Solving Recurrences

Eric Ruppert

November 28, 2007

1 Introduction

An (infinite) sequence is a function from the set $\mathbb{N} = \{0, 1, 2, \ldots\}$ of natural numbers to some set S. If $a : \mathbb{N} \to S$ is a sequence, we often denote $a(n)$ by a_n. The values $a_0, a_1, a_2, \ldots$ are called the elements or terms of the sequence.

A recurrence relation is a way of defining a sequence. A few of the first elements of the sequence are given explicitly. Then, the recurrence relation gives relationships between elements of the sequence that are sufficient to uniquely determine all the remaining elements' values.

Example 1 The famous Fibonacci sequence can be defined by the recurrence

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2}, \text{ for } n \geq 2. \end{aligned}$$

The first few terms of the sequence are $a_0 = 0, a_1 = 1, a_2 = 1, a_3 = 2, a_4 = 3$ and $a_5 = 4$. ■

Example 2 We can define a sequence T_n using the recurrence

$$\begin{aligned} T(1) &= 1 \\ T(n) &= T(\lceil n/2 \rceil) + 4, \text{ for } n \geq 2. \end{aligned}$$

(Recall that $\lceil x \rceil$ is x rounded up to the nearest integer, *i.e.*, the smallest integer that is greater than or equal to x. Similarly, $\lfloor x \rfloor$ is x rounded down to the nearest integer, *i.e.*, the largest integer that is smaller than or equal to x.) The first few terms in the sequence are $T(0) = 1, T(1) = 1, T(2) = 5, T(3) = 9, T(4) = 9, T(5) = 13, T(6) = 13$. We shall come back to this recurrence later. ■

The explicit description of the first few terms of the sequence (F_0 and F_1 in Example 1, and T_0 in Example 2) are called the *initial conditions* or *base cases* of the recurrence. When defining a sequence using a recurrence, you must be careful to include enough information in the initial conditions to properly define the entire sequence. For example, the following recurrence does *not* define a sequence properly:

$$\begin{aligned} T(0) &= 1 \\ T(n) &= T(\lceil n/2 \rceil), \text{ for } n \geq 1 \end{aligned}$$

When $n = 1$, the equation would be $T(1) = T(\lceil 1/2 \rceil) + 1 = T(1) + 1$, which is impossible. This could be fixed by adding another base case:

$$\begin{aligned} T(0) &= 1 \\ T(1) &= 2 \\ T(n) &= T(\lceil n/2 \rceil) + 1, \text{ for } n \geq 2. \end{aligned}$$

This works because $\lceil \frac{n}{2} \rceil \leq \frac{n}{2} + \frac{1}{2} < n$ whenever $n \geq 2$.

When defining a sequence $R(n)$ using a recurrence, one way to make sure that it is well-defined is to define $R(0), R(1), R(2), \ldots, R(n_0)$ as base cases and then, for $n > n_0$, use a recurrence relation that defines $R(n)$ in terms of $R(0), R(1), \ldots, R(n-1)$. (It is then easy to prove by strong induction the claim that $R(n)$ is well-defined for all natural numbers n.)

The elements of the sequence might not be numbers. For example, you can define a sequence of sets or a sequence of ordered pairs by a recurrence relation. But in these notes, we focus on the case where $S = \mathbb{R}$, so each term in the sequence is a number.

A recurrence is often the most natural way to define a sequence that we are interested in. However, it can be difficult to work with a sequence defined by a recurrence, or prove that such a sequence has certain properties. For example, if we want to compute the 1000th term of the Fibonacci sequence, F_{1000}, using the recurrence, we must first compute the 1000 terms $F_0, F_1, F_2, \ldots, F_{999}$. So, often, we would like to construct an explicit formula for the nth term in the sequence. For example, we shall prove, below, that $F_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$. The procedure of coming up with an explicit formula for elements of the sequence is called *solving the recurrence*.

The goal of these notes is to look at ways of solving recurrences using an elementary approach. Elementary does not mean "easy"; it just means that you do not need a lot of background knowledge to read them. In particular, the notes will not use calculus. However, you should be comfortable with high-school algebra (including solving linear equations), function notation and basic proof techniques (like mathematical induction).

2 Modelling Using Recurrences

[[[This section will be filled in later. For now, just read Section 7.1 of Rosen's text.]]] ***

2.1 Derangements

Suppose there are n people, and each one owns one hat. All n hats are thrown in a pile, and each person pulls one hat out of the pile to put it on. We would like to count the number of ways that this could be done such that *nobody* puts on his own hat. (This is called a *derangement* of the hats.) Let D_n denote the number of derangements when there are n people.

First, let's get some intuition: For $n = 1$, there is no way for one person to not get his own hat, so $D_1 = 0$. For $n = 2$, the only derangement is to have the two people swap hats, so $D_2 = 1$. For $n = 3$ there are two derangements: the three people can stand in a triangle and everyone passes his hat to the person on his right or everyone passes his hat to the person on his left, so $D_3 = 2$. For $n = 4$, there are 9 derangements described by the rows of the following table.

person	1	2	3	4
	2	1	4	3
	2	3	4	1
	2	4	1	3
	3	1	4	2
hats	3	4	1	2
	3	4	2	1
	4	1	2	3
	4	3	1	2
	4	3	2	1

We now derive a recurrence for D_n. Suppose person 1 gets the hat belonging to person k (where $2 \leq k \leq n$). To count the number of derangements for n people we consider two cases.
Case 1: person k gets the hat belonging to person 1. In this case, persons 1 and k have simply swapped hats. The remaining $n - 2$ people must mix up their hats so that nobody has his own hat. This can be done in D_{n-2} ways.
Case 2: person k does not get the hat belonging to person 1. Each possible assignment of hats to people in this case can be obtained as follows. First, persons 1 and k swap hats. Then, everybody except person 1 mix up their hats so that nobody ends up with the hat they began with. (Notice that since person k started with the hat that belongs to person 1, he will end up with some hat other than person 1's hat, which is one of the derangements we are trying to count in Case 2.) There are D_{n-1} ways to do this.

Thus, $D_n = (n - 1)(D_{n-2} + D_{n-1})$ for $n \geq 3$. (The $(n - 1)$ factor comes from the choice of k and the D_{n-2} comes from Case 1, while the D_{n-1} comes from Case 2.

3 Guessing a Solution

In the past you may have learned techniques for solving certain kinds of equations (e.g., linear equations). Perhaps you were shown an algorithm (a sequence of steps) that you could follow, and you would be guaranteed to get the answer by following those steps. There is no algorithm that you can follow that will allow you to solve all recurrences. In fact, there are some kinds of recurrences that cannot be solved: there is no simple, explicit formula for the elements of the sequence. However, there are algorithms for solving certain kinds of recurrence relations, and we shall see some of those. But, in general, solving a recurrence often takes some creativity. Experience helps: if you are solving a recurrence and you can remember having seen a similar one before, then you might be able to use the same technique. It also helps if you really understand the phenomenon that is being modelled by the recurrence: you can use information about it to dismiss implausible answers and focus on the more plausible ones.

The most basic technique for solving a recurrence is *guessing* the answer and then checking that it is correct. If you were just making random guesses, it would probably take you a very long time before finding the correct answer. So you should make educated guesses. In this section, we shall look at a few different ways of coming up with educated guesses.

3.1 Verifying a Guess

Since this is mathematics, just guessing the solution is not the end of the story because your guess could be wrong. After you have guessed the solution to the recurrence, you *must* prove that your guess is, in fact, the correct solution. Fortunately, this is usually pretty easy if you guessed right: it is usually a straightforward argument using mathematical induction. This is probably best illustrated using some examples.

Example 3 Consider the recurrence

$$\begin{aligned} a_0 &= 5 \\ a_n &= 2a_{n-1} + 1 \end{aligned}$$

After staring at it for a while, suppose you come up with an inspired guess: $a_n = 2^{n+2} + 2^{n+1} - 1$. (The rest of this section will describe some ways for coming up with such a guess but, for now, we just want to see how to verify the guess once it is made.) Thus, the claim we want to prove is:

Claim: For all $n \geq 0$, $a_n = 2^{n+2} + 2^{n+1} - 1$.

Proof (by mathematical induction on n):

Base case ($n = 0$): $a_0 = 5 = 4 + 2 - 1 = 2^2 + 2^1 - 1$.

Induction step: Let $n \geq 1$. Assume $a_{n-1} = 2^{n-1+2} + 2^{n-1+1} - 1$. Our goal is to prove $a_n = 2^{n+2} + 2^{n+1} - 1$. We have

$$\begin{aligned} a_n &= 2a_{n-1} + 1 && \text{by the definition of } a_n\text{, since } n \geq 1 \\ &= 2(2^{n-1+2} + 2^{n-1+1} - 1) + 1 && \text{by the induction hypothesis} \\ &= 2^{n+2} + 2^{n+1} - 2 + 1 \\ &= 2^{n+2} + 2^{n+1} - 1. \end{aligned}$$

This completes the proof by induction. ■

We used regular induction in Example 3 because the recurrence defined a_n in terms of a_{n-1}. If, instead each term of the recurrence is defined using several smaller terms, strong induction would work better. We also have to adjust the number of base cases, depending on what values of n the recurrence relation applies to. (Thus, the base cases of the induction step usually mirror the base cases of the recurrence relation.) These ideas are illustrated in the next example.

