SECTION 4.3 Recursive Definitions and Structural Induction

The best way to approach a recursive definition is first to compute several instances. For example, if you are given a recursive definition of a function f, then compute $f(0)$ through $f(8)$ to get a feeling for what is happening. Most of the time it is necessary to prove statements about recursively defined objects using structural induction (or mathematical induction or strong induction), and the induction practically takes care of itself, mimicking the recursive definition.

1. In each case, we compute $f(1)$ by using the recursive part of the definition with $n = 0$, together with the given fact that $f(0) = 1$. Then we compute $f(2)$ by using the recursive part of the definition with $n = 1$, together with the given value of $f(1)$. We continue in this way to obtain $f(3)$ and $f(4)$.
a) $f(1) = f(0)+2 = 1+2 = 3$; $f(2) = f(1)+2 = 3+2 = 5$; $f(3) = f(2)+2 = 5+2 = 7$; $f(4) = f(3)+2 = 7+2 = 9$
b) $f(1) = 3f(0) = 3 \cdot 1 = 3$; $f(2) = 3f(1) = 3 \cdot 3 = 9$; $f(3) = 3f(2) = 3 \cdot 9 = 27$; $f(4) = 3f(3) = 3 \cdot 27 = 81$
c) $f(1) = 2^{f(0)} = 2^1 = 2$; $f(2) = 2^{f(1)} = 2^2 = 4$; $f(3) = 2^{f(2)} = 2^4 = 16$; $f(4) = 2^{f(3)} = 2^{16} = 65{,}536$
d) $f(1) = f(0)^2+f(0)+1 = 1^2+1+1 = 3$; $f(2) = f(1)^2+f(1)+1 = 3^2+3+1 = 13$; $f(3) = f(2)^2+f(2)+1 = 13^2+13+1 = 183$; $f(4) = f(3)^2+f(3)+1 = 183^2+183+1 = 33{,}673$

3. In each case we compute the subsequent terms by plugging into the recursive formula, using the previously given or computed values.
a) $f(2) = f(1)+3f(0) = 2+3(-1) = -1$; $f(3) = f(2)+3f(1) = -1+3\cdot 2 = 5$; $f(4) = f(3)+3f(2) = 5+3(-1) = 2$; $f(5) = f(4)+3f(3) = 2+3\cdot 5 = 17$
b) $f(2) = f(1)^2 f(0) = 2^2 \cdot (-1) = -4$; $f(3) = f(2)^2 f(1) = (-4)^2 \cdot 2 = 32$; $f(4) = f(3)^2 f(2) = 32^2 \cdot (-4) = -4096$; $f(5) = f(4)^2 f(3) = (-4096)^2 \cdot 32 = 536{,}870{,}912$
c) $f(2) = 3f(1)^2 - 4f(0)^2 = 3 \cdot 2^2 - 4 \cdot (-1)^2 = 8$; $f(3) = 3f(2)^2 - 4f(1)^2 = 3 \cdot 8^2 - 4 \cdot 2^2 = 176$; $f(4) = 3f(3)^2 - 4f(2)^2 = 3 \cdot 176^2 - 4 \cdot 8^2 = 92{,}672$; $f(5) = 3f(4)^2 - 4f(3)^2 = 3 \cdot 92672^2 - 4 \cdot 176^2 = 25{,}764{,}174{,}848$
d) $f(2) = f(0)/f(1) = (-1)/2 = -1/2$; $f(3) = f(1)/f(2) = 2/(-\frac{1}{2}) = -4$; $f(4) = f(2)/f(3) = (-\frac{1}{2})/(-4) = 1/8$; $f(5) = f(3)/f(4) = (-4)/\frac{1}{8} = -32$

5. a) This is not valid, since letting $n = 1$ we would have $f(1) = 2f(-1)$, but $f(-1)$ is not defined.
b) This is valid. The basis step tells us what $f(0)$ is, and the recursive step tells us how each subsequent value is determined from the one before. It is not hard to look at the pattern and conjecture that $f(n) = 1-n$. We prove this by induction. The basis step is $f(0) = 1 = 1-0$; and if $f(k) = 1-k$, then $f(k+1) = f(k)-1 = 1-k-1 = 1-(k+1)$.
c) The basis conditions specify $f(0)$ and $f(1)$, and the recursive step gives $f(n)$ in terms of $f(n-1)$ for $n \geq 2$, so this is a valid definition. If we compute the first several values, we conjecture that $f(n) = 4-n$ if $n > 0$, but $f(0) = 2$. That is our "formula." To prove it correct by induction we need two basis steps: $f(0) = 2$, and $f(1) = 3 = 4-1$. For the inductive step (with $k \geq 1$), $f(k+1) = f(k) - 1 = (4-k)-1 = 4-(k+1)$.
d) The basis conditions specify $f(0)$ and $f(1)$, and the recursive step gives $f(n)$ in terms of $f(n-2)$ for $n \geq 2$, so this is a valid definition. The sequence of function values is 1, 2, 2, 4, 4, 8, 8, ..., and we can fit a formula to this if we use the floor function: $f(n) = 2^{\lfloor (n+1)/2 \rfloor}$. For a proof, we check the base cases: $f(0) = 1 = 2^{\lfloor (0+1)/2 \rfloor}$ and $f(1) = 2 = 2^{\lfloor (1+1)/2 \rfloor}$. For the inductive step: $f(k+1) = 2f(k-1) = 2 \cdot 2^{\lfloor k/2 \rfloor} = 2^{\lfloor k/2 \rfloor + 1} = 2^{\lfloor ((k+1)+1)/2 \rfloor}$.
e) The definition tells us explicitly what $f(0)$ is. The recursive step specifies $f(1)$, $f(3)$, ... in terms of $f(0)$, $f(2)$, ...; and it also gives $f(2)$, $f(4)$, ... in terms of $f(0)$, $f(2)$, ... So the definition is valid. We compute that $f(1) = 3$, $f(2) = 9$, $f(3) = 27$, and so conjecture that $f(n) = 3^n$. The basis step of the inductive proof is clear. For odd n greater than 0 we have $f(n) = 3f(n-1) = 3 \cdot 3^{n-1} = 3^n$, and for even n greater than 1 we have $f(n) = 9f(n-2) = 9 \cdot 3^{n-2} = 3^n$. Note that we used a slightly different notation here, letting n be the new value, rather than $k+1$, but the logic is the same.

