust the outcome of inductive reasoning. As long as there are cases that have not been considered, it remains possible that the general rule induced is wrong. For instance, everyday experience may have convinced you inductively that "it is always possible to cram one more person into a trolley". But a moment's thought shows that this rule is absurd. As a more mathematical example, consider the polynomial $p(n) = n^2 + n + 41$. If you compute $p(0)$, $p(1)$, $p(2), \ldots, p(10)$, you find 41, 43, 47, 53, 61, 71, 83, 97, 113, 131 and 151. It is straightforward to verify that all these integers are prime numbers. Therefore, it is natural to infer *by induction* that $p(n)$ is prime for all integer values of n. But in fact $p(40) = 1681 = 41^2$ is composite. For a beautiful geometric example of induction gone wrong, we encourage you to work Problem 1.19.

A more striking example of misleading induction is given by a conjecture of Euler's, which he formulated in 1769. Is it possible for the sum of three fourth powers to be a fourth power? Formally, can you find four positive integers A, B, C and D such that

$$A^4 + B^4 + C^4 = D^4?$$

After failing to come up with even a single example of this behaviour, Euler conjectured that this equation can never be satisfied. (This conjecture is related to Fermat's Last Theorem.) More than two centuries elapsed before Elkies in 1987 discovered the first counterexample, which involved seven and eight-figure numbers. It has since been shown by Frye, using hundreds of hours of computing time on various Connection Machines, that the *only* counterexample with D less than one million is

$$95800^4 + 217519^4 + 414560^4 = 422481^4$$

(not counting the solution obtained by multiplying each of these numbers by 2). Note that 422481^4 is a 23-figure number.

Pell's equation provides an even more extreme case of compelling but incorrect inductive reasoning. Consider the polynomial $p(n) = 991n^2 + 1$. The question is whether there is a positive integer n such that $p(n)$ is a perfect square. If you try various values for n, you will find it increasingly tempting to assume inductively

that the answer is negative. But in fact a perfect square can be obtained with this polynomial: the *smallest* solution is obtained when

$$n = 12\,055\,735\,790\,331\,359\,447\,442\,538\,767.$$

In contrast, deductive reasoning is not subject to errors of this kind. Provided that the rule invoked is correct and that it applies to the situation under discussion, the conclusion reached is necessarily correct. Mathematically, if it is true that some statement $P(x)$ holds for each x in some set X, and if indeed y belongs to X, then the fact that $P(y)$ holds can be roundly asserted. This is not to say that we cannot infer something false using deductive reasoning. From a false premise, we can deductively derive a false conclusion; this is the principle underlying indirect proofs. For instance, if it is correct that $P(x)$ is true for all x in X, but we are careless in applying this rule to some y that does not belong to X, we may erroneously believe that $P(y)$ holds. Similarly, if our belief that $P(x)$ is true for all x in X is based on careless inductive reasoning, then $P(y)$ may be false even if indeed y belongs to X. In conclusion, deductive reasoning can yield a wrong result, but *only* if the rules that are followed are incorrect or if they are not followed properly.

As a computer science example, consider again multiplication *à la russe*, described in Section 1.2. If you try this algorithm on several pairs of positive integers, you will find that it gives the correct answer each time. By induction, you may formulate the conjecture that the algorithm is always correct. In this case, the conjecture reached inductively happens to be right: we shall prove rigorously (by deductive reasoning) the correctness of this algorithm with Theorem 1.6.4. Once correctness has been established, if you use the algorithm to multiply 981 by 1234 and obtain 1210554, you may conclude that

$$981 \times 1234 = 1210554.$$

Here, the correctness of this specific instance of integer multiplication is a special case of the correctness of the algorithm in general. Therefore the conclusion that $981 \times 1234 = 1210554$ is based on deductive reasoning. However, the proof of correctness of the algorithm says nothing about its behaviour on negative and fractional numbers, and therefore you cannot deduce anything about the result given by the algorithm if run on -12 and 83.7.

You may well wonder at this point why anyone would use error-prone induction rather than fool-proof deduction. There are two basic reasons for using induction in the process of scientific discovery. If you are a physicist whose goal is to determine the fundamental laws that govern the Universe, you *must* use an inductive approach: the rules you infer should reflect actual data obtained from experiments. Even if you are a theoretical physicist—such as Einstein—you still need actual experiments carried out by others. For instance, it was by inductive reasoning that Halley predicted the return of his eponymous comet and that Mendeleev predicted not only the existence of yet undiscovered chemical elements, but their chemical properties as well.

But surely, only deduction is legitimate in mathematics and rigorous computer science? After all, mathematical statements such as the fact that there are infinitely

many prime numbers (Theorem 1.5.1) and that multiplication *à la russe* is a correct algorithm (Theorem 1.6.4) can be proved in a rigorous deductive manner, without any need for experimental data. Inductive reasonings are to be banned from mathematics. Right? Wrong! In reality, mathematics is often very much an experimental science. It is not unusual that a mathematician will discover a mathematical truth by considering several special cases and inferring from them *by induction* a general rule that seems plausible. For instance, if I notice that

$$\begin{aligned} 1^3 &= 1 = 1^2 \\ 1^3+2^3 &= 9 = 3^2 \\ 1^3+2^3+3^3 &= 36 = 6^2 \\ 1^3+2^3+3^3+4^3 &= 100 = 10^2 \\ 1^3+2^3+3^3+4^3+5^3 &= 225 = 15^2, \end{aligned}$$

I may begin to suspect that the sum of the cubes of the first n positive integers is always a perfect square. It turns out in this case that inductive reasoning yields a correct law. If I am even more perceptive, I may realize that this sum of cubes is precisely the square of the sum of the first n positive integers; see Problem 1.21.