Example 4 Consider the sequence defined by

$$\begin{aligned} b(0) &= 0 \\ b(1) &= 1 \\ b(n) &= b(\left\lfloor \frac{n}{2} \right\rfloor) + b(\left\lceil \frac{n}{2} \right\rceil), \text{ for } n \geq 2. \end{aligned}$$

If you look at the first five or six terms of this sequence, it is not hard to come up with a very simple guess: $b(n) = n$. We can prove it by strong induction.

Claim: For all $n \geq 0$, $b(n) = n$.

Proof by strong induction on n:

Base cases ($n = 0, 1$): $b(0) = 0$ and $b(1) = 1$ by definition.

Induction step: Let $n \geq 2$. Assume that $b(k) = k$ for $0 \leq k < n$. Our goal is to prove that $b(n) = n$. We have

$$\begin{aligned} b_(n) &= b(\left\lfloor \frac{n}{2} \right\rfloor) + b(\left\lceil \frac{n}{2} \right\rceil) && \text{since } n \geq 2 \\ &= \left\lfloor \frac{n}{2} \right\rfloor + \left\lceil \frac{n}{2} \right\rceil && \text{by induction hypothesis, since } 0 \leq \left\lfloor \frac{n}{2} \right\rfloor \leq \left\lceil \frac{n}{2} \right\rceil \leq \frac{n}{2} + \frac{1}{2} < n \\ &= n \end{aligned}$$

(The last equality can be proved using two cases, depending on whether n is even or odd. If $n = 2m$ for some natural number m, then $\lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil = \lfloor \frac{2m}{2} \rfloor + \lceil \frac{2m}{2} \rceil = m + m = n$. Otherwise, $n = 2m + 1$ for some natural number m and $\lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil = \lfloor \frac{2m+1}{2} \rfloor + \lceil \frac{2m+1}{2} \rceil = \lfloor m + \frac{1}{2} \rfloor + \lceil m + \frac{1}{2} \rceil = m + m + 1 = n$.)

This completes the proof of the claim. ■

Notice that mathematical induction is a very natural proof technique to use together with recurrences. The recurrence relates one sequence element to previous ones. The induction hypothesis of the mathematical induction proof allows you to assume that the previous terms are known. Then, the recurrence relation allows you to determine the value of the next term in the sequence to complete the induction step.

As long as you provide a formal proof that your guess is correct, it does not matter how you come up with your guess: it could be a wild guess or you could get it by asking Numeros, the mighty god of arithmetic, in a prayer. But because you probably do not want to rely on Numeros, who is a temperamental god at the best of times, we shall look at some techniques for coming up with a guess in the rest of this section.

3.2 Looking for Patterns

Some patterns are easy to guess just by looking at the first few terms of the sequence (as in Example 4). Others might be harder to see. One thing that is useful in identifying patterns is knowing what to look for to recognize certain kinds of patterns. Here we shall learn how to recognize some of the most common sequences.

3.2.1 Identifying Arithmetic Sequences

One of the most basic types of sequences are arithmetic sequences, where the terms are $a_n = c_1 n + c_0$ for some constants c_1 and c_0. If the sequence has this form, the difference between successive terms will always be c_1, and this is something that is easy to look for.

Example 5 Consider the recurrence

$$\begin{aligned} a_0 &= 12 \\ a_1 &= 17 \\ a_n &= a_{\lfloor n/2 \rfloor} + a_{\lceil n/2 \rceil} - 12, \text{ for } n \geq 2. \end{aligned}$$

The first few elements of the sequence are $12, 17, 22, 27, 32$. We can take differences of successive elements of this sequence:

n	a_n	$a_n - a_{n-1}$
0	12	-
1	17	5
2	22	5
3	27	5
4	32	5

From this table, we could guess that the sequence is an arithmetic sequence with $c_1 = 5$. Then solving $a_0 = c_1 \cdot 0 + c_0$ for c_0 yields $c_0 = 12$. So our guess for the solution would be $a_n = 5n + 12$. (Exercise: verify this guess satisfies the recurrence using mathematical induction.) ■

This same technique will work whenever the sequence is an arithmetic sequence.

3.2.2 Identifying Polynomial Sequences

We saw how to identify arithmetic sequences, which grow linearly as a function of n. What if the solution is a quadratic formula of the form $a_n = c_2 n^2 + c_1 n + c_0$, where c_0, c_1 and c_2 are constants? How can we identify this pattern by looking at the first few terms?

Let's look at the difference between successive terms again: For $n \geq 1$, let $b_n = a_n - a_{n-1} = c_2 n^2 + c_1 n + c_0 - (c_2(n-1)^2 + c_1(n-1) + c_0) = 2c_2 n - c_2 + c_1 = c'_1 n + c'_0$, where $c'_1 = 2c_2$ and $c'_0 = c_1 - c_2$. Thus, the sequence of differences, themselves, form an arithmetic sequence, which we already know how to identify.

Example 6 Consider the recurrence

$$\begin{aligned} a_0 &= 7 \\ a_1 &= 12 \\ a_n &= a_{n-2} + 8n - 2, \text{ for } n \geq 2. \end{aligned}$$

Let's look at the differences of successive elements, and then check if those differences form an arithmetic sequence by looking at *their* differences.

n	a_n	$b_n = a_n - a_{n-1}$	$b_n - b_{n-1}$
0	7	-	-
1	12	5	-
2	21	9	4
3	34	13	4
4	51	17	4
5	72	21	4
6	97	25	4

The entries in the third column of this table are sometimes called first-order differences, and the entries in the fourth column are called second-order differences (because they are differences of differences of successive terms). Because the second-order differences appear to be constant, a reasonable guess would be that the sequence itself is of the form $a_n = c_2 n^2 + c_1 n + c_0$. By plugging in the values of a_0, a_1 and a_2 and solving for c_0, c_1 and c_2, we could arrive at the guess $a_n = 2n^2 + 3n + 7$. (Exercise: verify this guess is correct.) ■

More generally, if the solution to the recurrence is a polynomial equation of degree d, *i.e.*, $a_n = \sum_{i=0}^{d} c_i n^i = c_d n^d + c_{d-1} n^{d-1} + c_{d-2} n^{d-2} + \cdots + c_1 n + c_0$ (where the c_i's are all constants), then the dth-order differences will be constant. This is because $b_n = a_n - a_{n-1}$ will be a polynomial of degree $d - 1$, so the $(d-1)$th order differences of the b_n's will be constant. (Check this by writing out $a_n - a_{n-1}$ and noticing that the leading degree-d terms of a_n and a_{n-1} cancel out, leaving only terms of degree at most $d - 1$.)

Exercise: Make the preceding discussion formal by proving the following claim by induction on d: if $a_n = p(n)$ where p is any polynomial of degree d, then the dth-order differences of a_n are constant.

Note: I am not being very formal in describing how to come up with guesses. This is because you can be totally informal in the way you arrive at your guess anyway: you are probably just going to be looking at the first few elements of the sequence, so there is no way to know whether the pattern you see carries on forever. Or you can even just depend on Numeros. But a reminder: if you use the method of guessing a solution you *must* prove your guess is correct in a formal way.

Let's look at another example where the solution is a polynomial.

Example 7 Consider the recurrence

$$\begin{aligned} s_0 &= 0 \\ s_n &= s_{n-1} + n^2, \text{ for } n \geq 1. \end{aligned}$$

In the following table, we look at the first few terms of the sequence, then take the differences of successive terms. Since those are not constant, we shall try taking 2nd-order differences. Those are still not constant, so we try taking 3rd-order differences. Aha! Those seem to be constant, at least for small n.

n	s_n	1st-order differences	2nd-order differences	3rd-order differences
0	0	-	-	-
1	1	1	-	-
2	5	4	3	-
3	14	9	5	2
4	30	16	7	2
5	55	25	9	2
6	91	36	11	2
7	140	49	13	2

Based on the above information, we shall guess that the solution is a degree-3 polynomial: $a_n = c_3n^3 + c_2n^2 + c_1n + c_0$. By plugging the first 4 values of the sequence into this formula, we can solve for c_0, c_1, c_2 and c_3 to get $a_n = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$. (Exercise: prove this guess is correct.)

Remark: You may have noticed that the recurrence is identical to the one that defines $\sum_{i=0}^{n} i^2 = \left(\sum_{i=0}^{n-1} i^2\right) + n^2$, with s_n playing the role of $\sum_{i=0}^{n} i^2$. We saw some techniques earlier in the course for computing sums; recall that we showed $s_n = \frac{n(n+1)(2n+1)}{6}$, which agrees with the guess we found here. ■

3.2.3 Identifying Geometric Sequences

A geometric sequence has the form $g_n = cr^n$ where c and r are constants. The way to identify these sequences is to look at the *ratio* between successive terms in the sequence. That ratio will always be r.