7. There are many correct answers for these sequences. We will give what we consider to be the simplest ones.

a) Clearly each term in this sequence is 6 greater than the preceding term. Thus we can define the sequence by setting $a_1 = 6$ and declaring that $a_{n+1} = a_n + 6$ for all $n \geq 1$.

b) This is just like part (a), in that each term is 2 more than its predecessor. Thus we have $a_1 = 3$ and $a_{n+1} = a_n + 2$ for all $n \geq 1$.

c) Each term is 10 times its predecessor. Thus we have $a_1 = 10$ and $a_{n+1} = 10a_n$ for all $n \geq 1$.

d) Just set $a_1 = 5$ and declare that $a_{n+1} = a_n$ for all $n \geq 1$.

9. We need to write $F(n+1)$ in terms of $F(n)$. Since $F(n)$ is the sum of the first n positive integers (namely 1 through n), and $F(n+1)$ is the sum of the first $n+1$ positive integers (namely 1 through $n+1$), we can obtain $F(n+1)$ from $F(n)$ by adding $n+1$. Therefore the recursive part of the definition is $F(n+1) = F(n)+n+1$. The initial condition is a specification of the value of $F(0)$; the sum of no positive integers is clearly 0, so we set $F(0) = 0$. (Alternately, if we assume that the argument for F is intended to be strictly positive, then we set $F(1) = 1$, since the sum of the first one positive integer is 1.)

11. We need to see how $P_m(n+1)$ relates to $P_m(n)$. Now $P_m(n+1) = m(n+1) = mn + m = P_m(n) + m$. Thus the recursive part of our definition is just $P_m(n+1) = P_m(n) + m$. The basis step is $P_m(0) = 0$, since $m \cdot 0 = 0$, no matter what value m has.

13. We prove this using the principle of mathematical induction. The base case is $n = 1$, and in that case the statement to be proved is just $f_1 = f_2$; this is true since both values are 1. Next we assume the inductive hypothesis, that

$$f_1 + f_3 + \cdots + f_{2n-1} = f_{2n},$$

and try to prove the corresponding statement for $n+1$, namely,

$$f_1 + f_3 + \cdots + f_{2n-1} + f_{2n+1} = f_{2n+2}.$$

We have

$$\begin{aligned} f_1 + f_3 + \cdots + f_{2n-1} + f_{2n+1} &= f_{2n} + f_{2n+1} \quad \text{(by the inductive hypothesis)} \\ &= f_{2n+2} \quad \text{(by the definition of the Fibonacci numbers).} \end{aligned}$$

15. We prove this using the principle of mathematical induction. The basis step is for $n = 1$, and in that case the statement to be proved is just $f_0f_1 + f_1f_2 = f_2^2$; this is true since $0 \cdot 1 + 1 \cdot 1 = 1^2$. Next we assume the inductive hypothesis, that

$$f_0f_1 + f_1f_2 + \cdots + f_{2n-1}f_{2n} = f_{2n}^2,$$

and try to prove the corresponding statement for $n+1$, namely,

$$f_0f_1 + f_1f_2 + \cdots + f_{2n-1}f_{2n} + f_{2n}f_{2n+1} + f_{2n+1}f_{2n+2} = f_{2n+2}^2.$$

Note that *two* extra terms were added, since the final subscript has to be even. We have

$$\begin{aligned} f_0f_1 + f_1f_2 + \cdots + f_{2n-1}f_{2n} + f_{2n}f_{2n+1} + f_{2n+1}f_{2n+2} &= f_{2n}^2 + f_{2n}f_{2n+1} + f_{2n+1}f_{2n+2} \\ &\quad \text{(by the inductive hypothesis)} \\ &= f_{2n}(f_{2n} + f_{2n+1}) + f_{2n+1}f_{2n+2} \\ &\quad \text{(by factoring)} \\ &= f_{2n}f_{2n+2} + f_{2n+1}f_{2n+2} \\ &\quad \text{(by the definition of the Fibonacci numbers)} \\ &= (f_{2n} + f_{2n+1})f_{2n+2} \\ &= f_{2n+2}f_{2n+2} = f_{2n+2}^2. \end{aligned}$$

