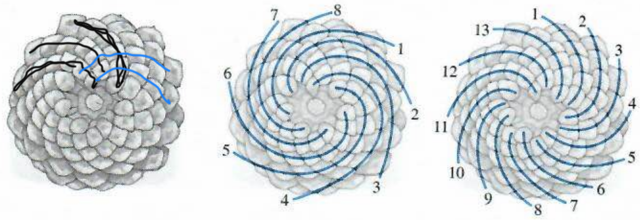
Other mathematical properties of the Fibonacci sequence are given in Example 3 and in the exercises at the end of this section. But it's not only mathematicians who are interested in the Fibonacci sequence. Fibonacci numbers often occur in nature. The number of petals on a daisy is often a Fibonacci number. Viewing a pine cone from its base, the seeds appear to be arranged in clockwise and counterclockwise spirals. Counting the number of each kind of spiral often gives two consecutive Fibonacci numbers (here 8 and 13). The same is true for seeds in flowers such as sunflowers, or for spirals on pineapples.



And in the worlds of art and architecture, the golden ratio is thought to create aesthetically pleasing proportions. The *golden ratio* is

$$\boxed{\frac{1+\sqrt{5}}{2} \approx 1.6180339}$$

and is the value approached by the ratio of two consecutive Fibonacci numbers $F(n+1)/F(n)$ for larger and larger values of n.

EXAMPLE 3 Prove that in the Fibonacci sequence

$$F(n+4) = 3F(n+2) - F(n) \text{ for all } n \geq 1$$

Because we want to prove something true for all $n \geq 1$, it is natural to think of a proof by induction. And because the value of $F(n)$ depends on both $F(n-1)$ and $F(n-2)$, the second principle of induction should be used. For the basis step of the inductive proof, we'll prove two cases, $n = 1$ and $n = 2$. For $n = 1$, by substituting 1 for n in the equation we want to prove, we get

$$F(5) = 3F(3) - F(1)$$

or (using values computed in Practice 2)

$$5 = 3(2) - 1$$

which is true. For $n = 2$,

$$F(6) = 3F(4) - F(2)$$

or

$$8 = 3(3) - 1$$

which is also true. Assume that for all r, $1 \leq r \leq k$,

$$F(r+4) = 3F(r+2) - F(r).$$

Several base cases required

Now show the case for $k + 1$, where $k + 1 \geq 3$. (We've already proved the case for $n = 1$ and the case for $n = 2$.) Thus we want to show

$$F(k + 1 + 4) \stackrel{?}{=} 3F(k + 1 + 2) - F(k + 1)$$

or

$$F(k + 5) \stackrel{?}{=} 3F(k + 3) - F(k + 1)$$

From the recurrence relation for the Fibonacci sequence, we have

$$F(k + 5) = F(k + 3) + F(k + 4) \qquad \text{(}F\text{ at any value is the sum of }F\text{ at the two previous values)}$$

and by the inductive hypothesis, with $r = k - 1$ and $r = k$, respectively,

$$F(k + 3) = 3F(k + 1) - F(k - 1)$$

and

$$F(k + 4) = 3F(k + 2) - F(k)$$

Therefore

$$\begin{aligned} F(k + 5) &= F(k + 3) + F(k + 4) \\ &= [3F(k + 1) - F(k - 1)] + [3F(k + 2) - F(k)] \\ &= 3[F(k + 1) + F(k + 2)] - [F(k - 1) + F(k)] \\ &= 3F(k + 3) - F(k + 1) \qquad \text{(using the recurrence relation again)} \end{aligned}$$

This completes the inductive proof. ✓

PRACTICE 3 In the inductive proof of Example 3, why is it necessary to prove $n = 2$ as a special case?