Example 8 Consider the recurrence

$$\begin{aligned} g_0 &= 2 \\ g_1 &= 6 \\ g_n &= g_{n-1} + 6g_{n-2}, \text{ for } n \geq 2. \end{aligned}$$

Let's look at the ratios between successive terms.

n	g_n	g_n/g_{n-1}
0	2	-
1	6	3
2	18	3
3	54	3
4	162	3

A reasonable guess at this point would be that $g_n = 2 \cdot 3^n$. ∎

3.2.4 Identifying Combinations of Sequences

Guessing becomes a little more complicated when the formula is a combination of several of the preceding types.

Many sequences are not exactly geometric, but "nearly" geometric. In Example 3, the first few terms of the sequence were 5,11,23,47,95,191. It is clear from this pattern (and from the recurrence relation itself) that each term is approximately double the preceding term, but not exactly. So we might guess that the solution will involve powers of 2. At the beginning of the sequence, the terms are close to 2^{n+2} but they get further and further away from 2^{n+2} as we go further along the sequence. But we might want to take 2^{n+2} as a first approximation of the answer. Then we can check how close this is to the correct answer:

n	a_n	2^{n+2}	$a_n - 2^{n+2}$
0	5	4	1
1	11	8	3
2	23	16	7
3	47	32	15
4	95	64	31
5	190	128	63

From the table, we see that the "error term" for our first approximation, $a_n - 2^{n+2}$, roughly doubles at each step, so it again looks like it should be close to powers of 2. In fact, it is not hard to guess that this error term is $2^{n+1} - 1$. Thus, we arrive at the guess $a_n = 2^{n+2} + 2^{n+1} - 1$, which we already verified was correct in Example 3.

Exercise 9 Guess a solution for the following recurrence and prove it correct

$$\begin{aligned} a_0 &= 1 \\ a_n &= 2(a_{n-1} - n + 2), \text{ for } n \geq 1. \end{aligned}$$

Hint: start with powers of 2 and look at the "error terms".

Exercise 10 Guess a solution for the following recurrence and prove it correct.

$$\begin{aligned} b_0 &= 2 \\ b_1 &= 8 \\ b_n &= b_{n-1} + 6b_{n-2} - 4n + 11, \text{ for } n \geq 2. \end{aligned}$$

3.3 Repeated Substitution

There is another method for coming up with a guess for the solution of a recurrence relation. Instead of starting at the base case and working forward, we can start with the term a_n and repeatedly apply the recurrence to express a_n in terms of earlier and earlier terms in the sequence. If we imagine carrying this procedure through until we have expressed a_n in terms of the base cases, then we will be able to read off the formula for a_n. We shall call this method *repeated substitution*. Let's see some examples.

Example 11 Consider the recurrence

$$\begin{aligned} a_0 &= 0 \\ a_n &= 3a_{n-1} + 2n \text{ for } n \geq 1. \end{aligned}$$

By applying the recurrence relation repeatedly, we get

$$\begin{aligned} a_n &= 3a_{n-1} + 2n \\ &= 3(3a_{n-2} + 2(n-1)) + 2n \\ &= 3^2 a_{n-2} + 2[n + 3(n-1)] \\ &= 3^2(3a_{n-3} + 2(n-2)) + 2[n + 3(n-1)] \\ &= 3^3 a_{n-3} + 2[n + 3(n-1) + 3^2(n-2)] \\ &= 3^3(3a_{n-4} + 2(n-3)) + 2[n + 3(n-1) + 3^2(n-2)] \\ &= 3^4 a_{n-4} + 2[n + 3(n-1) + 3^2(n-2) + 3^3(n-3)] \\ &\vdots \end{aligned}$$

We see that there is a pattern emerging: every time we do a substitution, the coefficient of a_i increases by a factor of 3 and the sum in square brackets is extended by one term. We conjecture that this pattern continues, so that after k substitutions we would have

$$a_n = 3^k a_{n-k} + 2\sum_{i=0}^{k-1} 3^i(n-i).$$

So, after n substitutions, we would have

$$a_n = 3^n a_0 + 2\sum_{i=0}^{n-1} 3^i(n-i) = 2\sum_{i=0}^{n-1} 3^i(n-i),$$

by using the fact that $a_0 = 0$. Here is where we get to use our abilities with figuring out sums from earlier in the course:

$$a_n = 2\sum_{i=0}^{n-1} 3^i(n-i)$$

$$
\begin{aligned}
&= \left(2n\sum_{i=0}^{n-1}3^i\right) - \left(2\sum_{i=0}^{n-1}3^i i\right) \\
&= 2n\cdot\frac{3^n-1}{3-1} - 2\cdot\frac{3^n(2n-3)+3}{4} \qquad \text{(by formulas from our work on sums)} \\
&= \frac{2n\cdot 3^n + 3^{n+1} - 4n - 3}{4}.
\end{aligned}
$$

Exercise: prove this guess is correct. ■

Example 12 Consider the recurrence

$$
\begin{aligned}
a_0 &= 12 \\
a_1 &= 20 \\
a_n &= 2a_{n-1} - a_{n-2} \text{ (for } n \geq 2).
\end{aligned}
$$

By applying the recurrence relation repeatedly, we have

$$
\begin{aligned}
a_n &= 2a_{n-1} - a_{n-2} = 2(2a_{n-2} - a_{n-3}) - a_{n-2} \\
&= 3a_{n-2} - 2a_{n-3} = 3(2a_{n-3} - a_{n-4}) - 2a_{n-3} \\
&= 4a_{n-3} - 3a_{n-4} = 4(2a_{n-4} - a_{n-5}) - 3a_{n-4} \\
&= 5a_{n-4} - 4a_{n-5} \\
&\vdots
\end{aligned}
$$

By this point, we can identify a pattern and conjecture that, after k substitutions, we shall have $a_n = (k+1)a_{n-k} - ka_{n-k-1}$. By taking $k = n-1$, we obtain the guess $a_n = (n-1+1)a_{n-(n-1)} - (n-1)a_{n-(n-1)-1} = na_1 - (n-1)a_0$. Then we can plug in the initial values, to obtain a guess $a_n = 20n - 12(n-1) = 8n+12$ (which still has to be proved correct, as usual). ■

Example 13 Consider the recurrence

$$
\begin{aligned}
a_0 &= 2 \\
a_n &= 2\sqrt{a_{n-1}}, \text{ for } n \geq 1.
\end{aligned}
$$

Let's start doing repeated substitutions until we find a pattern.

$$
\begin{aligned}
a_n &= 2\sqrt{a_{n-1}} && = 2^1 a_{n-1}^{1/2} \\
&= 2(2\sqrt{a_{n-2}})^{1/2} && = 2^{3/2} a_{n-2}^{1/4} \\
&= 2^{3/2}(2\sqrt{a_{n-3}})^{1/4} && = 2^{7/4} a_{n-3}^{1/8} \\
&= 2^{7/4}(2\sqrt{a_{n-4}})^{1/8} && = 2^{15/8} a_{n-4}^{1/16} \\
&= 2^{15/8}(2\sqrt{a_{n-5}})^{1/16} && = 2^{31/16} a_{n-4}^{1/32} \\
&\vdots
\end{aligned}
$$

Focus on the last expression in each line. The exponents on the a_i term form a nice geometric sequence $1/2, 1/4, 1/8, 1/16, \ldots$ and the exponents on the

2 also form a nice sequence: $1/1, 3/2, 7/4, 15/8, 31/16, \ldots$ (the denominators are powers of 2 and the numerators are each one less than a power of 2). By extending this pattern we can guess that after k substitutions, we would have

$$a_n = 2^{(2^k-1)/2^{k-1}} a_{n-k}^{1/2^k}.$$

So, when $k = n$, we should get

$$a_n = 2^{(2^n-1)/2^{n-1}} a_0^{1/2^n} = 2^{(2^n-1)/2^{n-1}} 2^{1/2^n} = 2^{2-1/2^n}.$$

(This guess still has to be proved correct, as usual.) ■

Often, the key to spotting a pattern when doing repeated substitution is to simplify after each substitution is performed, but sometimes you should not simplify too much because that can conceal the structure of the pattern. For instance, in Example 11, we did not combine all the terms in the square brackets while doing the substitutions; this way, the structure of each line (as a sum) was revealed.

4 Linear Recurrences

Guessing solutions to a recurrence is a useful technique because it is very widely applicable: you can use it to solve any recurrence, as long as you are able to spot the patterns in the sequence. But using that technique requires real work for each and every recurrence you solve. Each one could have a new pattern that you have not seen before and you have to be able to identify it. Some kinds of recurrences come up again and again (you could say that they recur) in all kinds of different applications. Instead of re-doing the work of discovering the pattern in each one, we can develop some specialized techniques to deal with those common kinds of recurrences.