17. Let d_n be the number of divisions used by Algorithm 6 in Section 3.6 (the Euclidean algorithm) to find $\gcd(f_{n+1}, f_n)$. We write the calculation in this order, since $f_{n+1} \ge f_n$. We begin by finding the values of d_n for the first few values of n, in order to find a pattern and make a conjecture as to what the answer is. For $n = 0$ we are computing $\gcd(f_1, f_0) = \gcd(1, 0)$. Without performing any divisions, we know immediately that the answer is 1, so $d_0 = 0$. For $n = 1$ we are computing $\gcd(f_2, f_1) = \gcd(1, 1)$. One division is used to show that $\gcd(1, 1) = \gcd(1, 0)$, so $d_1 = 1$. For $n = 2$ we are computing $\gcd(f_3, f_2) = \gcd(2, 1)$. One division is used to show that $\gcd(2, 1) = \gcd(1, 0)$, so $d_2 = 1$. For $n = 3$, the computation gives successively $\gcd(f_4, f_3) = \gcd(3, 2) = \gcd(2, 1) = \gcd(1, 0)$, for a total of 2 divisions; thus $d_3 = 2$. For $n = 4$, we have $\gcd(f_5, f_4) = \gcd(5, 3) = \gcd(3, 2) = \gcd(2, 1) = \gcd(1, 0)$, for a total of 3 divisions; thus $d_4 = 3$. At this point we see that each increase of 1 in n seems to add one more division, in order to reduce $\gcd(f_{n+1}, f_n)$ to $\gcd(f_n, f_{n-1})$. Perhaps, then, for $n \ge 2$, we have $d_n = n - 1$. Let us make that conjecture. We have already verified the basis step when we computed that $d_2 = 1$. Now assume the inductive hypothesis, that $d_n = n - 1$. We must show that $d_{n+1} = n$. Now d_{n+1} is the number of divisions used in computing $\gcd(f_{n+2}, f_{n+1})$. The first step in the algorithm is to divide f_{n+1} into f_{n+2}. Since $f_{n+2} = f_{n+1} + f_n$ (this is the key point) and $f_n < f_{n+1}$, we get a quotient of 1 and a remainder of f_n. Thus we have, after one division, $\gcd(f_{n+2}, f_{n+1}) = \gcd(f_{n+1}, f_n)$. Now by the inductive hypothesis we need exactly $d_n = n - 1$ more divisions, since the algorithm proceeds from this point exactly as it proceeded given the inputs for the case of n. Therefore $1 + (n - 1) = n$ divisions are used in all, and our proof is complete. The answer, then, is that $d_0 = 0$, $d_1 = 1$, and $d_n = n - 1$ for $n \ge 2$. (If we interpreted the problem as insisting that we compute $\gcd(f_n, f_{n+1})$, with that order of the arguments, then the analysis and the answer are slightly different: $d_0 = 1$, and $d_n = n$ for $n \ge 1$.)

19. The determinant of the matrix $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$, written $|\mathbf{A}|$, is by definition $ad - bc$; and the determinant has the multiplicative property that $|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}|$. Therefore the determinant of the matrix $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ in Exercise 16 is $1 \cdot 0 - 1 \cdot 1 = -1$, and $|\mathbf{A}^n| = |\mathbf{A}|^n = (-1)^n$. On the other hand, the determinant of the matrix $\begin{bmatrix} f_{n+1} & f_n \\ f_n & f_{n-1} \end{bmatrix}$ is by definition $f_{n+1}f_{n-1} - f_n^2$. In Exercise 18 we showed that $\mathbf{A}^n$ is this latter matrix. The identity in Exercise 14 follows.

21. Assume that the definitions given in Exercise 20 were as follows: the max or min of one number is itself; $\max(a_1, a_2) = a_1$ if $a_1 \ge a_2$ and a_2 if $a_1 < a_2$, whereas $\min(a_1, a_2) = a_2$ if $a_1 \ge a_2$ and a_1 if $a_1 < a_2$; and for $n \ge 2$,

$$\max(a_1, a_2, \ldots, a_{n+1}) = \max(\max(a_1, a_2, \ldots, a_n), a_{n+1})$$

and

$$\min(a_1, a_2, \ldots, a_{n+1}) = \min(\min(a_1, a_2, \ldots, a_n), a_{n+1}).$$

We can then prove the three statements here by induction on n.

a) For $n = 1$, both sides of the equation equal $-a_1$. For $n = 2$, we must show that $\max(-a_1, -a_2) = -\min(a_1, a_2)$. There are two cases, depending on the relationship between a_1 and a_2. If $a_1 \le a_2$, then $-a_1 \ge -a_2$, so by our definition, $\max(-a_1, -a_2) = -a_1$. On the other hand our definition implies that $\min(a_1, a_2) = a_1$ in this case. Therefore $\max(-a_1, -a_2) = -a_1 = -\min(a_1, a_2)$. The other case, $a_1 > a_2$, is similar: $\max(-a_1, -a_2) = -a_2 = -\min(a_1, a_2)$. Now we are ready for the inductive step. Assume the inductive hypothesis, that

$$\max(-a_1, -a_2, \ldots, -a_n) = -\min(a_1, a_2, \ldots, a_n).$$

We need to show the corresponding equality for $n+1$. We have

$$\begin{aligned}
&\max(-a_1, -a_2, \ldots, -a_n, -a_{n+1}) \\
&\quad = \max(\max(-a_1, -a_2, \ldots, -a_n), -a_{n+1}) \quad \text{(by definition)} \\
&\quad = \max(-\min(a_1, a_2, \ldots, a_n), -a_{n+1}) \quad \text{(by the inductive hypothesis)} \\
&\quad = -\min(\min(a_1, a_2, \ldots, a_n), a_{n+1}) \quad \text{(by the already proved case } n = 2\text{)} \\
&\quad = -\min(a_1, a_2, \ldots, a_n, a_{n+1}) \quad \text{(by definition)}.
\end{aligned}$$

b) For $n = 1$, the equation is simply the identity $a_1 + b_1 = a_1 + b_1$. For $n = 2$, the situation is a little messy. Let us consider first the case that $a_1 + b_1 \geq a_2 + b_2$. Then $\max(a_1 + b_1, a_2 + b_2) = a_1 + b_1$. Also note that $a_1 \leq \max(a_1, b_1)$, and $b_1 \leq \max(b_1, b_2)$, so that $a_1 + b_1 \leq \max(a_1, a_2) + \max(b_1, b_2)$. Therefore we have $\max(a_1 + b_1, a_2 + b_2) = a_1 + b_1 \leq \max(a_1, a_2) + \max(b_1, b_2)$. The other case, in which $a_1 + b_1 < a_2 + b_2$, is similar. Now for the inductive step, we first need a lemma: if $u \leq v$, then $\max(u, w) \leq \max(v, w)$; this is easy to prove by looking at the three cases determined by the size of w relative to the sizes of u and v. Now assuming the inductive hypothesis, we have