However, no matter how compelling the evidence becomes when more and more values of n are tried, a general rule of this sort cannot be asserted on the basis of inductive evidence only. The difference between mathematics and the inherently experimental sciences is that once a general mathematical law has been discovered by induction, we may hope to prove it rigorously by applying the deductive approach. Nevertheless, induction has its place in the mathematical process. Otherwise, how could you hope to prove rigorously a theorem whose statement has not even been formulated? To sum up, induction is necessary for formulating conjectures and deduction is equally necessary for proving them or sometimes disproving them. Neither technique can take the place of the other. Deduction alone is sufficient for "dead" or frozen mathematics, such as Euclid's *Elements* (perhaps history's highest monument to deductive mathematics, although much of its material was no doubt discovered by inductive reasoning). But induction is required to keep mathematics alive. As Pólya once said, "mathematics presented with rigor is a systematic deductive science but mathematics in the making is an experimental inductive science".

Finally, the punch line of this digression: one of the most useful *deductive* techniques available in mathematics has the misfortune to be called *mathematical induction*. This terminology is confusing, but we must live with it.

1.6.1 The principle of mathematical induction

Consider the following algorithm.

function $sq(n)$
  if $n = 0$ **then return** 0
  else return $2n + sq(n-1) - 1$

If you try it on a few small inputs, you find that

$$sq(0) = 0,\ sq(1) = 1,\ sq(2) = 4,\ sq(3) = 9,\ sq(4) = 16.$$

By induction, it seems obvious that $sq(n) = n^2$ for all $n \geq 0$, but how could this be proved rigorously? Is it even true? Let us say that the algorithm *succeeds* on integer n whenever $sq(n) = n^2$, and that it *fails* otherwise.

Consider any integer $n \geq 1$ and assume for the moment that the algorithm succeeds on $n - 1$. By definition of the algorithm, $sq(n) = 2n + sq(n-1) - 1$. By our assumption $sq(n-1) = (n-1)^2$. Therefore

$$sq(n) = 2n + (n-1)^2 - 1 = 2n + (n^2 - 2n + 1) - 1 = n^2 .$$

What have we achieved? We have proved that the algorithm must succeed on n whenever it succeeds on $n - 1$, provided $n \geq 1$. In addition, it clearly succeeds on $n = 0$.

The *principle of mathematical induction*, described below, allows us to infer from the above that the algorithm succeeds on all $n \geq 0$. There are two ways of understanding why this conclusion follows: constructively and by contradiction. Consider any positive integer m on which you wish to prove that the algorithm succeeds. For the sake of argument, assume that $m \geq 9$ (smaller values can be proved easily). We know already that the algorithm succeeds on 4. From the general rule that it must succeed on n whenever it succeeds on $n - 1$ for $n \geq 1$, we infer that it also succeeds on 5. Applying this rule again shows that the algorithm succeeds on 6 as well. Since it succeeds on 6, it must also succeed on 7, and so on. This reasoning continues as many times as necessary to arrive at the conclusion that the algorithm succeeds on $m - 1$. Finally, since it succeeds on $m - 1$, it must succeed on m as well. It is clear that we could carry out this reasoning explicitly—with no need for "and so on"—for any fixed positive value of m.

If we prefer a single proof that works for all $n \geq 0$ and that does not contain and-so-on's, we must accept the *axiom of the least integer*, which says that every nonempty set of positive integers contains a smallest element; see Problem 1.24. The axiom allows us to use this smallest number as a foundation from which to prove theorems.

Now, to prove the correctness of the algorithm, assume *for a contradiction* that there exists at least one positive integer on which the algorithm fails. Let n stand for the smallest such integer, which exists by the axiom of the least integer. Firstly, n must be greater than or equal to 5 since we have already verified that $sq(i) = i^2$ when $i = 1, 2, 3$ or 4. Secondly, the algorithm must succeed on $n - 1$ for otherwise n would not be the smallest positive integer on which it fails. But this implies by our general rule that the algorithm also succeeds on n, which contradicts our assumption about the choice of n. Therefore such an n cannot exist, which means that the algorithm succeeds on every positive integer. Since we also know that the algorithm succeeds on 0, we conclude that $sq(n) = n^2$ for all integers $n \geq 0$.