In this section we shall consider sequences where one term is a linear function of earlier terms in the sequence.

4.1 Linear Homogenous Recurrences

Let k be a positive integer. A *linear homogeneous recurrence relation of degree k with constant coefficients* has the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + c_3 a_{n-3} + \cdots + c_k a_{n-k}, \tag{1}$$

where the c_i's are constants. In order to completely define a sequence, this type of recurrence must also come with k base cases that specify the values of $a_0, a_1, \ldots, a_{k-1}$. In this section, we shall describe how to solve any recurrence of this form. We have already seen some examples of this kind of recurrence (see Examples 1, 8 and 12).

We are going to focus on just finding formulas that satisfy Equation (1), without worrying (for now) about also satisfying the base cases. (Notice that there can be many sequences that satisfy Equation (1); in fact there will be a different sequence for each possible set of base cases.) The following proposition is one of the key properties that make linear homogeneous recurrences fairly easy to solve.

Proposition 14 *If the sequence a_n satisfies Equation (1) and a'_n is another sequence that satisfies Equation (1), then $b_n = a_n + a'_n$ and $d_n = \alpha a_n$ are also sequences that satisfy Equation (1) (where α is any constant).*

Proof: Since a_n and a'_n satisfy Equation (1), we have

$$\begin{aligned} b_n &= a_n + a'_n \\ &= [c_1 a_{n-1} + c_2 a_{n-2} + \cdots c_k a_{n-k}] + [c_1 a'_{n-1} + c_2 a'_{n-2} + \cdots + c_k a'_{n-k}] \\ &= c_1(a_{n-1} + a'_{n-1}) + c_2(a_{n-2} + a'_{n-2}) + \cdots + c_k(a_{n-k} + a'_{n-k}) \\ &= c_1 b_{n-1} + c_2 b_{n-2} + \cdots + c_k b_{n-k}. \end{aligned}$$

Similarly, we have

$$\begin{aligned} d_n &= \alpha a_n \\ &= \alpha[c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}] \\ &= c_1(\alpha a_{n-1}) + c_2(\alpha a_{n-2}) + \cdots + c_k(\alpha a_{n-k}) \\ &= c_1 d_{n-1} + c_2 d_{n-2} + \cdots + c_k d_{n-k}. \end{aligned}$$

■

It follows from this proposition that if we find some basic solutions to Equation (1), then any linear combination of them will also be a solution to Equation (1). This is useful because it is (relatively) easy to find some solutions to Equation (1), and then we can combine the solutions we find by adding them together or multiplying them by constants in order to make them "fit" with the particular base cases that we are interested in. So now we focus on the task of finding the basic solutions to Equation (1).

At some point, people observed that geometric sequences come up *a lot* when solving linear homogeneous recurrences. So let's try to see if any sequences of the form $a_n = r^n$ satisfy Equation (1). In other, words, we would like to know if there is any r that satisfies

$$r^n = c_1 r^{n-1} + c_2 r^{n-2} + \cdots + c_k r^{n-k}.$$

By moving all terms on to the same side, we get

$$r^n - c_1 r^{n-1} - c_2 r^{n-2} - \cdots - c_k r^{n-k} = 0. \tag{2}$$

Dividing both sides by r^{n-k} yields

$$r^k - c_1 r^{k-1} - c_2 r^{k-1} - \cdots - c_k r^0 = 0. \tag{3}$$

Equation (3) is called the *characteristic equation* of the recurrence (1). For example, the characteristic equation of the Fibonacci recurrence $F_n = F_{n-1} + F_{n-2}$ is $r^2 - r - 1 = 0$. The next proposition follows immediately from the preceding discussion.

Proposition 15 *If r satisfies Equation (3), then $a_n = r^n$ satisfies Equation (1).*

Combining Propositions 15 and 14, we obtain the following Theorem.

Theorem 16 *If $r_1, r_2, \ldots, r_m$ all satisfy Equation (3), then for any constants $\alpha_1, \alpha_2, \ldots, \alpha_m$, the sequence $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \cdots + \alpha_m r^m$ satisfies Equation (1).*

Proof: By Proposition 15, we know that r_i^n satisfies Equation (1) (for each i). Thus, by Proposition 14, we know that $\alpha_i r_i^n$ satisfies Equation (1) (for each i). Applying Proposition 14 again (in fact, with an induction argument), we see that $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n + \cdots + \alpha_m r^m$ satisfies Equation (1). ■

Theorem 16 allows us to solve a lot of linear homogeneous recurrences very easily. The procedure is to find the solutions of the corresponding characteristic equation and then choose the constants α_i to satisfy the base cases of the recurrence.

Example 17 Let's use this technique to solve the Fibonacci recurrence from Example 1:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2}, \text{ for } n \geq 2. \end{aligned}$$

The characteristic equation of this recurrence is $r^2 - r - 1 = 0$. We can solve this equation for r using the quadratic formula: $r = \frac{1 \pm \sqrt{5}}{2}$. By Theorem 16, $F_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$ satisfies $F_n = F_{n-1} + F_{n-2}$, for any values of α_1 and α_2. Now we just have to choose values for α_1 and α_2 so that the equations $F_0 = 0$ and $F_1 = 1$ are also satisfied. Thus, we must have

$$\begin{aligned} \alpha_1 + \alpha_2 &= 0 \\ \left(\frac{1+\sqrt{5}}{2}\right)\alpha_1 + \left(\frac{1-\sqrt{5}}{2}\right)\alpha_2 &= 1 \end{aligned}$$

These are just two linear equations in two unknowns that we can solve for α_1 and α_2 to get $\alpha_1 = \frac{1}{\sqrt{5}}$ and $\alpha_2 = -\frac{1}{\sqrt{5}}$. Thus, $F_n = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2}\right)^n$.

Important remark: Now, this is not merely a guess at a correct solution. We have Theorem 16, which tells us that this formula for F_n satisfies the recurrence equation, and we just chose α_1 and α_2 to satisfy the base cases. So this formula is guaranteed to be correct; we do not have to prove it correct by induction. ■

In the example above, it was pretty easy to find the solutions to the characteristic equation. If k is bigger than 2, you may have to use your prowess at factoring polynomials to find the solutions of the characteristic equation. We now consider an example with $k = 3$.

Example 18 Consider the recurrence

$$
\begin{aligned}
a_0 &= 8 \\
a_1 &= 6 \\
a_2 &= 26 \\
a_n &= -a_{n-1} + 4a_{n-2} + 4a_{n-3}
\end{aligned}
$$

The characteristic equation is $r^3 + r^2 - 4r - 4 = 0$. Factoring this gives us $(r+1)(r+2)(r-2) = 0$, so the solutions are $r_1 = -1$, $r_2 = -2$ and $r_3 = 2$. Thus, for any constants α_1, α_2 and α_3, the formula $a_n = \alpha_1(-1)^n + \alpha_2(-2)^n + \alpha_3 2^n$ will satisfy the recurrence relation. Now we just have to choose α_1, α_2 and α_3 to satisfy the base cases, so we need

$$
\begin{aligned}
\alpha_1 + \alpha_2 + \alpha_3 &= 8 \\
-\alpha_1 - 2\alpha_2 + 2\alpha_3 &= 6 \\
\alpha_1 + 4\alpha_2 + 4\alpha_3 &= 26
\end{aligned}
$$

Solving these equations gives us $\alpha_1 = 2, \alpha_2 = 1$ and $\alpha_3 = 5$. So the solution to our recurrence is $a_n = 2(-1)^n + (-2)^n + 5 \cdot 2^n$. (Again, we do not need to prove this because it is established by Theorem 16.) ■

4.1.1 Multiple Roots of Characteristic Equation

The characteristic equation (3) may have up to k different solutions. If it has k different solutions, this technique will always work: it will be possible to find a solution of the type described in Theorem 16 that satisfies the k base cases of the recurrence as well as the recurrence equation. (We shall not prove this, since it requires some linear algebra, and we do not really need to know this fact in order to use the method, but it is nice to know that the method will work.) However, if the characteristic equation has fewer than k distinct solutions, the method presented above might not work, so we need to generalize it a little.

If the characteristic equation has k distinct solutions $r_1, r_2, \ldots, r_k$ it can be rewritten as

$$(r - r_1)(r - r_2) \cdots (r - r_k) = 0.$$

But what if, after factoring, the characteristic equation has $m + 1$ factors of $(r - r_1)$, for example. (When this happens, r_1 is called a solution of the characteristic equation *with multiplicity* $m + 1$.) When this happens, not only is r_1^n a solution to Equation (1), but so is $nr_1^n, n^2r_1^n, \ldots, n^m r_1^n$, as we see in the following proposition.

Proposition 19 *If r_0 is a solution of the characteristic equation with multiplicity at least $m + 1$, then $n^m r_0^n$ is a solution to Equation (1).*

The proof of this proposition is a little involved so it is presented in Section 4.1.2.