EXAMPLE 4 The formula

$$F(n + 4) = 3F(n + 2) - F(n) \text{ for all } n \geq 1$$

of Example 3 can also be proved without induction, using just the recurrence relation from the definition of Fibonacci numbers. The recurrence relation

$$F(n + 2) = F(n) + F(n + 1)$$

can be rewritten as

$$F(n + 1) = F(n + 2) - F(n) \tag{1}$$

Then

$$
\begin{aligned}
F(n+4) &= F(n+3) + F(n+2) \\
&= F(n+2) + F(n+1) + F(n+2) && \text{(rewriting } F(n+3)\text{)} \\
&= F(n+2) + [F(n+2) - F(n)] + F(n+2) && \text{(rewriting } F(n+1) \text{ using (1))} \\
&= 3F(n+2) - F(n)
\end{aligned}
$$

Recursively Defined Sets

The objects in a sequence are ordered—there is a first object, a second object, and so on. A set of objects is a collection of objects on which no ordering is imposed. Some sets can be defined recursively.

EXAMPLE 5 In Section 1.1 we noted that certain strings of statement letters, logical connectives, and parentheses, such as $(A \wedge B)' \vee C$, are considered legitimate, while other strings, such as $\wedge \wedge A''B$, are not legitimate. The syntax for arranging such symbols constitutes the definition of the set of propositional well-formed formulas, and it is a recursive definition.

1. Any statement letter is a wff.
2. If P and Q are wffs, so are $(P \wedge Q)$, $(P \vee Q)$, $(P \rightarrow Q)$, (P') and $(P \leftrightarrow Q)$.[2]

Using the rules of precedence for logical connectives established in Section 1.1, we can omit parentheses when doing so causes no confusion. Thus we write $(P \vee Q)$ as $P \vee Q$, or (P') as P'; the new expressions are technically not wffs by the definition just given, but they unambiguously represent wffs.

By beginning with statement letters and repeatedly using rule 2, any propositional wff can be built. For example, A, B, and C are all wffs by rule 1. By rule 2,

$$(A \wedge B) \text{ and } (C')$$

are both wffs. By rule 2 again,

$$((A \wedge B) \rightarrow (C'))$$

is a wff. Applying rule 2 yet again, we get the wff

$$(((A \wedge B) \rightarrow (C'))')$$

Eliminating some pairs of parentheses, we can write this wff as

$$((A \wedge B) \rightarrow C')'$$

[2]Sometimes there is a final rule added to the effect that there are no applicable rules besides those already given, which means that if something can't be generated using the rules already given, then it does not belong to the set being described. We'll assume that when we stop writing rules, there are no more applicable rules!

PRACTICE 4 Show how to build the wff $((A \vee (B')) \to C)$ from the definition in Example 5. ■

PRACTICE 5 A recursive definition for the set of people who are ancestors of James could have the following basis:

James's parents are ancestors of James.

Give the inductive step. ■

Strings of symbols drawn from a finite "alphabet" set are objects that are commonly encountered in computer science. Computers store data as **binary strings**, strings from the alphabet consisting of 0*s* and 1*s*; compilers view program statements as strings of *tokens*, such as key words and identifiers. The collection of all finite-length strings of symbols from an alphabet, usually called strings *over* an alphabet, can be defined recursively (see Example 6). Many sets of strings with special properties also have recursive definitions.

EXAMPLE 6 The set of all (finite-length) strings of symbols over a finite alphabet A is denoted by A^*. The recursive definition of A^* is

1. The **empty string** λ (the string with no symbols) belongs to A^*.
2. Any single member of A belongs to A^*.
3. If x and y are strings in A^*, so is xy, the **concatenation** of strings x and y.

Parts 1 and 2 constitute the basis, and part 3 is the recursive step of this definition. Note that for any string x, $x\lambda = \lambda x = x$. ●

PRACTICE 6 If $x = 1011$ and $y = 001$, write the strings xy, yx, and $yx\lambda x$. ■

1 Base cases; "0" & "1" & λ are palindromes,

PRACTICE 7 Give a recursive definition for the set of all binary strings that are **palindromes**, strings that read the same forward and backward. ■

A = {0,1} 2 Any string of the form xyx where x & y are palindromes.

EXAMPLE 7 Suppose that in a certain programming language, identifiers can be alphanumeric strings of arbitrary length but must begin with a letter. A recursive definition for the set of such strings is

1. A single letter is an identifier.
2. If A is an identifier, so is the concatenation of A and any letter or digit.

A more symbolic notation for describing sets of strings that are recursively defined is called **Backus–Naur form**, or **BNF**, originally developed to define the