LESSON 9
Recurrence Relations

(§5.3 pp 344–347 top only)
Plus the following passages
from the book
Discrete Mathematics
by Dossey, Otto, Spence, & Vanden Eynden
Third edition, Addison-Wesley 1997.

8.1 ❖ RECURRENCE RELATIONS

An infinite ordered list is called a **sequence.** The individual items in the list are called **terms** of the sequence. For example,

$$0!, 1!, 2!, \ldots, n!, \ldots$$

is a sequence with first term 0!, second term 1!, and so forth. In this case the nth term of the sequence is defined explicitly as a function of n, namely $(n - 1)!$.

In this chapter we will study sequences where a general term is defined as a function of preceding terms. An equation relating a general term to terms that precede it is called a **recurrence relation.** In Section 2.6 we saw that $n!$ could be defined recursively by specifying that

$$0! = 1 \quad \text{and} \quad n! = n(n-1)! \quad \text{for } n \geq 1.$$

9-2

In this definition the equation

$$n! = n(n-1)! \qquad \text{for } n \geq 1$$

is a recurrence relation. It defines each term of the sequence of factorials as a function of the immediately preceding term.

In order to determine the values of the terms in a recursively defined sequence, we must know the values of a specific set of terms in the sequence, usually the beginning terms. The assignment of values for these terms gives a set of **initial conditions** for the sequence. In the case of the factorials, there is a single initial condition, which is that $0! = 1$. Knowing this value, we can then compute values for the other terms in the sequence from the recurrence relation. For example,

$$\begin{aligned} 1! &= 1(0!) = 1 \cdot 1 = 1, \\ 2! &= 2(1!) = 2 \cdot 1 = 2, \\ 3! &= 3(2!) = 3 \cdot 2 = 6, \\ 4! &= 4(3!) = 4 \cdot 6 = 24, \end{aligned}$$

and so on.

Another example of a sequence that is defined by a recurrence relation is the sequence of Fibonacci numbers. Recall from Section 2.6 that the Fibonacci numbers satisfy the recurrence relation

$$F_n = F_{n-1} + F_{n-2} \qquad \text{for } n \geq 3.$$

Because F_n is defined as a function of the two preceding terms, we must know two consecutive terms of the sequence in order to compute subsequent ones. For the Fibonacci numbers the initial conditions are $F_1 = 1$ and $F_2 = 1$. Note that there are sequences other than the Fibonacci numbers that satisfy the same recurrence relation, for example,

$$3, 4, 7, 11, 18, 29, 47, 76, \ldots.$$

Here each term after the second is the sum of the two preceding terms, and so the sequence is completely determined by the initial conditions $s_1 = 3$ and $s_2 = 4$.

In this section we will examine other situations in which recurrence relations occur and illustrate how they can be used to solve problems involving counting.

Example 8.1 Let us consider from a recursive point of view the question of determining the number of edges e_n in the complete graph $\mathcal{K}_n$ with n vertices. We begin by considering how many new edges need to be drawn to obtain $\mathcal{K}_n$ from $\mathcal{K}_{n-1}$. The addition of one new vertex requires the addition of $n - 1$ new edges, one to each

9-3

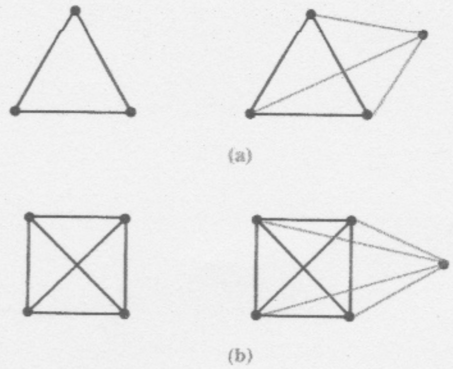


FIGURE 8.1

of the vertices in $\mathcal{K}_{n-1}$. (See Figure 8.1(a) for the case $n = 4$ and Figure 8.1(b) for the case $n = 5$.) Thus we see that the number of edges in $\mathcal{K}_n$ satisfies the recurrence relation

$$e_n = e_{n-1} + (n - 1) \qquad \text{for } n \geq 2.$$

In this equation the definition of e_n involves only the preceding term e_{n-1}, and so we need only one value of e_n to use the recurrence relation. Since the complete graph with 1 vertex has no edges, we see that $e_1 = 0$. This is the initial condition for this sequence. ❖

❖ **Example 8.2** The **Towers of Hanoi** game is played with a set of disks of graduated size with holes in their centers and a playing board having three spokes for holding the disks. (See Figure 8.2.) The object of the game is to transfer all the disks from spoke A to spoke C by moving one disk at a time without placing a larger disk on top of a smaller one. What is the minimum number of moves required when there are n disks?