We now spell out a simple version of the principle of mathematical induction, which is sufficient in many cases. A more powerful version of the principle is given in Section 1.6.3. Consider any property P of the integers. For instance, $P(n)$ could be "$sq(n) = n^2$", or "the sum of the cubes of the first n integers is equal to the square of the sum of those integers", or "$n^3 < 2^n$". The first two properties

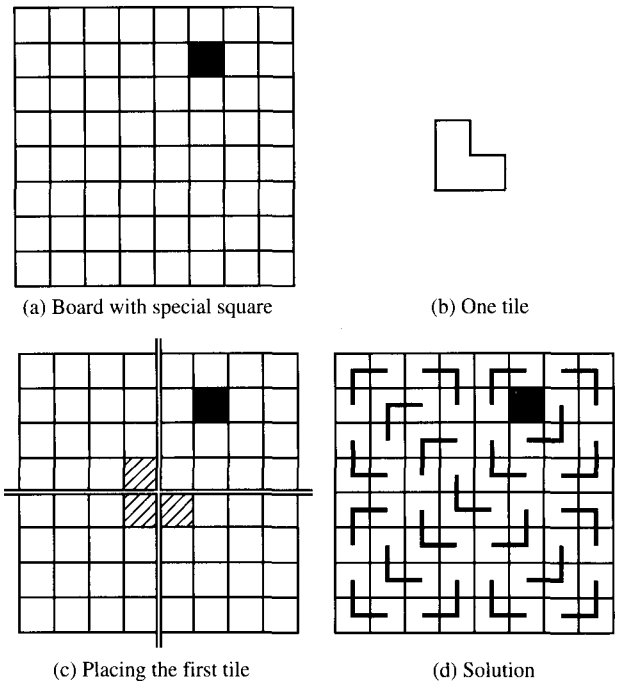
hold for every $n \geq 0$, whereas the third holds provided $n \geq 10$. Consider also an integer a, known as the *basis*. If

1. $P(a)$ holds and
2. $P(n)$ must hold whenever $P(n-1)$ holds, for each integer $n > a$,

then property $P(n)$ holds for all integers $n \geq a$. Using this principle, we could assert that $sq(n) = n^2$ for all $n \geq 0$, immediately after showing that $sq(0) = 0 = 0^2$ and that $sq(n) = n^2$ whenever $sq(n-1) = (n-1)^2$ and $n \geq 1$.

Our first example of mathematical induction showed how it can be used to prove rigorously the correctness of an algorithm. As a second example, let us see how proofs by mathematical induction can sometimes be *turned into* algorithms. This example is also instructive as it makes explicit the proper way to write a proof by mathematical induction. The discussion that follows stresses the important points common to all such proofs.

Consider the following tiling problem. You are given a board divided into equal squares. There are m squares in each row and m squares in each column, where m is a power of 2. One arbitrary square of the board is distinguished as *special*; see Figure 1.5(a).



(a) Board with special square (b) One tile

(c) Placing the first tile (d) Solution

Figure 1.5. The tiling problem

You are also given a supply of tiles, each of which looks like a 2×2 board with one square removed, as illustrated in Figure 1.5(b). Your puzzle is to cover the board

with these tiles so that each square is covered exactly once, with the exception of the special square, which is not covered at all. Such a covering is called a *tiling*. Figure 1.5(d) gives a solution to the instance given in Figure 1.5(a).

Theorem 1.6.1 *The tiling problem can always be solved.*

Proof The proof is by mathematical induction on the integer n such that $m = 2^n$.

- ◇ *Basis*: The case $n = 0$ is trivially satisfied. Here $m = 1$, and the 1×1 "board" is a single square, which is necessarily special. Such a board is tiled by doing nothing! (If you do not like this argument, check the next simplest case: if $n = 1$, then $m = 2$ and any 2×2 board from which you remove one square looks exactly like a tile by definition.)
- ◇ *Induction step*: Consider any $n \geq 1$. Let $m = 2^n$. Assume the *induction hypothesis* that the theorem is true for $2^{n-1} \times 2^{n-1}$ boards. Consider an $m \times m$ board, containing one arbitrarily placed special square. Divide the board into 4 equal sub-boards by halving it horizontally and vertically. The original special square now belongs to exactly one of the sub-boards. Place one tile in the middle of the original board so as to cover exactly one square of each of the other three sub-boards; see Figure 1.5(c). Call each of the three squares thus covered "special" for the corresponding sub-board. We are left with four $2^{n-1} \times 2^{n-1}$ sub-boards, each containing one special square. By our induction hypothesis, each of these sub-boards can be tiled. The final solution is obtained by combining the tilings of the sub-boards together with the tile placed in the middle of the original board.

Since the theorem is true when $m = 2^0$, and since its truth for $m = 2^n$ follows from its assumed truth for $m = 2^{n-1}$ for all $n \geq 1$, it follows from the principle of mathematical induction that the theorem is true for all m provided m is a power of 2. ■

The reader should have no difficulty in transforming this proof of a mathematical theorem into an algorithm for performing the actual tiling (perhaps not a computer algorithm, but at least an algorithm suitable for "hand processing"). This tiling algorithm follows the general template known as divide-and-conquer, which we encountered in Section 1.2, and which we study at length in Chapter 7. This situation is not unusual when a theorem is proved constructively by mathematical induction.

Let us now look in detail at all the aspects of a well-formed proof by mathematical induction, such as the one above. Consider again an abstract property P of the integers, an integer a, and assume you wish to prove that $P(n)$ holds for all $n \geq a$. You must begin your proof with the *basis step*, which consists of proving that $P(a)$ holds. This basis step is usually easy, sometimes even trivial, but it is crucial that it be carried out properly; otherwise, the whole "proof" is literally without foundation.