Using this proposition, together with Lemma 14, we get a technique for solving linear homogeneous recurrences even when the characteristic equation does not have distinct solutions. Again, we find the solutions of the characteristic equation. For solutions that have multiplicity 1, we get a simple geometric sequence that satisfies Equation (1). For solutions that have multiplicity greater than 1, we obtain sequences of the form described in Proposition 19 that satisfy Equation (1). Then, by Proposition 14, we know that any combination of multiples of those basic sequences we have found are also solutions to Equation (1). Then we can just find one of those combinations that satisfies the base cases. The overall procedure is very similar to the procedure used in Examples 17 and 18, so we just give one more example to illustrate.

Example 20 Consider the recurrence

$$\begin{aligned} a_0 &= 1 \\ a_1 &= 4 \\ a_2 &= 28 \\ a_3 &= 32 \\ a_n &= 8a_{n-2} - 16a_{n-4}, \text{ for } n \geq 4. \end{aligned}$$

The characteristic equation is $r^4 - 8r^2 + 16 = 0$. The left hand side can be factored: $r^4 - 8r^2 + 16 = (r^2 - 4)^2 = (r-2)^2(r+2)^2$. Thus, there are two solutions of the characteristic equation, $r = 2$ and $r = -2$, each with multiplicity 2. By Proposition 19, we know that $2^n, n2^n, (-2)^n$ and $n(-2)^n$ each satisfy the recurrence relation. Thus, by Proposition 14, $a_n = \alpha_1 2^n + \alpha_2 n 2^n + \alpha_3(-2)^n + \alpha_4 n(-2)^n$ also satisfies the recurrence equation. Now we just have to choose the α_i's to satisfy the base cases as well. Plugging the formula into the base cases yields 4 equations:

$$\begin{array}{ccccccccl} \alpha_1 & & & + & \alpha_3 & & & = 1 \\ 2\alpha_1 & + & 2\alpha_2 & - & 2\alpha_3 & - & 2\alpha_4 & = 4 \\ 4\alpha_1 & + & 8\alpha_2 & + & 4\alpha_3 & + & 8\alpha_4 & = 28 \\ 8\alpha_1 & + & 24\alpha_2 & - & 8\alpha_3 & - & 24\alpha_4 & = 32 \end{array}$$

Solving these yields $\alpha_1 = 1, \alpha_2 = 2, \alpha_3 = 0$ and $\alpha_4 = 1$. Thus, the explicit formula for the sequence is $a_n = 2^n + 2n2^n + n(-2)^n$. ∎

Finally, we remark that the characteristic equation will sometimes have solutions that are complex numbers. But all of the proofs in this section apply even if the solutions are complex numbers, so all of the techniques are still applicable in this case. (You just have to be careful to do your arithmetic with complex numbers instead of real numbers.)

4.1.2 Proof of Proposition 19

We now go back and fill in the (rather technical) proof of Proposition 19. Given the recurrence equation Equation (1), we can define $p(r)$ to be the left hand side of Equation (2):

$$p(r) = r^n - c_1 r^{n-1} - c_2 r^{n-2} - \cdots - c_k r^{n-k}.$$

Suppose r_0 is a solution of the equation $p(r) = 0$ with multiplicity at least $m+1$. Recall that this means $p(r) = (r - r_0)^{m+1} q(r)$ for some polynomial $q(r)$. Our goal is to show that $a_n = n^m r_0^n$ satisfies Equation (1). In other words, we must show that

$$n^m r_0^n = c_1(n-1)^m r_0^{n-1} + c_2(n-2)^m r_0^{n-2} + c_3(n-3)^m r_0^{n-3} + \cdots + c_k(n-k)^m r_0^{n-k}.$$

If we move all the terms to one side of this equation we see that our goal is to show

$$n^m r_0^n - c_1(n-1)^m r_0^{n-1} - c_2(n-2)^m r_0^{n-2} - c_3(n-3)^m r_0^{n-3} - \cdots - c_k(n-k)^m r_0^{n-k} = 0. \quad (4)$$

Given any polynomial $P(x) = \sum_{i=0}^{d} b_i x^i$, we can define[1] another polynomial $\hat{P}(x) = \sum_{i=0}^{d} i b_i x^i$.

Now define a sequence of polynomials: $p_0(r) = p(r)$ and $p_j(r) = \hat{p}_{j-1}(r)$. More explicitly,

$$p_j(r) = n^j r^n - c_1(n-1)^j r^{n-1} - c_2(n-2)^j r^{n-2} - \cdots - c_k(n-k)^j r^{n-k}.$$

Notice that our goal of showing Equation (4) holds is now simply to show that $p_m(r_0) = 0$. To do this, we first prove a bunch of facts about how $\hat{P}(x)$ can be expressed if we know $P(x)$.

Lemma 21 *Suppose $P_1(x)$ and $P_2(x)$ are polynomials, and $P(x) = P_1(x)P_2(x)$. Then $\hat{P}(x) = P_1(x)\hat{P}_2(x) + \hat{P}_1(x)P_2(x)$.*

Proof: Let $P_1(x) = \sum_{i=0}^{d} b_i x^i$ and $P_2(x) = \sum_{j=0}^{e} d_j x^j$. Then

$$\begin{aligned} P(x) &= \sum_{i=0}^{d}\sum_{j=0}^{e} b_i d_j x^{i+j} \text{ and} \\ \hat{P}(x) &= \sum_{i=0}^{d}\sum_{j=0}^{e} (i+j) b_i d_j x^{i+j}. \end{aligned}$$

[1] For those of you who have taken calculus, the definition of $\hat{P}(x)$ is actually $xP'(x)$. Then Lemma 21, below, is just the product rule for derivatives and Lemma 22, below is obtained trivially by taking the derivative of $(x-c)^m$.

We also have

$$\begin{aligned}
P_1(x)\hat{P}_2(x) + \hat{P}_1(x)P_2(x) &= \left(\sum_{i=0}^{d} b_i x^i\right)\left(\sum_{j=0}^{e} j d_j x^j\right) + \left(\sum_{i=0}^{d} i b_i x^i\right)\left(\sum_{j=0}^{e} d_j x^j\right) \\
&= \sum_{i=0}^{d}\sum_{j=0}^{e}(j b_i d_j x^{i+j} + i b_i d_j x^{i+j}) \\
&= \sum_{i=0}^{d}\sum_{j=0}^{e}(i+j) b_i d_j x^{i+j}.
\end{aligned}$$

■

Lemma 22 *For $m \geq 1$, if $P(x) = (x-c)^m$ then $\hat{P}(x) = xm(x-c)^{m-1}$.*

Proof: We prove this claim by induction on m.
Base case ($m = 1$): $P(x) = x - c$, so $\hat{P}(x) = x = x \cdot 1 \cdot (x-c)^0$.
Induction step: Let $m \geq 2$. Assume the claim is true for $m-1$. Let $P(x) = (x-c)^m = (x-c)(x-c)^{m-1}$. Then,

$$\begin{aligned}
\hat{P}(x) &= (x-c)x(m-1)(x-c)^{m-2} + x(x-c)^{m-1} \text{ by ind. hyp. and Lemma 21} \\
&= x(m-1)(x-c)^{m-1} + x(x-c)^{m-1} \\
&= (xm - x + x)(x-c)^{m-1} \\
&= xm(x-c)^{m-1}.
\end{aligned}$$

■

Lemma 23 *Let $Q(x)$ be a polynomial and let $P_0(x) = (x-c)^{m+1}Q(x)$. Let $P_j(x) = \hat{P}_{j-1}(x)$ for $j \geq 1$. Then, for $0 \leq j \leq m$, there exists a polynomial $Q_j(x)$ such that $P_j(x) = (x-c)^{m+1-j}Q_j(x)$.*

Proof: We prove this claim by induction on j.
Base case ($j = 0$): We take $Q_0(x) = Q(x)$ so that $P_0(x) = (x-c)^{m+1-0}Q_0(x)$.
Induction step: Let $j \geq 1$ and $j \leq m$. Assume the claim holds for $j-1$. Then, $P_j(x) = \hat{P}_{j-1}(x)$ by definition of P_j and $P_{j-1}(x) = (x-c)^{m+1-j+1}Q_{j-1}(x)$ by the induction hypothesis. Applying Lemmas 21 and 22, we get

$$\begin{aligned}
P_j(x) &= \hat{P}_{j-1}(x) \\
&= (x-c)^{m+1-j+1}\hat{Q}_{j-1}(x) + x(m+1-j+1)(x-c)^{m+1-j}Q_{j-1}(x) \\
&= (x-c)^{m+1-j}[(x-c)\hat{Q}_{j-1}(x) + x(m+1-j+1)Q_{j-1}(x)].
\end{aligned}$$

By taking $Q_j(x) = (x-c)\hat{Q}_{j-1}(x) + x(m+1-j+1)Q_{j-1}(x)$, we complete the induction step. ■

Now, going back to the polynomial p_m that we defined based on Equation (1), recall that we wanted to show that $p_m(r_0) = 0$ in order to establish Equation (4). We assumed that $p_0(r) = (r-r_0)^{m+1}q(r)$ for some polynomial $q(r)$. Thus, by Lemma 23 (taking $j = m$), we have $p_m(r) = (r-r_0)^1 q'(r)$ for some polynomial q'. Thus, $p_m(r_0) = 0$, completing the proof of Proposition 19.