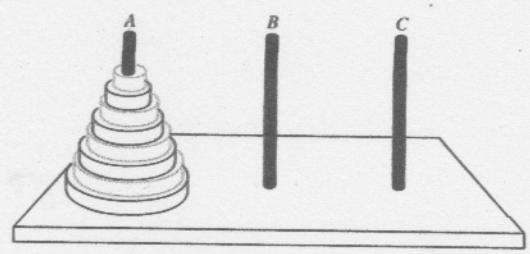
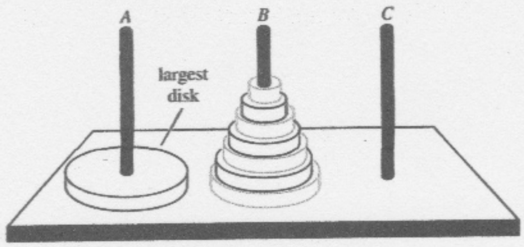


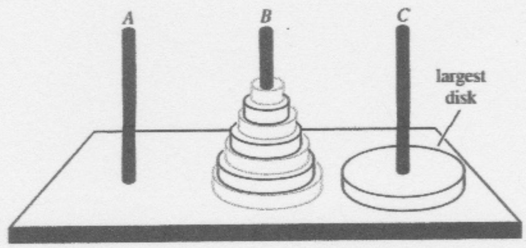
FIGURE 8.2

To answer the question, we will formulate a recurrence relation for m_n, the minimum number of moves to transfer n disks from one spoke to another. This will require expressing m_n in terms of previous terms m_i. It is easy to see that the most efficient procedure for winning the game with $n \geq 2$ disks is as follows. (See Figure 8.3.)

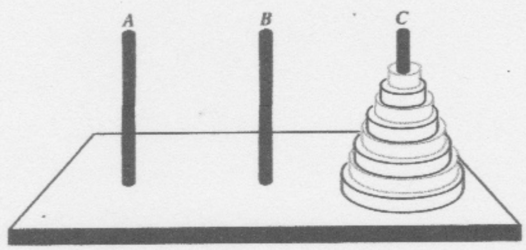
9-4



Step 1 Position after moving $n - 1$ disks from A to B



Step 2 Position after moving the bottom disk from A to C



Step 3 Position after moving $n - 1$ disks from B to C

FIGURE 8.3

(1) Move the smallest $n - 1$ disks (in accordance with the rules) as efficiently as possible from spoke A to spoke B.
(2) Move the largest disk from spoke A to spoke C.
(3) Move the smallest $n - 1$ disks (in accordance with the rules) as efficiently as possible from spoke B to spoke C.

Since step 1 requires moving $n - 1$ disks from one spoke to another, the minimum number of moves required in step 1 is just m_{n-1}. It then takes one move to accomplish step 2, and another m_{n-1} moves to accomplish step 3. This analysis produces the recurrence relation

$$m_n = m_{n-1} + 1 + m_{n-1},$$

9-5

which simplifies to the form

$$m_n = 2m_{n-1} + 1 \qquad \text{for } n \geq 2.$$

Again we need to know one value of m_n in order to use this recurrence relation. Because only one move is required to win a game with 1 disk, the initial condition for this sequence is $m_1 = 1$. By using the recurrence relation and the initial condition we can determine the number of moves required for any desired number of disks. For example,

$$\begin{aligned} m_1 &= 1, \\ m_2 &= 2(1) + 1 = 3, \\ m_3 &= 2(3) + 1 = 7, \\ m_4 &= 2(7) + 1 = 15, \qquad \text{and} \\ m_5 &= 2(15) + 1 = 31. \end{aligned}$$

In Section 8.2 we will obtain an explicit formula that expresses m_n in terms of n. ❖

9-6

STACKS: OPTIONAL EXAMPLE FOR CS STUDENTS

Example 8.5 A *stack* is an important data structure in computer science. It stores data subject to restriction that all insertions and deletions take place at one end of the stack (called the *top*). As a consequence of this restriction, the last item inserted into the stack must be the first item deleted, and so a stack is an example of a last-in-first-out structure.