The basis step is followed by the *induction step*, which is usually more substantial. This should start with "consider any $n > a$" (or equivalently "consider any $n \geq a + 1$"). It should continue with an explicit statement of the *induction hypothesis*, which essentially states that we assume $P(n-1)$ to hold. At that point, it remains to prove that we can infer that $P(n)$ holds assuming the induction hypothesis. Finally, an additional sentence such as the one at the end of the proof of Theorem 1.6.1 can be inserted to conclude the reasoning, but this is generally unnecessary.

Concerning the induction hypothesis, it is important to understand that we assume that $P(n-1)$ holds on a *provisional* basis; we do not really know that it holds until the theorem has been proved. In other words, the point of the induction step is to prove that the truth of $P(n)$ would follow logically from that of $P(n-1)$, regardless of whether or not $P(n-1)$ actually holds. If in fact $P(n-1)$ does *not* hold, the induction step does not allow us to conclude anything about the truth of $P(n)$.

For instance, consider the statement "$n^3 < 2^n$", which we shall denote $P(n)$. For positive integer n, it is easy to show that $n^3 < 2 \times (n-1)^3$ if and only if $n \geq 5$. Consider any $n \geq 5$ and provisionally assume that $P(n-1)$ holds. Now

$$\begin{array}{lcll} n^3 & < & 2 \times (n-1)^3 & \text{because } n \geq 5 \\ & < & 2 \times 2^{n-1} & \text{by the assumption that } P(n-1) \text{ holds} \\ & = & 2^n. & \end{array}$$

Thus we see that $P(n)$ follows logically from $P(n-1)$ whenever $n \geq 5$. Nevertheless $P(4)$ does *not* hold (it would say $4^3 < 2^4$, which is $64 < 16$) and therefore nothing can be inferred concerning the truth of $P(5)$. By trial and error, we find however that $P(10)$ does hold ($10^3 = 1000 < 2^{10} = 1024$). Therefore, it *is* legitimate to infer that $P(11)$ holds as well, and from the truth of $P(11)$ it follows that $P(12)$ holds also, and so on. By the principle of mathematical induction, since $P(10)$ holds and since $P(n)$ follows from $P(n-1)$ whenever $n \geq 5$, we conclude that $n^3 < 2^n$ is true for all $n \geq 10$. It is instructive to note that $P(n)$ holds also for $n = 0$ and $n = 1$, but that we cannot use these points as the basis of the mathematical induction because the induction step does not apply for such small values of n.

It may happen that the property to be proved is not concerned with the set of all integers not smaller than a given basis. Our tiling puzzle, for instance, concerns only the set of integers that are powers of 2. Sometimes, the property does not concern integers at all. For instance, it is not unusual in algorithmics to wish to prove a property of graphs. (It could even be said that our tiling problem is not *really* concerned with integers, but rather with boards and tiles, but that would be hairsplitting.) In such cases, if simple mathematical induction is to be used, the property to be proved should first be transformed into a property of the set of all integers not smaller than some basis point. (An alternative approach is given in Section 1.6.3.) In our tiling example, we proved that $P(m)$ holds for all powers of 2 by proving that $Q(n)$ holds for all $n \geq 0$, where $Q(n)$ is equivalent to $P(2^n)$. When this transformation is necessary, it is customary to begin the proof (as we did) with the words "The proof is by mathematical induction *on such-and-such a parameter*". Thus we find proofs on the number of nodes in a graph, on the length of a character string, on the depth of a tree, and so on.

There is one aspect of proofs by mathematical induction that most beginners find puzzling, if not downright paradoxical: *it is sometimes easier to prove a stronger statement than a weaker one!* We illustrate this with an example that we have already encountered. We saw that it is easy to conjecture by induction (not *mathematical* induction) that the sum of the cubes of the first n integers is always a perfect square. Proving this by mathematical induction is not easy. The difficulty is that an induction hypothesis like "the sum of the cubes of the first $n-1$ integers is a square" is not much help in proving that this is also the case for the first n integers because it does not say *which* square: in general, there is no reason to believe that a square is obtained when n^3 is added to another square. In contrast, it is easier to prove the stronger theorem that our sum of cubes is precisely the square of the sum of the first n integers: the induction hypothesis is now much more meaningful; see Problem 1.21.

1.6.2 A horse of a different colour

The most common pitfall in the design of proofs by mathematical induction deserves a subsection of its own. Consider the following absurd "theorem".

Theorem 1.6.2 *All horses are the same colour.*

Proof We shall prove that any set of horses contains only horses of a single colour. In particular, this will be true of the set of all horses. Let $\mathcal{H}$ be an arbitrary set of horses. Let us prove by mathematical induction on the number n of horses in $\mathcal{H}$ that they are all the same colour.