$$\begin{aligned}
&\max(a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n, a_{n+1} + b_{n+1}) \\
&\quad = \max(\max(a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n), a_{n+1} + b_{n+1}) \quad \text{(by definition)} \\
&\quad \leq \max(\max(a_1, a_2, \ldots, a_n) + \max(b_1, b_2, \ldots, b_n), a_{n+1} + b_{n+1}) \\
&\qquad \text{(by the inductive hypothesis and the lemma)} \\
&\quad \leq \max(\max(a_1, a_2, \ldots, a_n), a_{n+1}) + \max(\max(b_1, b_2, \ldots, b_n), b_{n+1}) \\
&\qquad \text{(by the already proved case } n = 2\text{)} \\
&\quad = \max(a_1, a_2, \ldots, a_n, a_{n+1}) + \max(b_1, b_2, \ldots, b_n, b_{n+1}) \quad \text{(by definition)}.
\end{aligned}$$

c) The proof here is exactly dual to the proof in part **(b)**. We replace every occurrence of "max" by "min," and invert each inequality. The proof then reads as follows. For $n = 1$, the equation is simply the identity $a_1 + b_1 = a_1 + b_1$. For $n = 2$, the situation is a little messy. Let us consider first the case that $a_1 + b_1 \leq a_2 + b_2$. Then $\min(a_1 + b_1, a_2 + b_2) = a_1 + b_1$. Also note that $a_1 \geq \min(a_1, a_2)$, and $b_1 \geq \min(b_1, b_2)$, so that $a_1 + b_1 \geq \min(a_1, a_2) + \min(b_1, b_2)$. Therefore we have $\min(a_1 + b_1, a_2 + b_2) = a_1 + b_1 \geq \min(a_1, a_2) + \min(b_1, b_2)$. The other case, in which $a_1 + b_1 > a_2 + b_2$, is similar. Now for the inductive step, we first need a lemma: if $u \geq v$, then $\min(u, w) \geq \min(v, w)$; this is easy to prove by looking at the three cases determined by the size of w relative to the sizes of u and v. Now assuming the inductive hypothesis, we have

$$\begin{aligned}
&\min(a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n, a_{n+1} + b_{n+1}) \\
&\quad = \min(\min(a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n), a_{n+1} + b_{n+1}) \quad \text{(by definition)} \\
&\quad \geq \min(\min(a_1, a_2, \ldots, a_n) + \min(b_1, b_2, \ldots, b_n), a_{n+1} + b_{n+1}) \\
&\qquad \text{(by the inductive hypothesis and the lemma)} \\
&\quad \geq \min(\min(a_1, a_2, \ldots, a_n), a_{n+1}) + \min(\min(b_1, b_2, \ldots, b_n), b_{n+1}) \\
&\qquad \text{(by the already proved case } n = 2\text{)} \\
&\quad = \min(a_1, a_2, \ldots, a_n, a_{n+1}) + \min(b_1, b_2, \ldots, b_n, b_{n+1}) \quad \text{(by definition)}.
\end{aligned}$$

23. We can define the set $S = \{x \mid x \text{ is a positive integer and } x \text{ is a multiple of } 5\}$ by the basis step requirement that $5 \in S$ and the recursive requirement that if $n \in S$, then $n + 5 \in S$. Alternately we can mimic Example 7, making the recursive part of the definition that $x + y \in S$ whenever x and y are in S.

25. **a)** Since we can generate all the even integers by starting with 0 and repeatedly adding or subtracting 2, a simple recursive way to define this set is as follows: $0 \in S$; and if $x \in S$ then $x + 2 \in S$ and $x - 2 \in S$.

b) The smallest positive integer congruent to 2 modulo 3 is 2, so we declare $2 \in S$. All the others can be obtained by adding multiples of 3, so our inductive step is that if $x \in S$, then $x + 3 \in S$.

c) The positive integers not divisible by 5 are the ones congruent to 1, 2, 3, or 4 modulo 5. Therefore we can proceed just as in part (b), setting $1 \in S$, $2 \in S$, $3 \in S$, and $4 \in S$ as the base cases, and then declaring that if $x \in S$, then $x + 5 \in S$.

27. a) If we apply each of the recursive step rules to the only element given in the basis step, we see that $(0,1)$, $(1,1)$, and $(2,1)$ are all in S. If we apply the recursive step to these we add $(0,2)$, $(1,2)$, $(2,2)$, $(3,2)$, and $(4,2)$. The next round gives us $(0,3)$, $(1,3)$, $(2,3)$, $(3,3)$, $(4,3)$, $(5,3)$, and $(6,3)$. And a fourth set of applications adds $(0,4)$, $(1,4)$, $(2,4)$, $(3,4)$, $(4,4)$, $(5,4)$, $(6,4)$, $(7,4)$, and $(8,4)$.

b) Let $P(n)$ be the statement that $a \le 2b$ whenever $(a,b) \in S$ is obtained by n applications of the recursive step. For the basis step, $P(0)$ is true, since the only element of S obtained with no applications of the recursive step is $(0,0)$, and indeed $0 \le 2 \cdot 0$. Assume the strong inductive hypothesis that $a \le 2b$ whenever $(a,b) \in S$ is obtained by k or fewer applications of the recursive step, and consider an element obtained with $k+1$ applications of the recursive step. Since the final application of the recursive step to an element (a,b) must be applied to an element obtained with fewer applications of the recursive step, we know that $a \le 2b$. So we just need to check that this inequality implies $a \le 2(b+1)$, $a+1 \le 2(b+1)$, and $a+2 \le 2(b+1)$. But this is clear, since we just add $0 \le 2$, $1 \le 2$, and $2 \le 2$, respectively, to $a \le 2b$ to obtain these inequalities.

c) This holds for the basis step, since $0 \le 0$. If this holds for (a,b), then it also holds for the elements obtained from (a,b) in the recursive step, since adding $0 \le 2$, $1 \le 2$, and $2 \le 2$, respectively, to $a \le 2b$ yields $a \le 2(b+1)$, $a+1 \le 2(b+1)$, and $a+2 \le 2(b+1)$.