We will insert all of the integers $1, 2, \ldots, n$ into a stack (in sequence) and count the possible sequences in which they can leave the stack. Note that each integer from 1 through n enters and leaves the stack exactly once. We will denote that integer k enters the stack by writing k and denote that integer k leaves the stack by writing $\overline{k}$. If $n = 1$, there is only one possible sequence, namely $1, \overline{1}$. For $n = 2$ there are two possibilities.

		Order leaving the stack
1.	$1, 2, \overline{2}, \overline{1}$	2, 1
2.	$1, \overline{1}, 2, \overline{2}$	1, 2

Thus if $n = 2$, there are two possible sequences in which the integers 1, 2 can leave a stack. Now consider the case $n = 3$. There are only five possibilities for inserting the integers 1, 2, 3 into a stack and deleting them from it.

		Order leaving the stack
1.	$1, 2, 3, \overline{3}, \overline{2}, \overline{1}$	3, 2, 1
2.	$1, 2, \overline{2}, 3, \overline{3}, \overline{1}$	2, 3, 1
3.	$1, 2, \overline{2}, \overline{1}, 3, \overline{3}$	2, 1, 3
4.	$1, \overline{1}, 2, 3, \overline{3}, \overline{2}$	1, 3, 2
5.	$1, \overline{1}, 2, \overline{2}, 3, \overline{3}$	1, 2, 3

Thus of the six possible permutations of the integers 1, 2, 3, only five can result from the insertion and deletion of 1, 2, 3 using a stack.

We will count the number c_n of permutations of $1, 2, \ldots, n$ that can result from the use of a stack in this manner. (Thus c_n is just the number of different ways that the integers 1 through n can leave a stack if they enter it in sequence.) The preceding paragraph shows that

$$c_1 = 1, \qquad c_2 = 2, \qquad c_3 = 5.$$

It is convenient also to define $c_0 = 1$. For an arbitrary positive integer n, we consider when the integer 1 is deleted from the stack. If it is the first integer deleted from the stack, then the sequence of operations begins:

$$1, \overline{1}, 2, \ldots.$$

The number of permutations that can result from such a sequence of operations is just the number of possible ways that $2, 3, \ldots, n$ can leave a stack if they enter it in sequence. This number is $c_{n-1} = c_0 c_{n-1}$.

9-7

If 1 is the second integer deleted from the stack, then the first integer deleted from the stack must be 2. Thus the sequence of operations must begin:

$$1, 2, \overline{2}, \overline{1}, 3, \ldots .$$

The number of permutations that can result from such a sequence of operations is $c_1 c_{n-2}$.

If 1 is the third integer deleted from the stack, then the first two integers deleted from the stack must be 2 and 3. Thus 1 must enter the stack, 2 and 3 must enter and leave the stack in some sequence, then 1 must leave, and finally 4, 5, . . . , n must enter and leave the stack in some sequence. The number of permutations that can result from such a sequence of operations is $c_2 c_{n-3}$.

In general, suppose that 1 is the kth integer deleted from the stack. Then the $k - 1$ integers 2, 3, . . . , k must enter and leave the stack in some sequence before deleting integer 1, and the $n - k$ integers $k + 1, k + 2, \ldots, n$ must enter and leave the stack in some sequence after deleting integer 1. The multiplication principle shows that the number of ways to perform these two operations is $c_{k-1}c_{n-k}$. Thus the addition principle gives

$$c_n = c_0 c_{n-1} + c_1 c_{n-2} + \cdots + c_{n-1} c_0 \qquad \text{for } n \geq 1.$$

Since we know that $c_0 = 1$, the recurrence relation above can be used to compute subsequent values of the sequence. For example,

$$\begin{aligned}
c_1 &= c_0 c_0 = 1 \cdot 1 = 1, \\
c_2 &= c_0 c_1 + c_1 c_0 = 1 \cdot 1 + 1 \cdot 1 = 2, \\
c_3 &= c_0 c_2 + c_1 c_1 + c_2 c_0 = 1 \cdot 2 + 1 \cdot 1 + 2 \cdot 1 = 5, \\
c_4 &= c_0 c_3 + c_1 c_2 + c_2 c_1 + c_3 c_0 = 1 \cdot 5 + 1 \cdot 2 + 2 \cdot 1 + 5 \cdot 1 = 14, \\
c_5 &= c_0 c_4 + c_1 c_3 + c_2 c_2 + c_3 c_1 + c_4 c_0 \\
&= 1 \cdot 14 + 1 \cdot 5 + 2 \cdot 2 + 5 \cdot 1 + 14 \cdot 1 = 42,
\end{aligned}$$

and so forth.