- ◇ *Basis*: The case $n = 0$ is trivially true: if there are no horses in $\mathcal{H}$, then surely they are all the same colour! (If you do not like this argument, check the next simplest case: if $n = 1$, then there is only one horse in $\mathcal{H}$, and again it is vacuously clear that "they" are "all" the same colour.)
- ◇ *Induction step*: Consider any number n of horses in $\mathcal{H}$. Call these horses $h_1, h_2, \ldots, h_n$. Assume the induction hypothesis that any set of $n-1$ horses contains only horses of a single colour (but of course the horses in one set could a priori be a different colour from the horses in another). Let $\mathcal{H}_1$ be the set obtained by removing horse h_1 from $\mathcal{H}$, and let $\mathcal{H}_2$ be defined similarly; see Figure 1.6.

$$\begin{array}{lccccc} \mathcal{H}_1: & & h_2 & h_3 & h_4 & h_5 \\ \mathcal{H}_2: & h_1 & & h_3 & h_4 & h_5 \end{array}$$

Figure 1.6. Horses of the same colour ($n = 5$)

There are $n-1$ horses in each of these two new sets. Therefore, the induction hypothesis applies to them. In particular, all the horses in $\mathcal{H}_1$ are of a single colour, say c_1, and all the horses in $\mathcal{H}_2$ are also of a single (possibly different) colour, say c_2. But is it really possible for colour c_1 to be different from

colour c_2? Surely not, since horse h_n belongs to both sets and therefore both c_1 and c_2 must be the colour of that horse! Since all the horses in $\mathcal{H}$ belong to either $\mathcal{H}_1$ or $\mathcal{H}_2$ (or both), we conclude that they are all the same colour $c = c_1 = c_2$. This completes the induction step and the proof by mathematical induction. ■

Before you continue, figure out the fallacy in the above "proof". If you think the problem is that our induction hypothesis ("any set of $n - 1$ horses must contain only horses of a single colour") was absurd, think again!

Solution: The problem is that "h_n belongs to both sets" is *not* true for $n = 2$ since h_2 does *not* belong to $\mathcal{H}_2$! Our reasoning was impeccable for the basis cases $n = 0$ and $n = 1$. Moreover, it is true that our theorem follows for sets of n horses assuming that it is true for $n - 1$, *but only when* $n \geq 3$. We can go from 2 to 3, from 3 to 4, and so on, but *not* from 1 to 2. Since the basis cases contain only 0 and 1, and since we are not allowed to go from 1 to 2, the induction step cannot get started. This small missing link in the proof is enough to invalidate it completely. We encountered a similar situation when we proved that $n^3 < 2^n$: the induction step did not apply for $n < 5$, and thus the fact that the statement is true for $n = 0$ and $n = 1$ was irrelevant. The important difference was that $n^3 < 2^n$ is true for $n = 10$, and therefore also for all larger values of n.

1.6.3 Generalized mathematical induction

The principle of mathematical induction described so far is appropriate for proving many interesting statements. There are cases, however, when a slightly more powerful principle is preferable. This is known as *generalized* mathematical induction. The situation is illustrated by the following example.

Suppose you wish to prove that every composite integer can be expressed as a product of prime numbers. (The *fundamental theorem of arithmetic* tells us that this decomposition is unique; this is *not* what we are trying to prove here.) Let us not worry about the basis of the mathematical induction yet, but rather let us jump right into the induction step. When trying to prove that n can be expressed as a product of prime numbers (assuming that it is composite), the "natural" induction hypothesis would be that $n - 1$ can be so decomposed. However, we challenge the reader to find anything in the prime decomposition of $n - 1$ that can be useful or even relevant to the prime decomposition of n. What we really need is the stronger induction hypothesis that *every* composite integer smaller than n can be decomposed into a product of prime numbers. The correct proof of our theorem is given below as Theorem 1.6.3, after we state formally the generalized principle of mathematical induction.

Another useful generalization concerns the basis. It is sometimes necessary to prove an *extended* basis, that is to prove the basis on more than one point. Note that we proved extended bases for the correctness of the *sq* algorithm and for the tiling problem, but it was a luxury: the induction step could really have been applied to prove the case $n = 1$ from the basis $n = 0$. Such is not always the case: sometimes we must prove independently the validity of several basis points before the induction step can take off. We shall encounter examples of this behaviour later in this book; see Problems 1.27 and 1.28 for instance.

We are now ready to formulate a more general principle of mathematical induction. Consider any property P of the integers, and two integers a and b such that $a \leq b$. If

1. $P(n)$ holds for all $a \leq n < b$ and
2. for any integer $n \geq b$, the fact that $P(n)$ holds follows from the assumption that $P(m)$ holds for all m such that $a \leq m < n$,

then property $P(n)$ holds for all integers $n \geq a$.

Yet a further generalization of the principle of mathematical induction is convenient when we are not interested in proving a statement about *every* integer not smaller than the basis. It is often the case that we wish to prove that some property P holds, but only for those integers for which some other property Q holds as well. We have seen two examples of this situation already: the tiling problem applies only when m is a power of 2, and our statement about prime decomposition applies only to composite numbers (although we could extend it in a natural way to prime numbers). When this occurs, it suffices to mention Q explicitly in the statement of the theorem to be proved, to prove the (possibly extended) basis only for points on which Q applies, and to prove the induction step also only on those points. Of course, the induction hypothesis will be similarly weakened. Consider any n beyond the basis such that $Q(n)$ holds. To prove that $P(n)$ holds, you are only entitled to assume that $P(m)$ holds when $a \leq m < n$ and when $Q(m)$ holds as well. In our tiling example, we are allowed to use the induction hypothesis to tile 4×4 boards when proving that an 8×8 board can be tiled, but we are *not* allowed to assume that a 5×5 board can be tiled.