29. a) Since we are working with positive integers, the smallest pair in which the sum of the coordinates is even is $(1,1)$. So our basis step is $(1,1) \in S$. If we start with a point for which the sum of the coordinates is even and want to maintain this parity, then we can add 2 to the first coordinate, or add 2 to the second coordinate, or add 1 to each coordinate. Thus our recursive step is that if $(a,b) \in S$, then $(a+2,b) \in S$, $(a,b+2) \in S$, and $(a+1,b+1) \in S$. To prove that our definition works, we note first that $(1,1)$ has an even sum of coordinates, and if (a,b) has an even sum of coordinates, then so do $(a+2,b)$, $(a,b+2)$, and $(a+1,b+1)$, since we added 2 to the sum of the coordinates in each case. Conversely, we must show that if $a+b$ is even, then $(a,b) \in S$ by our definition. We do this by induction on the sum of the coordinates. If the sum is 2, then $(a,b) = (1,1)$, and the basis step put (a,b) into S. Otherwise the sum is at least 4, and at least one of $(a-2,b)$, $(a,b-2)$, and $(a-1,b-1)$ must have positive integer coordinates whose sum is an even number smaller than $a+b$, and therefore must be in S by our definition. Then one application of the recursive step shows that $(a,b) \in S$ by our definition.

b) Since we are working with positive integers, the smallest pairs in which there is an odd coordinate are $(1,1)$, $(1,2)$, and $(2,1)$. So our basis step is that these three points are in S. If we start with a point for which a coordinate is odd and want to maintain this parity, then we can add 2 to that coordinate. Thus our recursive step is that if $(a,b) \in S$, then $(a+2,b) \in S$ and $(a,b+2) \in S$. To prove that our definition works, we note first that $(1,1)$, $(1,2)$, and $(2,1)$ all have an odd coordinate, and if (a,b) has an odd coordinate, then so do $(a+2,b)$ and $(a,b+2)$, since adding 2 does not change the parity. Conversely (and this is the harder part), we must show that if (a,b) has at least one odd coordinate, then $(a,b) \in S$ by our definition. We do this by induction on the sum of the coordinates. If $(a,b) = (1,1)$ or $(a,b) = (1,2)$ or $(a,b) = (2,1)$, then the basis step put (a,b) into S. Otherwise either a or b is at least 3, so at least one of $(a-2,b)$ and $(a,b-2)$ must have positive integer coordinates whose sum is smaller than $a+b$, and therefore must be in S by our definition, since we haven't changed the parities. Then one application of the recursive step shows that $(a,b) \in S$ by our definition.

c) We use two basis steps here, $(1,6) \in S$ and $(2,3) \in S$. If we want to maintain the parity of $a+b$ and the fact that b is a multiple of 3, then we can add 2 to a (leaving b alone), or we can add 6 to b (leaving a

alone). So our recursive step is that if $(a,b) \in S$, then $(a+2,b) \in S$ and $(a,b+6) \in S$. To prove that our definition works, we note first that $(1,6)$ and $(2,3)$ satisfy the condition, and if (a,b) satisfies the condition, then so do $(a+2,b)$ and $(a,b+6)$, since adding 2 or 6 does not change the parity of the sum, and adding 6 maintains divisibility by 3. Conversely (and this is the harder part), we must show that if (a,b) satisfies the condition, then $(a,b) \in S$ by our definition. We do this by induction on the sum of the coordinates. The smallest sums of coordinates satisfying the condition are 5 and 7, and the only points are $(1,6)$, which the basis step put into S, $(2,3)$, which the basis step put into S, and $(4,3) = (2+2,3)$, which is in S by one application of our recursive definition. For a sum greater than 7, either $a \geq 3$, or $a \leq 2$ and $b \geq 9$ (since $2+6$ is not odd). This implies that either $(a-2,b)$ or $(a,b-6)$ must have positive integer coordinates whose sum is smaller than $a+b$ and satisfy the condition for being in S, and hence are in S by our definition. Then one application of the recursive step shows that $(a,b) \in S$ by our definition.

31. The answer depends on whether we require fully parenthesized expressions. Assuming that we do not, then the following definition is the most straightforward. Let F be the required collection of formulae. The basis step is that all specific sets and all variables representing sets are to be in F. The recursive part of the definition is that if α and β are in F, then so are $\overline{\alpha}$, (α), $\alpha \cup \beta$, $\alpha \cap \beta$, and $\alpha - \beta$. If we insist on parentheses, then the recursive part of the definition is that if α and β are in F, then so are $\overline{\alpha}$, $(\alpha \cup \beta)$, $(\alpha \cap \beta)$, and $(\alpha - \beta)$.