The numbers c_n are called **Catalan numbers** after Eugène Charles Catalan (1814–1894), who showed that they represent the number of ways in which n pairs of parentheses can be inserted into the expression

$$x_1 x_2 \ldots x_{n+1}$$

to group the factors into n products of pairs of numbers. For example, the $c_3 = 5$ different groupings of $x_1x_2x_3x_4$ into three products of pairs of numbers are

$$((x_1x_2)x_3)x_4, \quad x_1(x_2(x_3x_4)), \quad (x_1(x_2x_3))x_4,$$
$$x_1((x_2x_3)x_4), \quad \text{and} \quad (x_1x_2)(x_3x_4).$$

The Catalan numbers occur in several basic problems of computer science. ❖

End of optional example.

The preceding examples have shown several situations in which recurrence relations arise in dealing with counting. Recurrence relations are also invaluable tools in examining change over time in discrete settings, as shown by the next example.

Example 8.6 A grain elevator company receives 200 tons of corn per week from farmers once harvest starts. The elevator operators plan to ship out 30% of the corn on hand each week once the harvest season begins. If the company has 600 tons of corn on hand at the beginning of harvest, what recurrence relation describes the amount of corn on hand at the end of each week throughout the harvest season?

If g_n represents the number of tons of corn on hand at the end of week n of the harvest season, we can express the situation described in the preceding paragraph by the recurrence relation

$$g_n = g_{n-1} - 0.30g_{n-1} + 200 \qquad \text{for } n \geq 1$$

with the initial condition $g_0 = 600$, that is,

$$g_n = 0.70g_{n-1} + 200 \qquad \text{for } n \geq 1 \qquad \text{and} \qquad g_0 = 600.$$

The 0.70 coefficient of g_{n-1} reflects that 70% of the corn on hand is not shipped during the week, and the constant term 200 represents the amount of new corn brought to the elevator within the week. ❖

Recurrence relations are also often used to study the current or projected status of financial accounts.

Example 8.7 The Thompsons are purchasing a new house costing \$200,000 with a down payment of \$25,000 and a 30-year mortgage. Interest on the unpaid balance of the mortgage is to be compounded at the monthly rate of 1%, and the monthly payments will be \$1800. How much will the Thompsons owe after n months of payments?

Let b_n denote the balance (in dollars) that will be owed on the mortgage after n months of payments. We will obtain a recurrence relation expressing b_n in terms of previous balances. Note that the balance owed after n months will equal the balance owed after $n - 1$ months plus the monthly interest minus one monthly payment. Symbolically, we have

$$b_n = b_{n-1} + .01b_{n-1} - 1800,$$

which simplifies to the form

$$b_n = 1.01b_{n-1} - 1800 \qquad \text{for } n \geq 1.$$

Since this equation expresses b_n in terms of b_{n-1} only, we need just one term to use this recurrence relation. Now the amount owed initially is the purchase price minus the down payment, and so the initial condition is $b_0 = 175{,}000$. ❖

Recurrence relations, when applied to the study of change as shown in Examples 8.6 and 8.7, are sometimes referred to as **discrete dynamical systems.** They are the discrete analogs of the differential equations used to study change in continuous settings.

8.2 ❖ THE METHOD OF ITERATION

In Example 8.2 we saw that the minimum number of moves required to shift n disks from one spoke to another in the Towers of Hanoi game satisfies the recurrence relation

$$m_n = 2m_{n-1} + 1 \qquad \text{for } n \geq 2$$

and the initial condition $m_1 = 1$. From this information we can determine the value of m_n for any positive integer n. For example, the first few terms of the sequence defined by these conditions are:

$$\begin{aligned}
m_1 &= 1, \\
m_2 &= 2(1) + 1 = 2 + 1 = 3, \\
m_3 &= 2(3) + 1 = 6 + 1 = 7, \\
m_4 &= 2(7) + 1 = 14 + 1 = 15, \quad \text{and} \\
m_5 &= 2(15) + 1 = 30 + 1 = 31.
\end{aligned}$$

We can continue evaluating terms of the sequence in this manner, and so we can eventually determine the value of any particular term. This process can be quite tedious, however, if we need to evaluate m_n when n is large. In Example 8.7, for instance, we might need to know the unpaid balance of the mortgage after 20 years (240 months), which would require us to evaluate b_{240}. Although straightforward, this calculation would be quite time-consuming if we were evaluating the terms by hand in this manner.

