Recurrence Relations

1 Introduction

In certain instances, it is sometimes easier to provide a definition of an object in terms of *itself*. The general name for this concept is **recursion**. We see examples of recursive definitions very often, as in "*the virus doubles every hour*", or "*your money will grow by 10% every year*". Note that both of these descriptions relate the amount of something to its value at a previous date. We can define sequences by describing a relationship between an element of the sequence and some *previous* elements, as we saw in our section on sequences. The same could be done for functions, or even structural objects such as sets. In your further studies, you will most likely see this concept applied to *algorithms*, in studying *recursive algorithms*. In this section, we will introduce recursively defined functions and sequences.

2 Recursive definitions

Recursive definitions of sequences (and functions) are based on describing a *relationship* between the elements of the sequence. Such relationships are often called *recurrence relations*. As an example of such a relationship, consider a petri dish of bacteria, where the *number* of bacteria doubles every hour. Let a_n represent the number of bacteria after n hours. The *relationship* is given by $a_n = 2a_{n-1}$. This equation states that the number of bacteria at hour n is exactly double what it was at hour $n-1$, thus *relating* the nth element of the sequence a_n to a *previous element*. In order to completely determine the number of bacteria over time, we also need to know the *initial number* of bacteria in the dish. Certainly, if we start with 5 bacteria, then after 1 hour, we have 10 bacteria, whereas if we start with only 1 bacteria, then after the first hour we only have 2 bacteria. Thus in order to complete the description of the problem, we would also need to specify the *initial conditions*, for example, that $a_1 = 5$. The relationship, together with the initial condition, completely specify the elements of the sequence.

2.1 Recursively defined sequences:

In order to completely describe a sequence using a recursive definition, we require the two steps mentioned above:

- **Initial terms:** Specify a finite number of initial elements, such as $a_0, a_1, \ldots, a_{k-1}$.
- **Recurrence relation:** An equation that expresses a_n in terms of one or more previous terms of the sequence, for all $n \geq k$. The values of a_n that are *not* included in the initial terms can be found by using the recurrence relation.

Example 1. *Write the first 5 terms of the sequence defined by* $a_0 = 1, a_1 = 2, a_n = a_{n-1} + 3a_{n-2}, n \geq 2$.

Solution: We use the recurrence relation for a_2, a_3 and a_4, since a_0 and a_1 are included in the initial terms. We solve for them *in increasing order* of n using the recurrence relation: $a_2 = a_1 + 3a_0 = 2 + 3(1) = 5$, then $a_3 = a_2 + 3a_1 = 5 + 3(2) = 11$, and finally $a_4 = a_3 + 3a_2 = 11 + 3(5) = 26$.

Example 2. *(From introduction). Let a_n be the number of bacteria in a petri dish after n hours. At hour 1, there are 5 bacteria in the dish and the number of bacteria doubles every hour*

Solution: The initial value is $a_1 = 5$ and the recurrence relation is $a_n = 2a_{n-1}$. Thus $a_2 = 2(5) = 10, a_3 = 2(10) = 20, a_4 = 2(20) = 40$, and $a_5 = 2(40) = 80$.

Notice that we don't have a *closed-form* equation for determining the elements of the sequence. Instead, in the above two examples, we found the elements of the sequence *iteratively* by starting with the initial terms and working through the elements one at a time using the recurrence relation.

2.2 Recursively defined functions:

Functions that are defined on the set of non-negative integers can also be defined recursively. The two steps are identical, as shown in the next example.

Example 3. *Let $f(n)$ be defined recursively as:*

$$f(0) = 3$$

$$f(n+1) = 3f(n) + 6$$

Find $f(4)$.

Solution: Since we do not have a closed-form example, we cannot simply "plug-in" the value of $n = 4$ and determine the value of $f(4)$. Instead, we use the recursive definition to determine $f(1), f(2), f(3), f(4)$.

$$\begin{aligned} f(1) &= 3f(0) + 6 = 15 \\ f(2) &= 3f(1) + 6 = 51 \\ f(3) &= 3f(2) + 6 = 159 \\ f(4) &= 3f(3) + 6 = 483 \end{aligned}$$

Although we usually focus on how to convert a recursive definition to a closed-formula, it is also possible to do the opposite, as shown in the following example.