Before we illustrate this principle, note that it allows the basis to be empty. This happens when $a = b$ because in that case there are no integers n such that $a \leq n < b$. It can also happen when $a < b$ if $Q(n)$ never holds when $a \leq n < b$. This does not invalidate the proof because in such a case the validity of $P(n)$ for the smallest n on which the induction step applies is proved under an empty induction hypothesis, which is to say that it is proved without any assumptions at all. Our first example illustrates this. The second shows how to prove the correctness of an algorithm by generalized mathematical induction.

Theorem 1.6.3 *Every positive composite integer can be expressed as a product of prime numbers.*

Proof The proof is by generalized mathematical induction. In this case, there is no need for a basis.

◇ *Induction step*: Consider any composite integer $n \geq 4$. (Note that 4 is the smallest positive composite integer, hence it would make no sense to consider smaller values of n.) Assume the induction hypothesis that any positive composite integer smaller than n can be expressed as a product of prime numbers. (In the

smallest case $n = 4$, this induction hypothesis is vacuous.) Consider the smallest integer d that is larger than 1 and that is a divisor of n. As argued in the proof of Theorem 1.5.1, d is necessarily prime. Let $m = n/d$. Note that $1 < m < n$ because n is composite and $d > 1$. There are two cases.

- If m is prime, we have decomposed n as the product of two primes: $n = d \times m$.
- If m is composite, it is positive and smaller than n, and therefore the induction hypothesis applies: m can be expressed as a product of prime numbers, say $m = p_1 p_2 \cdots p_k$. Therefore $n = d \times m$ can be expressed as $n = d p_1 p_2 \cdots p_k$, also a product of prime numbers.

In either case, this completes the proof of the induction step and thus of the theorem. ■

Until now, the induction hypothesis was always concerned with a finite set of instances (exactly one for simple mathematical induction, usually many but sometimes none for generalized mathematical induction). In our final example of proof by generalized mathematical induction, the induction hypothesis covers an infinity of cases even when proving the induction step on a finite instance! This time, we shall prove that multiplication *à la russe* correctly multiplies any pair of positive integers. The key observation is that the tableau produced when multiplying 490 by 2468 is almost identical to Figure 1.2, which was used to multiply 981 by 1234. The only differences are that the first line is missing when multiplying 490 by 2468 and that consequently the term 1234 found in that first line is not added into the final result; see Figure 1.7. What is the relationship between instances (981 1234) and (490 2468)? Of course, it is that $490 = 981 \div 2$ and $2468 = 2 \times 1234$.

981	1234	1234			
490	2468		490	2468	
245	4936	4936	245	4936	4936
122	9872		122	9872	
61	19744	19744	61	19744	19744
30	39488		30	39488	
15	78976	78976	15	78976	78976
7	157952	157952	7	157952	157952
3	315904	315904	3	315904	315904
1	631808	631808	1	631808	631808
		1210554			1209320

Figure 1.7. Proving multiplication *à la russe*

Theorem 1.6.4 *Multiplication à la russe correctly multiplies any pair of positive integers.*

Proof Suppose we wish to multiply m by n. The proof is by mathematical induction on the value of m.

- ◇ *Basis*: The case $m = 1$ is easy: we have only one row consisting of 1 in the left-hand column and n in the right-hand column. That row is not crossed out since 1 is not even. When we "add up" the only number that "remains" in the right-hand column, we obviously get n, which is the correct result of multiplying 1 by n.
- ◇ *Induction step*: Consider any $m \geq 2$ and any positive integer n. Assume the induction hypothesis that multiplication *à la russe* correctly multiplies s by t for any positive integer s smaller than m and for any positive integer t. (Note that we do *not* require t to be smaller than n.) There are two cases to consider.
  - If m is even, the second row in the tableau obtained when multiplying m by n contains $m/2$ in the left-hand column and $2n$ in the right-hand column. This is identical to the first row obtained when multiplying $m/2$ by $2n$. Because any noninitial row in these tableaux depends only on the previous row, the tableau obtained when multiplying m by n is therefore identical to the tableau obtained when multiplying $m/2$ by $2n$, except for its additional first row, which contains m in the left-hand column and n in the right-hand column. Since m is even, this additional row will be crossed out before the final addition. Therefore, the final result obtained when multiplying m by n *à la russe* is the same as when multiplying $m/2$ by $2n$. But $m/2$ is positive and smaller than m. Thus, the induction hypothesis applies: the result obtained when multiplying $m/2$ by $2n$ *à la russe* is $(m/2)\times(2n)$ as it should be. Therefore, the result obtained when multiplying m by n *à la russe* is also equal to $(m/2)\times(2n) = mn$ as it should be.
  - The case when m is odd is similar, except that $m/2$ must be replaced throughout by $(m-1)/2$ and the first row when multiplying m by n is not crossed out. Therefore the final result of multiplying m by n *à la russe* is equal to n plus the result of multiplying $(m-1)/2$ by $2n$ *à la russe*. By the induction hypothesis, the latter is correctly computed as $((m-1)/2)\times 2n$, and thus the former is computed as $n + ((m-1)/2)\times 2n$, which is mn as it should be.