4.2 Linear Non-homogeneous Recurrences

A *linear non-homogeneous recurrence relation of degree k with constant coefficients* has the form

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + c_3 a_{n-3} + \cdots + c_k a_{n-k} + f(n). \tag{5}$$

Once again, the c_i's are constants. The difference between this recurrence equation and a linear homogeneous recurrence is the extra $f(n)$ term, where f can be any function. (Notice that linear homogeneous recurrences are a special case where $f(n) = 0$.) In order to define a sequence properly, this recurrence has to be combined with base cases that give the first k elements of the sequence. We have already seen several examples of non-homogeneous linear recurrences: Example 3 had $f(n) = 1$, Example 6 had $f(n) = 8n - 2$, Example 7 had $f(n) = n^2$ and Example 11 had $f(n) = 2n$.

We are going to use an approach similar to the one we used for homogeneous recurrences: first we shall try to find a family of sequences that satisfy the recurrence equation, ignoring the base cases, and then choose the one that does satisfy the base cases. The nonhomogeneous recurrence (Equation (5)) has an *associated homogeneous recurrence equation* which is obtained just by dropping the $f(n)$ term:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + c_3 a_{n-3} + \cdots + c_k a_{n-k} \tag{6}$$

There is an important relationship between sequences that satisfy Equation (5) and sequences that satisfy Equation (6): the *difference* between any two sequences that satisfy Equation (5) must satisfy Equation (6).

Proposition 24 *Let a_n be a sequence that satisfies Equation (5). Another sequence b_n satisfies Equation (5) if and only if $h_n = b_n - a_n$ satisfies Equation (6).*

Proof: First we prove the "if" part of the proposition. Suppose a_n satisfies Equation (5) and h_n satisfies Equation (6). So we have

$$\begin{aligned} a_n &= c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f(n), \text{ and} \\ h_n &= c_1 h_{n-1} + c_2 h_{n-2} + \cdots + c_k h_{n-k}. \end{aligned}$$

Adding these two equations, we get

$$\begin{aligned} a_n + h_n &= c_1(a_{n-1} + h_{n-1}) + c_2(a_{n-2} + h_{n-2}) + \cdots + c_k(a_{n-k} + h_{n-k}) + f(n), \text{ so} \\ b_n &= c_1 b_{n-1} + c_2 b_{n-2} + \cdots + c_k b_{n-k} + f(n), \end{aligned}$$

since $b_n = a_n + h_n$.

Now we prove the "only if" part: Suppose a_n satisfies Equation (5) and b_n satisfies Equation (6). So we have

$$\begin{aligned} a_n &= c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} + f(n), \text{ and} \\ b_n &= c_1 b_{n-1} + c_2 b_{n-2} + \cdots + c_k b_{n-k} + f(n). \end{aligned}$$

Subracting the first equation from the second, we get

$$\begin{aligned} b_n - a_n &= c_1(b_{n-1} - a_{n-1}) + c_2(b_{n-2} - a_{n-2}) + \cdots + c_k(b_{n-k} - a_{n-k}), \text{ so} \\ h_n &= c_1 h_{n-1} + c_2 h_{n-2} + \cdots + c_k h_{n-k}, \end{aligned}$$

since $h_n = b_n - a_n$. ■

So, if we have a single sequence a_n that satisfies Equation (5), we can get *every* sequence that satisfies Equation (5) just by adding a sequence h_n that satisfies Equation (6) to a_n. But we already know how to find sequences h_n that satisfy Equation (6): we developed a complete technique for finding solutions to linear homogeneous recurrences in Section 4.1. This means that we just have to figure out a way to find *one* sequence that satisfies Equation (5).

Finding that one particular sequence that satisfies Equation (5) can be difficult. Fortunately, it turns out that, for many common functions $f(n)$ it is possible to find a solution whose formula is similar to $f(n)$ itself. For example, if $f(n)$ is a polynomial function, there will always be a a polynomial sequence that satisfies Equation (5), which can be found by guessing that the sequence is a polynomial and then figuring out what the coefficients in that polynomial are. (The proof that such a polynomial always exists is quite lengthy. Since we are going to be explicitly finding that polynomial when we solve Equation (5), we do not need the general proof that it always exists, but it is comforting to know that it does.) Let's look at an example.

Example 25 Consider the recurrence

$$\begin{aligned} a_0 &= 2 \\ a_1 &= 3 \\ a_n &= a_{n-1} + a_{n-2} + 3n + 1 \text{ for } n \geq 2. \end{aligned}$$

As before, we are going to focus on first finding sequences that satisfy the recurrence equation, and then figure out which one also satisfies the base cases.

Our first task is to find a particular solution a'_n to the non-homogeneous recurrence equation. Here the function $f(n)$ is a linear polynomial $3n + 1$. So let's try guessing that there is a sequence a'_n that is also given by a linear polynomial: $a'_n = cn + b$ (where c and b are constants that we shall figure out). Plugging this into the recurrence equation, we get

$$\begin{aligned} cn + b &= c(n-1) + b + c(n-2) + b + 3n + 1 \\ \leftrightarrow \quad cn + b &= cn - c + b + cn - 2c + b + 3n + 1 \\ \leftrightarrow \quad 0 &= (3 + c)n + (b - 3c + 1) \end{aligned}$$

In order to make this true for all $n \geq 2$, we must have $3+c = 0$ and $b-3c+1 = 0$. This means we should choose $c = -3$ and $b = -10$. So, we see that $a'_n = -3n-10$ satisfies the recurrence equation. However, this sequence does not satisfy the base cases, so we still need to search for another sequence a_n that satisfies both the recurrence equation and the base cases.

Fortunately, we know, by Proposition 24, that the sequence we are looking for must be of the form $a_n = a'_n + b_n$, where b_n is a solution to the homogeneous equation associated with our recurrence:

$$b_n = b_{n-1} + b_{n-2}.$$

This is just the Fibonacci recurrence, so we already saw in Example 17 that all sequences of the form $b_n = \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$ satisfy this equation (where α_1 and α_2 are constants). So (by Proposition 24) we know that any sequence of the form

$$a_n = -3n - 10 + \alpha_1 \left(\frac{1+\sqrt{5}}{2}\right)^n + \alpha_2 \left(\frac{1-\sqrt{5}}{2}\right)^n$$

satisfies our non-homogeneous recurrence equation. Let's see if we can pick α_1 and α_2 so that the base cases are also satisfied. Plugging in $n = 0$ and $n = 1$ we get

$$\begin{aligned} -10 + \alpha_1 + \alpha_2 &= 2 \\ -13 + \frac{1+\sqrt{5}}{2}\alpha_1 + \frac{1-\sqrt{5}}{2}\alpha_2 &= 3 \end{aligned}$$

If we solve these for α_1 and α_2, we get $\alpha_1 = 6 + 2\sqrt{5}$ and $\alpha_2 = 6 - 2\sqrt{5}$. So we have successfully found a solution to the recurrence:

$$a_n = -3n - 10 + (6 + 2\sqrt{5}) \left(\frac{1+\sqrt{5}}{2}\right)^n + (6 - 2\sqrt{5}) \left(\frac{1-\sqrt{5}}{2}\right)^n.$$

■

5 Bounding Recurrences

Sometimes, the nature of a recurrence makes it difficult to come up with an explicit formula for the sequence it defines. In some cases, we might be satisfied if we can, instead, give upper or lower bounds on the size of the elements. For example, we might be interested in describing the asymptotic behaviour of the sequence using big-O or big-Ω notation.

The most basic method for accomplishing this is again to guess a bound (after looking at the first few values of the sequence, or using any other information about the sequence we might have). Proving the bound is then usually fairly straightforward, using an induction argument, as in the following examples.

Example 26 Consider the recurrence[2] of Example 2:

$$\begin{aligned} T(1) &= 1 \\ T(n) &= T(\lceil n/2 \rceil) + 4, \text{ for } n \geq 2. \end{aligned}$$

[2] For those who have done some recursive programming, this recurrence describes the running time of a particular implementation of binary search on an array of size n.

We want to come up with a guess about an upper bound for the sequence defined by this recurrence. In order to avoid having to worry about the ceilings, we can look at the values of $T(n)$ when n is a power of 2: $T(1) = 1, T(2) = 5, T(4) = 9, T(8) = 13, T(16) = 17$. It's pretty clear from the recurrence that the sequence values depend linearly on the exponent on the 2: we would guess that $T(2^i) = 4i - 3$ (and it would be very easy to verify this by induction on i). From this, we can guess that, for all n, $T(n)$ is $O(\log_2 n)$, since $i = \log_2(2^i)$.