We see, therefore, that it is often convenient to have a formula for computing the general term of a sequence defined by a recurrence relation without needing to calculate all of the preceding terms. A simple method that can be used to try to find such a formula is to start with the initial conditions and compute successive terms of the sequence, as illustrated above. If a pattern can be found, we can then guess an explicit formula for the general term and try to prove it by mathematical induction. This procedure is called the **method of iteration.**

We will use the method of iteration to find an explicit formula for the general term of the sequence satisfying the Towers of Hanoi recurrence

$$m_n = 2m_{n-1} + 1 \qquad \text{for } n \geq 2$$

with the initial condition $m_1 = 1$. We computed above the first few terms of the sequence satisfying these conditions. Although it is possible to see a pattern developing from these computations, it is helpful to repeat these calculations *without* simplifying the results to a numerical value.

$$\begin{aligned}
m_1 &= 1 \\
m_2 &= 2(1) + 1 = 2 + 1 \\
m_3 &= 2(2 + 1) + 1 = 2^2 + 2 + 1 \\
m_4 &= 2(2^2 + 2 + 1) + 1 = 2^3 + 2^2 + 2 + 1 \\
m_5 &= 2(2^3 + 2^2 + 2 + 1) + 1 = 2^4 + 2^3 + 2^2 + 2 + 1 \\
&\vdots
\end{aligned}$$

From these calculations we can guess an explicit formula for m_n:

$$m_n = 2^{n-1} + 2^{n-2} + \cdots + 2^2 + 2 + 1.$$

By using a familiar algebraic identity (see Example 2.57)

$$1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1},$$

this formula can be expressed in an even more compact manner:

$$m_n = \frac{2^n - 1}{2 - 1} = 2^n - 1.$$

At this point, the formula obtained above is nothing more than an educated guess. To verify that it does indeed give the correct values for m_n, we must prove by induction that the formula is correct. To do so, we must show that if a sequence $m_1, m_2, m_3, \ldots$ satisfies the recurrence relation

$$m_n = 2m_{n-1} + 1 \qquad \text{for } n \geq 2$$

and the initial condition $m_1 = 1$, then $m_n = 2^n - 1$ for all positive integers n. Clearly the formula is correct for $n = 1$ because

$$2^1 - 1 = 2 - 1 = 1 = m_1.$$

Now we assume that the formula is correct for some nonnegative integer k, that is, we assume that

$$m_k = 2^k - 1.$$

It remains to show that the formula is correct for $k + 1$. From the recurrence relation we know that

$$m_{k+1} = 2m_k + 1.$$

Hence

$$\begin{aligned} m_{k+1} &= 2(2^k - 1) + 1 \\ &= 2^{k+1} - 2 + 1 \\ &= 2^{k+1} - 1, \end{aligned}$$

which proves the formula for $k + 1$. It follows from the principle of mathematical induction that the formula

$$m_n = 2^n - 1$$

is correct for all positive integers n.

Certain formulas are very useful for simplifying the algebraic expressions that arise when using the method of iteration. One of these is the identity

$$1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}$$

from Example 2.57. Another is the formula for the sum of the first n positive integers

$$1 + 2 + 3 + \cdots + n = \frac{n(n + 1)}{2},$$

which was proved in Exercise 11 of Section 2.6.

Example 8.8 In Example 8.1 we saw that the number e_n of edges in the complete graph $\mathcal{K}_n$ satisfies the recurrence relation

$$e_n = e_{n-1} + (n - 1) \qquad \text{for } n \geq 2$$

and the initial condition $e_1 = 0$. We will use the method of iteration to obtain a formula for e_n. To begin, we use the recurrence relation to compute several terms of the sequence.