Example 4. *Define $f(n) = n!$ using a recursive definition.*

Solution: Recall that $n! = n(n-1)(n-2)\dots 1$. Using the closed-formula, if we were asked to evaluate $f(6)$ for example, we would simply apply $f(6) = 6!$. In order to describe this using a recursive definition, we need to identify a *relationship* between the value of $f(n)$ and some previous values. Note that $n! = n(n-1)!$, and so a relationship can be easily described as $f(n) = nf(n-1)$. It remains only to define the initial terms. Normally we think of $n = 1$ is being the smallest term for which we take the factorial. However $0!$ is defined as $0! = 1$. Thus the initial term of the recursion will be $f(0) = 1$. The recursive definition of $f(n) = n!$ is given by:

$$f(0) = 1$$

$$f(n) = nf(n-1)$$

Example 5. *Let $f(n) = n^2$ for $n \geq 1$. Give a recursive definition of this function.*

Solution: Certainly $f(1) = 1$. Then we need a way to relate $f(n)$ to $f(n-1)$. Notice that $(n-1)^2 = n^2 - 2n + 1$. Thus $f(n-1) = f(n) - 2n + 1$. Our recursive definition is:

$$f(1) = 1$$

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

Your recursive definition must be well-defined:

When describing a function, the two steps that define the recurrence must describe a *well-defined* function. Suppose for example that we define a function $f(n)$ as:

$$f(0) = 0, f(n) = 2f(n-2)$$

In this definition, we cannot determine the value of $f(1)$ using the recursive step. Thus this function is *not* well-defined.

Example 6. *Give a recursive definition of the binomial coefficients*

Solution:. We saw in a previous lecture that the binomial coefficients satisfied Pascal's identity:

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

It is not enough to simply state this as the recursive definition, since we require some *initial elements* which will guarantee that we can find $\binom{n}{k}$ using this recurrence and that we will not end up with an ill-defined recurrence. The "*outer edges*" of Pascal's triangle form the initial elements. Thus the initial elements of the recursive definition are:

$$\binom{0}{0} = 1$$

$$\binom{n}{0} = 1$$

$$\binom{n}{n} = 1$$

This defines all the "1"s that form the outer edges of the triangle. Then any inner value can be found using the the recursive step : for $0 < k < n$

$$\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$$

2.3 Recursively defined sets

A set S can be described recursively, by defining an *initial set* of elements that belong to S and then a *recursive rule* to include more elements in S. Let's consider the following example which creates the set of all even positive numbers:

Example 7. *Let S be the set of all even positive integers. Define the set S recursively as:*

- *Initial elements:* $2 \in S$
- *Recursive step: if $x, y \in S$, then $x + y \in S$*

Show that this defines the set of positive even numbers.

Solution:. The recursive step above can be applied for any elements x and y in S. Each time we apply it, we "discover" a new element in S. Initially we know that $2 \in S$, but by setting $x = 2$ and $y = 2$, then $x + y = 4 \in S$. Now we know that the set S contains 2 and 4. Continuing in this way, we find that $2 + 4 = 6$ and thus $6 \in S$. Repeating this procedure yields $S = \{2, 4, 6, 8, \ldots\}$.

3 Modelling problems with Recurrences

In this section, we look at how to take a specific relationship and turn it into a recursive definition. As we saw above, one such relationship was that of number of bacteria in the dish that doubled every hour, and we were able to express that relationship using the equation $a_n = 2a_{n-1}$.

Example 8. *Write a recursive definition for the population after n years in Figi, where currently the population is* 2000 *and each year the population increases by* 10%*.*

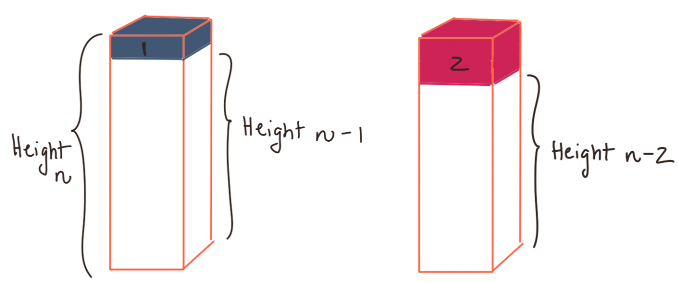
Solution:. Let p_n be the population in Figi after n years. The initial population is given by $p_0 = 2000$. The population each year *increases* by 10%, in other words, the new population at year n is the old population (p_{n-1}) *plus* another 10 percent. We can write this as $p_n = p_{n-1} + .10p_{n-1}$. Therefore, the recursive definition for this population is:

$$p_0 = 2000$$

$$p_n = (1.1)p_{n-1}, n \geq 1$$

Example 9. *Write a recursive definition for the number of ways of building a tower of height n using blocks of size* 2 *or* 1*.*