Notice that the statement "$T(n)$ is $O(\log_2 n)$" is *not* a claim that we can prove by induction on n because it *does not say anything about* n! In that statement, n is just a dummy variable. Remember that the statement is shorthand for "There exist c, k such that for all $n \geq k$, $T(n) \leq c \log_2 n$." Since we want to prove a bound on $T(n)$ by induction on n, we have to formulate a claim that *can* be proved by induction. Let's try to show $T(n) \leq c \log_2 n$. (As we go through the proof, we shall pick a value for c that makes the proof work.) This claim does not make sense for $n = 0$, since $\log_2 0$ is undefined, and there is not much hope of proving it for $n = 1$, since $\log_2 1 = 0$, so we shall just prove it for $n \geq 2$. We shall use strong induction because the recurrence relates $T(n)$ not to $T(n-1)$ but to some earlier term in the sequence.

Claim: For all $n \geq 2$, $T(n) \leq c \log_2 n$.

Proof (by induction on n):

Base case ($n = 2$): $T(2) = T(1) + 4 = 5 = 5 \log_2 2$. Thus, the base case holds as long as we choose $c \geq 5$.

Induction step: Let $n \geq 3$. Assume that, for $2 \leq k < n$, $T(k) \leq c \log_2 k$. Then,

$$\begin{aligned}
T(n) &= T(\lceil n/2 \rceil) + 4 \text{ since } n \geq 2 \\
&\leq c \log_2(\lceil n/2 \rceil) + 4 \text{ by ind. hyp., since } 2 = \lceil 3/2 \rceil \leq \lceil n/2 \rceil \leq \frac{n+1}{2} < n \\
&\leq c \log_2 \frac{n+1}{2} + 4 \text{ since } \log_2 \text{ is increasing and } \lceil n/2 \rceil \leq \frac{n+1}{2} \\
&\leq c \log_2 \left(\frac{2n}{3}\right) + 4 \text{ since } \log_2 \text{ is increasing and } \frac{n+1}{2} \leq \frac{2n}{3} \text{ for } n \geq 3 \\
&= c \log_2 n + c \log_2 \left(\frac{2}{3}\right) + 4
\end{aligned}$$

To complete the induction step, we just have to make sure that c is chosen so that $c \log_2 \left(\frac{2}{3}\right) + 4 \leq 0$, which is true whenever $c \geq \frac{4}{-\log_2(2/3)} = 6.838045\ldots$. So if we pick $c = 7$, that will work for both the base case and the induction step.

Now, if you were designing a nice, polished proof, you might want to go back and replace c by 7 everywhere in the above argument. We just left the value of c undefined initially so that you would see how the constant 7 was discovered. ■

Exercise 27 Prove that $T(n)$ defined in the preceding example is $\Omega(\log_2 n)$.

Sometimes, proving a bound on a recursively-defined sequence is not quite so straightforward.

Example 28 Consider the recurrence[3]

$$\begin{aligned} T(1) &= 0 \\ T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1, \text{ for } n \geq 2. \end{aligned}$$

This recurrence is very similar to the one in Example 4, except for the +1. So we might guess that this function will also grow linearly with n. Let's guess that $T(n) \leq cn$, where c is a constant that we shall figure out in the course of the proof.

Claim: $T(n) \leq cn$ for all $n \geq 1$.

Attempted Proof by strong induction on n:

Base case: $T(1) = 0 \leq c \cdot 1$ as long as we choose $c \geq 1$.

Induction step: Let $n \geq 2$. Assume that $T(k) \leq ck$ for $1 \leq k < n$. Then,

$$\begin{aligned} T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1 \text{ since } n \geq 2 \\ &\leq c\lfloor n/2 \rfloor + c\lceil n/2 \rceil + 1 \text{ by ind. hyp., since } 1 \leq \lfloor 2/2 \rfloor \leq \lfloor n/2 \rfloor \leq \lceil n/2 \rceil \leq \frac{n+1}{2} < n \\ &= cn + 1 \end{aligned}$$

Now we are stuck: we wanted to prove $T(n) \leq cn$, but we only managed to prove that $T(n) \leq cn + 1$. So we have to abandon this proof.

It is actually easier to prove a stronger claim: $T(n) \leq cn - d$, where c and d are constants that will be determined during the proof. (It may seem paradoxical that a stronger claim is easier to prove, but this is because we can use, in the induction step, a stronger induction hypothesis. This is sometimes called *inductive loading*.)

Claim: $T(n) \leq cn - d$ for all $n \geq 1$.

Proof by strong induction on n:

Since we had trouble last time with the induction step, let's do that first and come back to the base case later.

Induction step: Let $n \geq 2$. Assume that $T(k) \leq ck - d$ for $1 \leq k < n$. Then,

$$\begin{aligned} T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + 1 \text{ since } n \geq 2 \\ &\leq c\lfloor n/2 \rfloor - d + c\lceil n/2 \rceil - d + 1 \text{ by ind. hyp., since } 1 \leq \lfloor 2/2 \rfloor \leq \lfloor n/2 \rfloor \leq \lceil n/2 \rceil \leq \frac{n+1}{2} < n \\ &= cn - d + (1 - d) \end{aligned}$$

The induction step will be complete as long as we choose d so that $(1 - d) \leq 0$. Let's choose $d = 1$.

Base case: $T(1) = 0 \leq c \cdot 1 - 1$ as long as we choose $c \geq 1$.

Thus, the entire proof works if we have $c = 1$ and $d = 1$, so $T(n) \leq n - 1$ for all $n \geq 1$. ■

[3] For those of you who have done some programming, this recurrence describes, for example, the running time of a divide-and-conquer algorithm for finding the smallest element in an array of n elements.

6 Divide-and-Conquer Recurrences

The name of these recurrences comes from the Roman maxim "Divide et impera". This type of recurrence arises frequently in computer science when a problem is solved by first dividing the problem into smaller subproblems, then solving those subproblems, and then combining the solutions for those smaller subproblems to solve (or conquer) the original problem. The subproblems are of the same form as the original problem (except smaller), so they can be solved using the same technique: subdivide them still further, solve the sub-subproblems and combine results to solve the subproblem. (The sub-subproblems are again divided up, and this continues until the subproblems are so small that they can be solved trivially.)

If this approach is used (and the size of the subproblems is a constant fraction of the original problem size) the time required (as a function of the problem size) can be expressed using a *divide-and-conquer recurrence*, which has the form

$$T(n) = aT(n/b) + f(n), \text{ for } n > 1$$

where $a \geq 1$ and $b > 1$ are constants In order to make the recurrence well defined, the $T(n/b)$ term will actually be either $T(\lfloor n/b \rfloor)$ or $T(\lceil n/b \rceil)$. We can even have both in the same recurrence equation. To keep things simple, we shall assume b is an integer (which is the most frequently encountered case anyway). The recurrence will also have to have associated base cases. Here, we assume that the base case defines $T(1)$. If the recurrence includes floors (and $b > 2$) then it is also necessary to define $T(0)$ as a base case. Notice that no more base cases are required: If $n > 1$, $0 \leq \lfloor n/b \rfloor \leq \lceil n/b \rceil \leq \frac{n}{b} + (1 - \frac{1}{b}) < n$ since $1 - \frac{1}{b} < n(1 - \frac{1}{b})$. For simplicity, we shall assume that $T(0) = T(1)$ if $T(0)$ needs to be defined. (The exact nature of the base cases does not really affect the asymptotic behaviour of the recurrence, which is what we shall be studying: that behaviour really only depends on a, b and $f(n)$, but our extra assumptions about the base cases will make the proofs below simpler.)

We have already seen several examples of divide-and-conquer recurrences. In Examples 2 and 26 we had $a = 1, b = 2$ and $f(n) = 4$. In Example 28, we had $a = 2, b = 2$ and $f(n) = 1$. The function $f(n)$ need not be constant. For example, consider the recurrence[4]

$$\begin{aligned} T(1) &= 0 \\ T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n, \text{ for } n \geq 2. \end{aligned}$$

Here, we have $a = 2, b = 2$ and $f(n) = n$.

We are going to come up with a theorem that gives the asymptotic behaviour of any sequence defined by a divide-and-conquer recurrence with $f(n)$ is a monomial (*i.e.*, $f(n) = c \cdot n^d$ for constants $c > 0$ and $d \geq 0$). The results of this section are summarized in the following theorem, which is sometimes called the Master Theorem. (Actually, the Master Theorem is a little more general, since it handles other types of functions $f(n)$ too.)

[4] For programmers: this recurrence gives the running time of mergesort.