This completes the proof of the induction step and thus of the theorem. ■

1.6.4 Constructive induction

Mathematical induction is used primarily as a proof technique. Too often, it is employed to prove assertions that seem to have been produced from nowhere like a rabbit out of a hat. While the truth of these assertions is thus established, their origin remains mysterious. However, mathematical induction is a tool sufficiently powerful to allow us to discover not merely the truth of a theorem, but also its precise statement. By applying the technique of *constructive* induction described in this section, we can simultaneously prove the truth of a partially specified assertion and discover the missing specifications thanks to which the assertion is correct. We illustrate this technique with two examples featuring the *Fibonacci sequence*,

defined below. The second example shows how the technique can be useful in the analysis of algorithms.

The sequence named for Fibonacci, an Italian mathematician of the twelfth century, is traditionally introduced in terms of rabbits (although this time not out of a hat). This is how Fibonacci himself introduced it in his *Liberabaci*, published in 1202. Suppose that every month a breeding pair of rabbits produce a pair of offspring. The offspring will in their turn start breeding two months later, and so on. Thus if you buy a pair of baby rabbits in month 1, you will still have just one pair in month 2. In month 3 they will start breeding, so you now have two pairs; in month 4 they will produce a second pair of offspring, so you now have three pairs; in month 5 both they and their first pair of offspring will produce baby rabbits, so you now have five pairs; and so on. If no rabbits ever die, the number of pairs you have each month will be given by the terms of the Fibonacci sequence, defined more formally by the following recurrence:

$$\begin{cases} f_0 = 0; f_1 = 1 & \text{and} \\ f_n = f_{n-1} + f_{n-2} & \text{for } n \geq 2. \end{cases}$$

The sequence begins $0, 1, 1, 2, 3, 5, 8, 13, 21, 34 \ldots$ It has numerous applications in computer science, in mathematics, and in the theory of games. De Moivre obtained the following formula, which is easy to prove by mathematical induction (see Problem 1.27):

$$f_n = \frac{1}{\sqrt{5}}[\phi^n - (-\phi)^{-n}],$$

where $\phi = (1 + \sqrt{5})/2$ is the *golden ratio*. Since $0 < \phi^{-1} < 1$, the term $(-\phi)^{-n}$ can be neglected when n is large. Hence the value of f_n is roughly $\phi^n/\sqrt{5}$, which is exponential in n.

But where does de Moivre's formula come from? In Section 4.7 we shall see a general technique for solving Fibonacci-like recurrences. In the meantime, assume you do not know any such techniques, nor do you know de Moivre's formula, yet you would like to have an idea of the behaviour of the Fibonacci sequence. If you compute the sequence for a while, you soon discover that it grows quite rapidly (f_{100} is a 21-figure number). Thus, the conjecture "the Fibonacci sequence grows exponentially fast" is reasonable. How would you prove it? The difficulty is that this conjecture is too vague to be proved directly by mathematical induction: remember that it is often easier to prove a stronger theorem than a weaker one. Let us therefore guess that there exists a real number $x > 1$ such that $f_n \geq x^n$ for each sufficiently large integer n. (This statement could not possibly be true for *every* positive integer n since it obviously fails on $n \leq 2$.) In symbols,

Conjecture: $$(\exists\, x > 1)\,(\overset{\infty}{\forall}\, n \in \mathbb{N})\,[f_n \geq x^n].$$

There are two unknowns in the theorem we wish to prove: the value of x and the precise meaning of "for each sufficiently large". Let us not worry about the latter for the time being. Let $P_x(n)$ stand for "$f_n \geq x^n$". Consider any sufficiently large integer n. The approach by constructive induction consists of asking ourselves for which values of x $P_x(n)$ follows from the *partially specified induction hypothesis* that

$P_x(m)$ holds for each integer m that is less than n but that is still sufficiently large. Using the definition of the Fibonacci sequence and this hypothesis, and provided $n-1$ and $n-2$ are also "sufficiently large",

$$f_n = f_{n-1} + f_{n-2} \geq x^{n-1} + x^{n-2} = (x^{-1} + x^{-2})\, x^n.$$

To conclude that $f_n \geq x^n$, we need $x^{-1} + x^{-2} \geq 1$, or equivalently $x^2 - x - 1 \leq 0$. By elementary algebra, since we are only interested in the case $x > 1$, solving this quadratic equation implies that $1 < x \leq \phi = (1 + \sqrt{5})/2$.

We have established that $P_x(n)$ follows from $P_x(n-1)$ and $P_x(n-2)$ provided $1 < x \leq \phi$. This corresponds to proving the induction step in a proof by mathematical induction. To apply the principle of mathematical induction and conclude that the Fibonacci sequence grows exponentially fast, we must also take care of the basis. In this case, because the truth of $P_x(n)$ depends only on that of $P_x(n-1)$ and $P_x(n-2)$, it is sufficient to verify that property P_x holds on two consecutive positive integers to assert that it holds from that point on.