9-11

$$\begin{aligned} e_1 &= 0 \\ e_2 &= 0 + 1 \\ e_3 &= (0 + 1) + 2 \\ e_4 &= (0 + 1 + 2) + 3 \\ e_5 &= (0 + 1 + 2 + 3) + 4 \\ &\vdots \end{aligned}$$

From these calculations we conjecture that

$$\begin{aligned} e_n &= 0 + 1 + 2 + \cdots + (n-1) \\ &= \frac{(n-1)n}{2} \\ &= \frac{n^2 - n}{2}. \end{aligned}$$

To verify that the formula is correct, we again need a proof by induction to show that the terms of a sequence that satisfies the recurrence relation

$$e_n = e_{n-1} + (n-1) \qquad \text{for } n \geq 2$$

and the initial condition $e_1 = 0$ are given by the formula

$$e_n = \frac{n^2 - n}{2}.$$

The formula is correct for $n = 1$ because

$$\frac{n^2 - n}{2} = \frac{1^2 - 1}{2} = 0 = e_1.$$

Assume that

$$e_k = \frac{k^2 - k}{2}$$

for some $k \geq 1$. Then

$$\begin{aligned} e_{k+1} &= e_k + [(k+1) - 1] \\ &= \frac{k^2 - k}{2} + k \\ &= \frac{k^2 - k}{2} + \frac{2k}{2} \\ &= \frac{(k^2 + 2k + 1) - (k+1)}{2} \\ &= \frac{(k+1)^2 - (k+1)}{2}. \end{aligned}$$

Thus the formula is correct for $k + 1$. It now follows from the principle of mathematical induction that the formula is correct for all positive integers n. ❖

9-12

8.3 LINEAR DIFFERENCE EQUATIONS WITH CONSTANT COEFFICIENTS

The simplest type of recurrence relation gives s_n as a function of s_{n-1} for $n \geq 1$. We call an equation of the form

$$s_n = as_{n-1} + b,$$

where a and b are constants and $a \neq 0$, a **first-order linear difference equation with constant coefficients.** For example, the recurrence relations below are all first-order linear difference equations with constant coefficients:

$$s_n = 3s_{n-1} - 1, \qquad s_n = s_{n-1} + 7, \qquad \text{and} \qquad s_n = 5s_{n-1}.$$

Recurrence relations of this type occur frequently in applications, especially in the analysis of financial transactions. The recurrence relations in Examples 8.2 and 8.7 are first-order linear difference equations with constant coefficients.

Since a first-order linear difference equation with constant coefficients expresses s_n in terms of s_{n-1}, a sequence defined by such a difference equation is completely determined if a single term is known. We will use the method of iteration to find an explicit formula giving s_n as a function of n and s_0.

Consider the first-order linear difference equation with constant coefficients $s_n = as_{n-1} + b$ that has first term s_0. The first few terms of the sequence defined by this equation are

$$\begin{aligned}
s_0 &= s_0,\\
s_1 &= as_0 + b,\\
s_2 &= as_1 + b = a(as_0 + b) + b = a^2s_0 + ab + b,\\
s_3 &= as_2 + b = a(a^2s_0 + ab + b) + b = a^3s_0 + a^2b + ab + b,\\
s_4 &= as_3 + b = a(a^3s_0 + a^2b + ab + b) + b\\
&= a^4s_0 + a^3b + a^2b + ab + b.
\end{aligned}$$

It appears that

$$\begin{aligned}
s_n &= a^ns_0 + a^{n-1}b + a^{n-2}b + \cdots + a^2b + ab + b\\
&= a^ns_0 + (a^{n-1} + a^{n-2} + \cdots + a^2 + a + 1)b.
\end{aligned}$$

If $a = 1$, the expression in parentheses equals n; otherwise it can be simplified by using the identify from Example 2.57:

$$1 + x + x^2 + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1}.$$

Substituting in the expression above for s_n gives

$$\begin{aligned}
s_n &= a^ns_0 + \left(\frac{a^n - 1}{a - 1}\right)b\\
&= a^ns_0 + a^n\left(\frac{b}{a - 1}\right) - \left(\frac{b}{a - 1}\right)\\
&= a^n(s_0 + c) - c,
\end{aligned}$$

9-13

where

$$c = \frac{b}{a-1}.$$

We will state this result as Theorem 8.1, leaving a formal proof by mathematical induction as an exercise.