Solution: Let a_n be the number of ways of building a tower of height n. Suppose the tower is only 1 block high. The only possible way of building such a tower is by using a block of size 1. Thus $a_1 = 1$, since we have only *one* way of building the tower. If the tower were two blocks high, then we could have used *two* blocks of size 1, or *one* block of size 2. Thus there are two options, and $a_2 = 2$. In general, if we have constructed a tower of height n, then the last block we put on was *either* a 1-block *or* a 2-block. In the first case, the remaining tower below has height $n - 1$, and there are a_{n-1} ways of building that tower. In the second case, the remaining tower below has height $n - 2$, and there are a_{n-2} ways of building that tower.



The *total* number of ways of creating the overall tower is then $a_{n-1} + a_{n-2}$. The recursive definition is:

$$a_1 = 1, a_2 = 2$$

$$a_n = a_{n-1} + a_{n-2}$$

Example 10. *Suppose that the number of fish caught by a fisherman in year n ($n \geq 2$) is exactly the average of the number of fish he caught the previous two years. Find a recurrence for the number of fish caught by the fisherman, if in year 1 he caught* 200 *and in year 2 he caught* 600*.*

Solution: Let a_n be the number of fish caught in year n. The initial terms are $a_1 = 200$ and $a_2 = 600$. The recurrence relation is based on the fact that the number of fish caught is the average of the previous two years. Thus for $n>2$:

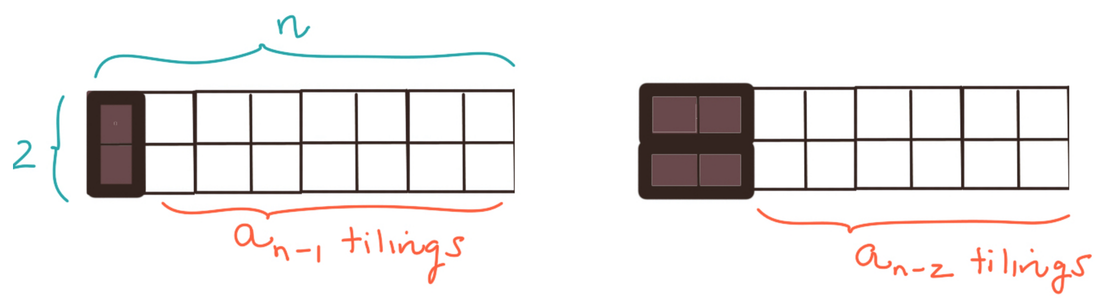
$$a_n = \frac{1}{2}(a_{n-1} + a_{n-2})$$

Example 11. *Find a recurrence for the number of ways of tiling a* $2 \times n$ *checkerboard using regular dominos of size* 1×2.

Solution: Let a_n be the number of ways to tile a $2 \times n$ checkerboard using dominos. The smallest checkerboard would have size 2×1. In that case, the only way to tile the board is by placing a domino vertically. Thus $a_1 = 1$. For a checkerboard of size 2×2, there are *two* ways of tiling that board (two vertically or two horizontally). So $a_2 = 2$.

For a checkerboard of size $2 \times n$, we have two cases for tiling the front section:

- **Case 1:** We could place a single domino vertically, in which case we would have a board of size $2 \times (n-1)$ left to tile. The number of ways to tile that board is a_{n-1}.
- **Case 2:** We could place *two* dominos horizontally, and have a board of size $2 \times (n-2)$ left to tile. The number of ways to tile the remaining board is a_{n-2}.



The total number of ways to tile the board is the sum of the possibilities from each case: $a_{n-1} + a_{n-2}$. Thus the recursive definition is:

$$a_1 = 1, a_2 = 2$$

$$a_n = a_{n-1} + a_{n-2} \text{ for } n > 2$$

In the next section, we look at one of the most well-known recurrence relations, first discovered in the 13th century, called the Fibonacci Numbers.