33. Let $D = \{0,1,2,3,4,5,6,7,8,9\}$ be the set of decimal digits. We think of a string as either being an element of D or else coming from a shorter string by appending an element of D, as in Definition 2. This problem is like Example 9.

a) The basis step is for a string of length 1, i.e., an element of D. If $x \in D$, then $m(x) = x$. For the recursive step, if the string $s = tx$, where $t \in D^*$ and $x \in D$, then $m(s) = \min(m(s), x)$. In other words, if the last digit in the string is smaller than the minimum digit in the rest of the string, then the last digit is the smallest digit in the string; otherwise the smallest digit in the rest of the string is the smallest digit in the string.

b) Recall the definition of concatenation (Definition 3). The basis step does not apply, since s and t here must be nonempty. Let $t = wx$, where $w \in D^*$ and $x \in D$. If $w = \lambda$, then $m(st) = m(sx) = \min(m(s), x) = \min(m(s), m(x))$ by the recursive step and the basis step of the definition of m in part **(a)**. Otherwise, $m(st) = m((sw)x) = \min(m(sw), x)$ by the definition of m in part **(a)**. But $m(sw) = \min(m(s), m(w))$ by the inductive hypothesis of our structural induction, so $m(st) = \min(\min(m(s), m(w)), x) = \min(m(s), \min(m(w), x))$ by the meaning of min. But $\min(m(w), x) = m(wx) = m(t)$ by the recursive step of the definition of m in part **(a)**. Thus $m(st) = \min(m(s), m(t))$.

35. The string of length 0, namely the empty string, is its own reversal, so we define $\lambda^R = \lambda$. A string w of length $n+1$ can always be written as vy, where v is a string of length n (the first n symbols of w), and y is a symbol (the last symbol of w). To reverse w, we need to start with y, and then follow it by the first part of w (namely v), reversed. Thus we define $w^R = y(v^R)$. (Note that the parentheses are for our benefit—they are not part of the string.)

37. We set $w^0 = \lambda$ (the concatenation of no copies of w should be defined to be the empty string). For $i \geq 0$, we define $w^{i+1} = ww^i$, where this notation means that we first write down w and then follow it with w^i.

39. The recursive part of this definition tells us that the only way to modify a string in A to obtain another string in A is to tack a 0 onto the front and a 1 onto the end. Starting with the empty string, then, the only strings we get are λ, 01, 0011, 000111, ... In other words, $A = \{\, 0^n 1^n \mid n \geq 0 \,\}$.

41. The basis step is $i = 0$, where we need to show that the length of w^0 is 0 times the length of w. This is true, no matter what w is, since $l(w^0) = l(\lambda) = 0$. Assume the inductive hypothesis that $l(w^i) = i \cdot l(w)$. Then

$l(w^{i+1}) = l(ww^i) = l(w) + l(w^i)$, this latter equality having been shown in Example 14. Now by the inductive hypothesis we have $l(w) + l(w^i) = l(w) + i \cdot l(w) = (i+1) \cdot l(w)$, as desired.

43. This is similar to Theorem 2. For the full binary tree consisting of just the root r the result is true since $n(T) = 1$ and $h(T) = 0$, and $1 \geq 2 \cdot 0 + 1$. For the inductive hypothesis we assume that $n(T_1) \geq 2h(T_1) + 1$ and $n(T_2) \geq 2h(T_2) + 1$ where T_1 and T_2 are full binary trees. By the recursive definitions of $n(T)$ and $h(T)$, we have $n(T) = 1 + n(T_1) + n(T_2)$ and $h(T) = 1 + \max(h(T_1), h(T_2))$. Therefore $n(T) = 1 + n(T_1) + n(T_2) \geq 1 + 2h(T_1) + 1 + 2h(T_2) + 1 \geq 1 + 2 \cdot \max(h(T_1), h(T_2)) + 2$ since the sum of two nonnegative numbers is at least as large as the larger of the two. But this equals $1 + 2(\max(h(T_1), h(T_2)) + 1) = 1 + 2h(T)$, and our proof is complete.

45. The basis step requires that we show that this formula holds when $(m, n) = (0, 0)$. The inductive step requires that we show that if the formula holds for all pairs smaller than (m, n) in the lexicographic ordering of $\mathbf{N} \times \mathbf{N}$, then it also holds for (m, n). For the basis step we have $a_{0,0} = 0 = 0 + 0$. For the inductive step, assume that $a_{m',n'} = m' + n'$ whenever (m', n') is less than (m, n) in the lexicographic ordering of $\mathbf{N} \times \mathbf{N}$. By the recursive definition, if $n = 0$ then $a_{m,n} = a_{m-1,n} + 1$; since $(m-1, n)$ is smaller than (m, n), the inductive hypothesis tells us that $a_{m-1,n} = m - 1 + n$, so $a_{m,n} = m - 1 + n + 1 = m + n$, as desired. Now suppose that $n > 0$, so that $a_{m,n} = a_{m,n-1} + 1$. Again we have $a_{m,n-1} = m + n - 1$, so $a_{m,n} = m + n - 1 + 1 = m + n$, and the proof is complete.

47. a) It is clear that $P_{mm} = P_m$, since a number exceeding m can never be used in a partition of m.

b) We need to verify all five lines of this definition, show that the recursive references are to a smaller value of m or n, and check that they take care of all the cases and are mutually compatible. Let us do the last of these first. The first two lines take care of the case in which either m or n is equal to 1. They are consistent with each other in case $m = n = 1$. The last three lines are mutually exclusive and take care of all the possibilities for m and n if neither is equal to 1, since, given any two numbers, either they are equal or one is greater than the other. Note finally that the third line allows $m = 1$; in that case the value is defined to be P_{11}, which is consistent with line one, since $P_{1n} = 1$.

Next let us make sure that the logic of the definition is sound, specifically that P_{mn} is being defined in terms of P_{ij} for $i \leq m$ and $j \leq n$, with at least one of the inequalities strict. There is no problem with the first two lines, since these are not recursive. The third line is okay, since $m < n$, and P_{mn} is being defined in terms of P_{mm}. The fourth line is also okay, since here P_{mm} is being defined in terms of $P_{m,m-1}$. Finally, the last line is okay, since the subscripts satisfy the desired inequalities.