Theorem 29 *If a sequence is defined by a recurrence equation*

$$T(n) = aT(n/b) + cn^d \quad (\textit{for } n > 1)$$

where $a \geq 1, b \geq 2, c > 0$ *and* $d \geq 0$ *are constants and* n/b *is actually either* $\lfloor n/b \rfloor$ *or* $\lceil n/b \rceil$*, then one of the following holds (depending on the values of* a, b *and* d*).*
$T(n)$ *is* $\Theta(n^d)$ *if* $a < b^d$
$T(n)$ *is* $\Theta(n^d \log n)$ *if* $a = b^d$
$T(n)$ *is* $\Theta(n^{log_b a})$ *if* $a > b^d$*.*

We shall derive these formulas by starting with the special case where n is a power of b: $n = b^k$. This avoids any messy floors and ceilings. For this special case, we shall actually be able to come up with an exact expression for $T(n)$. We can arrive at the formula by doing repeated substitutions.

$$\begin{aligned}
T(b^k) &= aT(b^{k-1}) + cb^{kd} \\
&= a^1T(b^{k-1}) + cb^{kd} \\
&= a^1(aT(b^{k-2}) + cb^{(k-1)d}) + cb^{kd} \\
&= a^2T(b^{k-2}) + cb^{kd} + acb^{(k-1)d} \\
&= a^2(aT(b^{k-3}) + cb^{(k-2)d}) + cb^{kd} + acb^{(k-1)d} \\
&= a^3T(b^{k-3}) + cb^{kd} + acb^{(k-1)d} + a^2cb^{(k-2)d} \\
&= a^3(T(b^{k-4}) + cb^{(k-3)d}) + cb^{kd} + acb^{(k-1)d} + a^2cb^{(k-2)d} \\
&= a^4T(b^{k-4}) + cb^{kd} + acb^{(k-1)d} + a^2cb^{(k-2)d} + a^3cb^{(k-3)d} \\
&\vdots
\end{aligned}$$

If we imagine continuing this for k iterations, we conjecture that

$$T(b^k) = a^kT(1) + \sum_{i=0}^{k-1} ca^ib^{(k-i)d} = a^kT(1) + cb^{kd}\sum_{i=0}^{k-1}\left(\frac{a}{b^d}\right)^i. \tag{7}$$

(Exercise: prove this conjecture by induction on k.)

Now, let's try to compute the sum in the formula above. We notice that it is just a geometric series. We have a formula for that sum (which works as long as $\frac{a}{b^d} \neq 1$.

Case 1 ($a \neq b^d$): In this case, we can use the formula for geometric sums to compute the sum in Equation (7). We have

$$\begin{aligned}
T(b^k) &= a^kT(1) + cb^{kd}\frac{1 - \left(\frac{a}{b^d}\right)^k}{1 - \frac{a}{b^d}} \\
&= a^kT(1) + \frac{cb^{kd}}{1 - \frac{a}{b^d}} - \frac{cb^{kd}\frac{a^k}{b^{kd}}}{1 - \frac{a}{b^d}} \\
&= \left(T(1) - \frac{c}{1 - \frac{a}{b^d}}\right)a^k + \frac{c}{1 - \frac{a}{b^d}} \cdot b^{kd} \\
&= c'a^k + c''b^{kd} \qquad \text{(where } c' = T(1) - \tfrac{c}{1-\frac{a}{b^d}} \text{ and } c'' = \tfrac{c}{1-\frac{a}{b^d}}\text{).}
\end{aligned}$$

But $a^k = a^{\log_b n} = a^{\log_a n \log^b a} = n^{\log_b a}$ and $b^{kd} = n^d$. So, (when n is a power of b),

$$T(n) = c' n^{\log_b a} + c'' n^d.$$

If $a < b^d$, then $\log_b a < d$, so $T(n)$ is $\Theta(n^d)$. If $a > b^d$, then $\log_b a > d$, so $T(n)$ is $\Theta(n^{\log_b a})$. Both of these conclusions hold *only when n is a power of b.* We shall deal with n's that are not powers of b below.

Case 2 ($a = b^d$): In this case, every term in the sum in Equation (7) is 1, so the sum is just equal to the number of terms in it. Thus, we have

$$\begin{aligned} T(b^k) &= a^k T(1) + cb^{kd} \cdot k \\ &= b^{kd} T(1) + cb^{kd} \cdot k \text{ since } a = b^d \\ &= b^{kd}(T(1) + ck) \\ &= n^d (T(1) + c \log_b n) \text{ since } n = b^k \end{aligned}$$

Since $T(1)$ and c are just constants, it is easy to see that $T(n)$ is $\Theta(n^d \log n)$, *when n is a power of b.*

Although the above arguments look a little messy and complicated, that is just because the constants are complicated expressions; there was really nothing more difficult in computing the formulas than a geometric sum and making sure we did our arithmetic correctly.

Now, in each of the three cases, we have shown that $T(n)$ satisfies the bounds given in Theorem 29, but we have only done this for values of n that are powers of b. Fortunately, this is enough to conclude that the bounds are accurate for all values of n as we shall prove in the following lemmas. Because $T(n)$ is a non-decreasing function, and we have shown that it satisfies the asymptotic bounds at infinitely many values of n, we shall be able to easily show that it satisfies the asymptotic bounds everywhere.

Lemma 30 *If T is defined by the recurrence equation $T(n) = aT(n/b) + cn^d$ for $n > 1$ (along with base cases $T(1)$ and, if necessary, $T(0) = T(1)$), then $T(n)$ is non-decreasing (for all n).*

Proof: We use strong induction on n to prove that, for all $n \geq 1$, $T(n) \geq T(n-1)$.

Base cases $T(1) \geq T(0)$ by assumption. Using $2/b$ to represent either $\lceil 2/b \rceil$ or $\lfloor 2/b \rfloor$, depending on the recurrence, we have $T(2) = aT(2/b) + c2^d \geq T(2/b) = T(1)$ since $a \geq 1$ and $c > 0$.

Induction step: Let $n \geq 3$. Assume that $T(k) \geq T(k-1)$ for $1 \leq k < n$. We must show that $T(n) \geq T(n-1)$. Again, using n/b to represent either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$, depending on the recurrence, we have

$$\begin{aligned} T(n) - T(n-1) &= aT(n/b) + cn^d - aT((n-1)/b) + c(n-1)^d \\ &\geq a(T(n/b) - T((n-1)/b)). \end{aligned}$$

But $T(n/b) - T((n-1)/b)$ is either 0 (if $n/b = (n-1)/b$ because of the rounding) or positive, by the induction hypothesis. This completes the induction step. ■

Lemma 31 *Let p be a constant. Suppose $T(n)$ is a non-decreasing function and $c_1 n^p \le T(n) \le c_2 n^p$ whenever n is a (large) power of 2. Then, $T(n)$ is $\Theta(n^p)$.*

Proof: For large n, choose k such that $b^k \le T(n) < b^{k+1}$. Then,

$$\begin{aligned} T(n) &\le T(b^{k+1}) && \text{since } T \text{ is non-decreasing and } n \le b^{k+1} \\ &\le c_2 b^{(k+1)p} \\ &= bc_2 b^{kp} \\ &\le bc_2 n^p && \text{since } b^k \le n \end{aligned}$$

and

$$\begin{aligned} T(n) &\ge T(b^k) && \text{since } T \text{ is nondecreasing and } n \ge b^k \\ &\ge c_1 b^{kp} \\ &\ge \tfrac{c_1}{b} b^{(k+1)p} \\ &> \tfrac{c_1}{b} n^p && \text{since } b^{k+1} > n. \end{aligned}$$

So, for all large n, $\frac{c_1}{b} n^p < T(n) \le c_2 b n^p$. Thus, $T(n)$ is $\Theta(n^p)$. ■

It follows from the reasoning in Case 1 above, Lemma 30 and Lemma 31 that the first and third cases of Theorem 29 are true.

Exercise 32 Prove a lemma similar to Lemma 31 to complete the proof of the second case of Theorem 29.

6.1 Applying the Master Theorem

Proving the Master Theorem was a little tricky, but applying it is simple.

In Examples 2 and 26 we had $a = 1, b = 2, c = 4$, and $d = 0$. Thus, $a = 1 = b^d$ so the second case of the Master Theorem applies and $T(n)$ is $\Theta(n^0 \log n) = \Theta(\log n)$.

In Example 28, we had $a = 2, b = 2$ and $c = 1$ and $d = 0$. Thus, $a = 2 > 1 = b^d$, so the third case of Theorem 29 applies, and we have $T(n)$ is $\Theta(n^{\log_2 2}) = \Theta(n)$.

For the recurrence

$$\begin{aligned} T(1) &= 0 \\ T(n) &= T(\lfloor n/2 \rfloor) + T(\lceil n/2 \rceil) + n, \text{ for } n \ge 2 \end{aligned}$$

we have $a = 2, b = 2, c = 1$ and $d = 1$. Here, $a = 2 = b^d$ so the second case of the Master Theorem applies and $T(n)$ is $\Theta(n \log n)$.

For the recurrence

$$\begin{aligned} x_0 &= 1 \\ x_n &= 7x_{\lfloor n/5 \rfloor} + 9n^2 \text{ for } n \ge 1 \end{aligned}$$

we have $a = 7, b = 5, c = 9$ and $d = 2$ so $a = 7 < 25 = b^d$ and the first case of the Master Theorem applies. It follows that x_n is $\Theta(n^2)$.