It turns out that there are no integers n such that $f_n \geq \phi^n$. However, finding two consecutive integers on which property P holds is easy for any x strictly smaller than ϕ. For instance, both $P_x(11)$ and $P_x(12)$ hold when $x = 3/2$. Therefore, $f_n \geq (\frac{3}{2})^n$ for all $n \geq 11$. This completes the proof that the Fibonacci sequence grows *at least* exponentially. The same process can be used to prove that it grows *no faster* than exponentially: $f_n \leq y^n$ for every positive integer n provided $y \geq \phi$. Here again, the condition on y is not God-given: it is obtained by constructive induction when trying to find constraints on y that make the induction step go through. Putting those observations together, we conclude that f_n grows exponentially; more precisely, it grows like a power of a number close to ϕ. The remarkable thing is that we can reach this conclusion with no need for an explicit formula such as de Moivre's.

Our second example of constructive induction concerns the analysis of the obvious algorithm for computing the Fibonacci sequence.

```
function Fibonacci(n)
    if n < 2 then return n
    else return Fibonacci(n − 1)+Fibonacci(n − 2)        (⋆)
```

Let $g(n)$ stand for the number of times instruction $(\star)$ is performed when *Fibonacci*(n) is called (counting the instructions performed in recursive calls). This function is interesting because $g(n)$ gives a bound on the time required by a call on *Fibonacci*(n).

Clearly, $g(0) = g(1) = 0$. When $n \geq 2$, instruction $(\star)$ is executed once at the top level, and $g(n-1)$ and $g(n-2)$ times by the first and second recursive calls, respectively. Therefore,

$$\begin{cases} g(0) = g(1) = 0 & \text{and} \\ g(n) = g(n-1) + g(n-2) + 1 & \text{for } n \geq 2. \end{cases}$$

This formula is similar to the recurrence that defines the Fibonacci sequence itself. It is therefore reasonable to conjecture the existence of positive real constants

a and b such that $af_n \leq g(n) \leq bf_n$ for each sufficiently large integer n. Using constructive induction, it is straightforward to find that $af_n \leq g(n)$ holds for each sufficiently large n provided it holds on two consecutive integers, regardless of the value of a. For instance, taking $a = 1$, $f_n \leq g(n)$ holds for all $n \geq 2$.

However when we try to prove the other part of our conjecture, namely that there exists a b such that $g(n) \leq bf_n$ for each sufficiently large n, we run into trouble. To see what happens, let $P_b(n)$ stand for "$g(n) \leq bf_n$", and consider any sufficiently large integer n (to be made precise later). We wish to determine conditions on the value of b that make $P_b(n)$ follow from the hypothesis that $P_b(m)$ holds for each sufficiently large $m < n$. Using the definition of the Fibonacci sequence and this partially specified induction hypothesis, and provided $n - 1$ and $n - 2$ are also sufficiently large,

$$g(n) = g(n-1) + g(n-2) + 1 \leq bf_{n-1} + bf_{n-2} + 1 = bf_n + 1,$$

where the last equality comes from $f_n = f_{n-1} + f_{n-2}$. Thus, we infer that $g(n) \leq bf_n + 1$, but not that $g(n) \leq bf_n$. Regardless of the value of b, we cannot make the induction step work!

Does this mean the original conjecture was false, or merely that constructive induction is powerless to prove it? The answer is: neither. The trick is to use constructive induction to prove there exist positive real constants b and c such that $g(n) \leq bf_n - c$ for each sufficiently large n. This may seem odd, since $g(n) \leq bf_n - c$ is a stronger statement than $g(n) \leq bf_n$, which we were unable to prove. We may hope for success, however, on the ground that if the statement to be proved is stronger, then so too is the induction hypothesis it allows us to use; see the end of Section 1.6.1.

Consider any sufficiently large integer n. We must determine for which values of b and c the truth of $g(n) \leq bf_n - c$ follows from the partially specified induction hypothesis that $g(m) \leq bf_m - c$ for each sufficiently large $m < n$. Using the definition of the Fibonacci sequence and this hypothesis, and provided $n - 1$ and $n - 2$ are also sufficiently large,

$$\begin{aligned} g(n) &= g(n-1) + g(n-2) + 1 \\ &\leq bf_{n-1} - c + bf_{n-2} - c + 1 = bf_n - 2c + 1. \end{aligned}$$

To conclude that $g(n) \leq bf_n - c$, it suffices that $-2c + 1 \leq -c$, or equivalently that $c \geq 1$. We have thus established that the truth of our conjecture on any given integer n follows from its assumed truth on the two previous integers provided $c \geq 1$, regardless of the value of b. Before we can claim the desired theorem, we still need to determine values of b and c that make it work on two consecutive integers. For instance, $b = 2$ and $c = 1$ make it work on $n = 1$ and $n = 2$, and therefore $g(n) \leq 2f_n - 1$ for all $n \geq 1$.

The key idea of strengthening the incompletely specified statement to be proved when constructive induction fails may again appear to be produced like a rabbit out of a hat. Nevertheless, this idea comes very naturally with experience. To gain such experience, work Problems 1.31 and 1.33. Unlike the Fibonacci examples, which could have been handled easily by the techniques of Section 4.7, the cases tackled in these problems are best handled by constructive induction.