Finally, we need to check the content of each line. (Note that so far we have hardly even discussed what P_{mn} means!) The first line says that there is only one way to write the number 1 as the sum of positive integers, none of which exceeds n, and that is patently true, namely as $1 = 1$. The second line says that there is only one way to write the number m as the sum of positive integers, none of which exceeds 1, and that, too, is obvious, namely $m = 1 + 1 + \cdots + 1$. The third line says that the number of ways to write m as the sum of integers not exceeding n is the same as the number of ways to write m as the sum of integers not exceeding m as long as $m < n$. This again is true, since we could never use a number from $\{m+1, m+2, \ldots, n\}$ in such a sum anyway. Now we begin to get to the meat. The fourth line says that the number of ways to write m as the sum of positive integers not exceeding m is 1 plus the number of ways to write m as the sum of positive integers not exceeding $m - 1$. Indeed, there is exactly one way to write m as the sum of positive integers not exceeding m that actually uses m, namely $m = m$; all the rest use only numbers less than or equal to $m - 1$. This verifies line four. The real heart of the matter is line five. How can we write m as the sum of positive integers not exceeding n? We may use an n, or we may not. There are exactly $P_{m,n-1}$ ways to form the sum without using n, since in that case each summand is less than or equal to $n - 1$. If we do use at least one n, then we have $m = n + (m - n)$. The number of ways this can be done, then, is the same as the number of ways to complete the partition by writing $(m - n)$ as the sum of positive integers not exceeding n. Thus there are $P_{m-n,n}$ ways to write m as the sum of numbers not exceeding n, at least one of which equals n. By the sum rule (see Chapter 5), we have $P_{mn} = P_{m,n-1} + P_{m-n,n}$, as desired.

c) We expand each P_{mn} according to the definition. For the first problem we have $P_5 = P_{55} = 1 + P_{54} = 1+P_{53}+P_{14} = 1+P_{52}+P_{23}+1 = 1+P_{51}+P_{32}+P_{22}+1 = 1+1+P_{31}+P_{12}+1+P_{21}+1 = 1+1+1+1+1+1+1 = 7$. For the second problem we have $P_6 = P_{66} = 1+P_{65} = 1+P_{64}+P_{15} = 1+P_{63}+P_{24}+1 = 1+P_{62}+P_{33}+P_{22}+1 = 1+P_{61}+P_{42}+1+P_{32}+1+P_{21}+1 = 1+1+P_{41}+P_{22}+1+P_{31}+P_{12}+1+1+1 = 1+1+1+1+P_{21}+1+1+1+1+1+1 = 1+1+1+1+1+1+1+1+1+1+1 = 11$.

49. We prove this by induction on m. The basis step is $m = 1$, so we need to compute $A(1,2)$. Line four of the definition tells us that $A(1,2) = A(0, A(1,1))$. Since $A(1,1) = 2$, by line three, we see that $A(1,2) = A(0,2)$. Now line one of the definition applies, and we see that $A(1,2) = A(0,2) = 2 \cdot 2 = 4$, as desired. For the inductive step, assume that $A(m-1,2) = 4$, and consider $A(m,2)$. Applying first line four of the definition, then line three, and then the inductive hypothesis, we have $A(m,2) = A(m-1, A(m,1)) = A(m-1,2) = 4$.

51. **a)** We use the results of Exercises 49 and 50: $A(2,3) = A(1, A(2,2)) = A(1,4) = 2^4 = 16$.
b) We have $A(3,3) = A(2, A(3,2)) = A(2,4)$ by Exercise 49. Now one can show by induction (using the result of Exercise 50) that $A(2,n)$ is equal to $2^{2^{\cdot^{\cdot^{2}}}}$, with n 2's in the tower. Therefore the answer is $2^{2^{2^2}} = 2^{16} = 65{,}536$.

53. It is often the case in proofs by induction that you need to prove something stronger than the given proposition, in order to have a stronger inductive hypothesis to work with. This is called **inductive loading** (see the preamble to Exercise 70 in Section 4.1). That is the case with our proof here. We will prove the statement "$A(m,k) > A(m,l)$ if $k > l$ for all m, k, and l," and we will use **double induction**, inducting first on m, and then within the inductive step for that induction, inducting on k (using strong induction). Note that this stronger statement implies the statement we are trying to prove—just take $k = l + 1$.

The basis step is $m = 0$, in which the statement at hand reduces (by line one of the definition) to the true conditional statement that if $k > l$, then $2k > 2l$. Next we assume the inductive hypothesis on m, namely that $A(m,x) > A(m,y)$ for *all* values of x and y with $x > y$. We want now to show that if $k > l$, then $A(m+1,k) > A(m+1,l)$. This we will do by induction on k. For the basis step, $k = 0$, there is nothing to prove, since the condition $k > l$ is vacuous. Similarly, if $k = 1$, then $A(m+1,k) = 2$ and $A(m+1,l) = 0$ (since necessarily $l = 0$), so the desired inequality holds. So assume the inductive hypothesis (using strong induction), that $A(m+1,r) > A(m+1,s)$ whenever $k > r > s$, where $k \geq 2$. We need to show that $A(m+1,k) > A(m+1,l)$ if $k > l$. Now $A(m+1,k) = A(m, A(m+1,k-1))$ by line four of the definition. Since $k - 1 \geq l$, we apply the inductive hypothesis on k to yield $A(m+1,k-1) > A(m+1,l-1)$, and therefore by the inductive hypothesis on m, we have $A(m, A(m+1,k-1)) > A(m, A(m+1,l-1))$. But this latter value equals $A(m+1,l)$, as long as $l \geq 2$. Thus we have shown that $A(m+1,k) > A(m+1,l)$ as long as $l \geq 2$. On the other hand, if $l = 0$ or 1, then $A(m+1,l) \leq 2$ (by lines two and three of the definition), whereas $A(m+1,2) = 4$ by Exercise 49. Therefore $A(m+1,k) \geq A(m+1,2) > A(m+1,l)$. This completes the proof.