Theorem 8.1 The general term of the first-order linear difference equation with constant coefficients $s_n = as_{n-1} + b$ that has initial value s_0 satisfies

$$s_n = \begin{cases} a^n(s_0 + c) - c & \text{if } a \neq 1 \\ s_0 + nb & \text{if } a = 1, \end{cases}$$

where

$$c = \frac{b}{a-1}.$$

Example 8.12 Find a formula for s_n if s_n $3s_{n-1} - 1$ for $n \geq 1$ and $s_0 = 2$. Here $a = 3$ and $b = -1$ in the notation of Theorem 8.1. Thus

$$\epsilon = \frac{b}{a-1} = \frac{-1}{3-1} = \frac{-1}{2},$$

and so

$$\begin{aligned} s_n &= a^n(s_0 + c) - c \\ &= 3^n\left[2 + \frac{-1}{2}\right] - \frac{-1}{2} \\ &= \frac{3}{2}\left(3^n\right) + \frac{1}{2} \\ &= \frac{1}{2}\left(3^{n+1} + 1\right). \end{aligned}$$

Substituting $n = 0, 1, 2, 3, 4$, and 5 into this formula gives

$$s_0 = 2,\ s_1 = 5,\ s_2 = 14,\ s_3 = 41,\ s_4 = 122, \text{ and } s_5 = 365,$$

which are easily checked by using the recurrence relation

$$s_n = 3s_{n-1} - 1 \qquad \text{for } n \geq 1$$

and the initial condition $s_0 = 2$.

Example 8.13 In Example 8.7 find a formula for b_n, the unpaid balance of the Thompson's mortgage after n months.

We saw in Example 8.7 that b_n satisfies the recurrence relation

$$b_n = 1.01b_{n-1} - 1800 \qquad \text{for } n \geq 1$$

9-14

and the initial condition $b_0 = 175{,}000$. Since this recurrence relation is a first-order linear difference equation with constant coefficients, Theorem 8.1 can be used to find a formula expressing b_n as a function of n and b_0. In the notation of Theorem 8.1 we have $a = 1.01$ and $b = -1800$. Hence

$$c = \frac{b}{a-1} = \frac{-1800}{1.01 - 1} = \frac{-1800}{0.01} = -180{,}000.$$

Thus the desired formula for b_n is

$$\begin{aligned} b_n &= a^n(b_0 + c) - c \\ &= (1.01)^n\,[175{,}000 + (-180{,}000)] - (-180{,}000) \\ &= -5000(1.01)^n + 180{,}000. \end{aligned}$$

For example, the balance of the loan after 20 years (240 months) of payments is

$$\begin{aligned} b_{240} &= -5000(1.01)^{240} + 180{,}000 \approx -54{,}462.77 + 180{,}000 \\ &= 125{,}537.23. \end{aligned}$$

Thus the Thompsons will still owe \$125,537.23 after 20 years. ❖

❖ **Example 8.14** A lumber company owns 7000 birch trees. Each year the company plans to harvest 12% of its trees and plant 600 new ones.

(a) How many trees will there be after 10 years?

(b) How many trees will there be in the long run?

Let s_n denote the number of trees after n years. During year n, 12% of the trees existing in year $n - 1$ will be harvested; the number is $0.12s_{n-1}$. Since 600 additional trees will be planted during year n, the number of trees after n years is described by the equation

$$s_n = s_{n-1} - 0.12s_{n-1} + 600,$$

that is,

$$s_n = 0.88s_{n-1} + 600.$$

This is a first-order linear difference equation with constant coefficients $a = 0.88$ and $b = 600$. We are interested in the solution of this equation satisfying the initial condition $s_0 = 7000$. In the notation of Theorem 8.1

$$c = \frac{b}{a-1} = \frac{600}{0.88 - 1} = \frac{600}{-0.12} = -5000.$$

Hence a formula expressing s_n in terms of n is

$$\begin{aligned} s_n &= a^n(s_0 + c) - c \\ &= (0.88)^n(7000 - 5000) + 5000 \\ &= 2000(0.88)^n + 5000. \end{aligned}$$

(a) Therefore after 10 years the number of trees will be

$$s_{10} = 2000(0.88)^{10} + 5000 \approx 5557.$$

(b) As n increases, the quantity $(0.88)^n$ decreases to zero. Hence the formula

$$s_n = 2000(0.88)^n + 5000$$

implies that the number of trees approaches 5000. (Note that as the number of trees approaches 5000, the number of trees being harvested each year approaches the number of new trees being planted.) ❖