55. We repeatedly invoke the result of Exercise 54, which says that $A(m+1,j) \geq A(m,j)$. Indeed, we have $A(i,j) \geq A(i-1,j) \geq \cdots \geq A(0,j) = 2j \geq j$.

57. Let $P(n)$ be the statement "F is well-defined at n; i.e., $F(n)$ is a well-defined number." We need to show that $P(n)$ is true for all n. We do this by strong induction. First $P(0)$ is true, since $F(0)$ is well-defined by the specification of $F(0)$. Next assume that $P(k)$ is true for all $k < n$. We want to show that $P(n)$ is also true, in other words that $F(n)$ is well-defined. Since the definition gave $F(n)$ in terms of $F(0)$ through $F(n-1)$, and since we are assuming that these are all well-defined (our inductive hypothesis), we conclude that $F(n)$ is well-defined, as desired.

59. **a)** This would be a proper definition if the recursive part were stated to hold for $n \geq 2$. As it stands, however, $F(1)$ is ambiguous.

b) This definition makes no sense as it stands; $F(2)$ is not defined, since $F(0)$ isn't.

c) For $n = 4$, the recursive part makes no sense, since we would have to know $F(4/3)$. Also, $F(3)$ is ambiguous.

d) The definition is ambiguous about $n = 1$, since both the second clause and the third clause seem to apply. If the second clause is restricted to odd $n \geq 3$, then the sequence is well-defined and begins 1, 2, 2, 3, 3, 3, 4, 4, 5, 4.

e) We note that $F(1)$ is defined explicitly, but we run into problems trying to compute $F(2)$:

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

This not only leaves us begging the question as to what $F(2)$ is, but is a contradiction, since $0 \neq 1$.

61. In each case we will apply the definition to compute $\log^{(0)}$, then $\log^{(1)}$, then $\log^{(2)}$, then $\log^{(3)}$ and so on. As soon as we get an answer no larger than 1 we stop; the last "exponent" is the answer. In other words $\log^* n$ is the number of times we need to apply the log function until we get a value less than or equal to 1. Note that $\log^{(1)} n = \log n$ for $n > 0$. Similarly, $\log^{(2)} n = \log(\log n)$ as long as it is defined ($n > 1$), $\log^{(3)} n = \log(\log(\log n))$ as long as it is defined ($n > 2$), and so on. Normally the parentheses are understood and omitted.

a) $\log^{(0)} 2 = 2$, $\log^{(1)} 2 = \log 2 = 1$; therefore $\log^* 2 = 1$, the last "exponent".

b) $\log^{(0)} 4 = 4$, $\log^{(1)} 4 = \log 4 = 2$, $\log^{(2)} 4 = \log 2 = 1$; therefore $\log^* 4 = 2$, the last "exponent". We had to take the log twice to get from 4 down to 1.

c) $\log^{(0)} 8 = 8$, $\log^{(1)} 8 = \log 8 = 3$, $\log^{(2)} 8 = \log 3 \approx 1.585$, $\log^{(3)} 8 \approx \log 1.585 \approx 0.664$; therefore $\log^* 8 = 3$, the last "exponent". We had to take the log three times to get from 8 down to something no bigger than 1.

d) $\log^{(0)} 16 = 16$, $\log^{(1)} 16 = \log 16 = 4$, $\log^{(2)} 16 = \log 4 = 2$, $\log^{(3)} 16 = \log 2 = 1$; therefore $\log^* 16 = 3$, the last "exponent". We had to take the log three times to get from 16 down to 1.

e) $\log^{(0)} 256 = 256$, $\log^{(1)} 256 = \log 256 = 8$; by part **(c)**, we need to take the log three more times in order to get from 8 down to something no bigger than 1, so we have to take the log four times in all to get from 256 down to something no bigger than 1. Thus $\log^* 256 = 4$.

f) $\log 65536 = 16$; by part **(d)**, we need to take the log three more times in order to get from 16 down to 1, so we have to take the log four times in all to get from 65536 down to 1. Thus $\log^* 65536 = 4$.

g) $\log 2^{2048} = 2048$; taking log four more times gives us, successively, 11, approximately 3.46, approximately 1.79, approximately 0.84. So $\log^* 2^{2048} = 5$.

63. Each application of the function f subtracts another a from the argument. Therefore iterating this function k times (which is what $f^{(k)}$ does) has the effect of subtracting ka. Therefore $f^{(k)}(n) = n - ka$. Now $f_0^*(n)$ is the smallest k such that $f^{(k)}(n) \leq 0$, i.e., $n - ka \leq 0$. Solving this for k easily yields $k \geq n/a$. Thus $f_0^*(n) = \lceil n/a \rceil$ (we need to take the ceiling function because k must be an integer).

65. Each application of the function f takes the square root of its argument. Therefore iterating this function k times (which is what $f^{(k)}$ does) has the effect of taking the $(2^k)^{\text{th}}$ root. Therefore $f^{(k)}(n) = n^{1/2^k}$. Now $f_2^*(n)$ is the smallest k such that $f^{(k)}(n) \leq 2$, that is, $n^{1/2^k} \leq 2$. Solving this for n easily yields $n \leq 2^{2^k}$, so

$k \geq \log\log n$, where logarithm is taken to the base 2. Thus $f_2^*(n) = \lceil \log\log n \rceil$ for $n \geq 2$ (we need to take the ceiling function because k must be an integer) and $f_2^*(1) = 0$.