Solutions to Homework 5

Section 4.1

4.

(a) Plugging in $n = 1$ we have that $P(1)$ is the statement $1^3 = [1 \cdot (1+1)/2]^2$.
(b) Both sides of $P(1)$ shown in part (a) equal 1.
(c) The inductive hypothesis is the statement that

$$1^3 + 2^3 + \cdots + k^3 = \left(\frac{k(k+1)}{2}\right)^2.$$

(d) For the inductive step, we want to show for each $k \geq 1$ that $P(k)$ implies $P(k+1)$. In other words, we want to show that assuming the inductive hypothesis we can prove

$$[1^3 + 2^3 + \cdots + k^3] + (k+1)^3 = \left(\frac{(k+1)(k+2)}{2}\right)^2.$$

(e) Replacing the quantity in brackets on the left-hand side of part (d) by what it equals by virtue of the inductive hypothesis, we have

$$\left(\frac{k(k+1)}{2}\right)^2 + (k+1)^3 = (k+1)^2\left(\frac{k^2}{4} + k + 1\right) = \left(\frac{(k+1)(k+2)}{2}\right)^2$$

as desired.

(f) We have completed both the basis step and the inductive step, so by the principle of mathematical induction, the statement is true for every positive integer n.

20.

The basis step is $n = 7$, and indeed $3^7 < 7!$, since $2187 < 5040$. Assume the statement for k. Then $3^{k+1} = 3 \cdot 3^k < (k+1) \cdot 3^k < (k+1) \cdot k! = (k+1)!$, the statement for $k+1$.

38.

The basis step is trivial, as usual: $A_1 \subseteq B_1$ implies that $\bigcup_{j=1}^{1} A_j \subseteq \bigcup_{j=1}^{1} B_j$ because the union of one set is itself. Assume the inductive hypothesis that if $A_j \subseteq B_j$ for $j = 1, 2, \ldots, k$, then $\bigcup_{j=1}^{k} A_j \subseteq \bigcup_{j=1}^{k} B_j$. We want to show that if $A_j \subseteq B_j$ for $j = 1, 2, \ldots, k+1$, then $\bigcup_{j=1}^{k+1} A_j \subseteq \bigcup_{j=1}^{k+1} B_j$. To show that one set is a subset of another we show that an arbitrary element of the first set must be an element of the second set. So let $x \in \bigcup_{j=1}^{k+1} A_j = \left(\bigcup_{j=1}^{k} A_j\right) \cup A_{k+1}$. Either $x \in \bigcup_{j=1}^{k} A_j$ or $x \in A_{k+1}$. In the first case we know by the inductive hypothesis that $x \in \bigcup_{j=1}^{k} B_j$, in the second case, we know from the given fact that $A_{k+1} \subseteq B_{k+1}$ that $x \in B_{k+1}$. Therefore in either case $x \in \left(\bigcup_{j=1}^{k} B_j\right) \cup B_{k+1} = \bigcup_{j=1}^{k+1} B_j$.

Section 4.2

4.

(a) $P(18)$ is true, because we can form 18 cents of postage with one 4-cent stamp and two 7-cent stamps. $P(19)$ is true, because we can form 19 cents of postage with three 4-cent stamps and one 7-cent stamp. $P(20)$ is true, because we can form 20 cents of postage with five 4-cent stamps. $P(21)$ is true, because we can form 21 cents of postage with three 7-cent stamps.

(b) The inductive hypothesis is the statement that using just 4-cent and 7-cent stamps we can form j cents postage for all j with $18 \leq j \leq k$, where we assume that $k \geq 21$.

(c) In the inductive step we must show, assuming the inductive hypothesis, that we can form $k+1$ cents postage using just 4-cent and 7-cent stamps.

(d) We want to form $k+1$ cents of postage. Since $k \geq 21$, we know that $P(k-3)$ is true, that is , that we can form $k-3$ cents of postage. Put one more 4-cent stamp on the envelope, and we have formed $k+1$ cents of postage, as desired.

(e) We have completed both the basis step and inductive step, so by the principle of strong induction, the statement is true for every integer n greater than or equal to 18.

12.

The basis step is to note that $1 = 2^0$. Notice for subsequent steps that $2 = 2^1$, $3 = 2^1+2^0$, $4 = 2^2$, $5 = 2^2+2^0$, and so on. Indeed this is simply the representation of a number in binary form (base two). Assume the inductive hypothesis, that every positive integer up to k can be written as a sum of distinct powers of 2. We must show that $k+1$ can be written as a sum of distinct powers of 2. If $k+1$ is odd, then k is even, so 2^0 was not part of the sum for k. Therefore the sum for $k+1$ is the same as the sum for k with the extra term 2^0 added. If $k+1$ is even, then $(k+1)/2$ is a positive integer, so by the inductive hypothesis $(k+1)/2$ can be written as a sum of distinct powers of 2. Increasing each exponent by 1 doubles the value and gives us the desired sum for $k+1$.

Section 4.3

6.

(a) This is valid, since we are provided with the value at $n = 0$, and each subsequent value is determined by the previous one. Since all that changes from one value to the next is the sign, we conjecture that $f(n) = (-1)^n$. This is true for $n = 0$, since $(-1)^0 = 1$. If it is true for $n = k$, then we have $f(k+1) = -f(k+1-1) = -f(k) = -(-1)^k$ by the inductive hypothesis, $f(k+1) = (-1)^{k+1}$.

(c) This is invalid. We are told that $f(2)$ is defined in terms of $f(3)$, but $f(3)$ has not been defined.

12.

The basis step is clear, since $f_1^2 = f_1 f_2 = 1$. Assume the inductive hypothesis. Then $f_1^2 + f_2^2 + \cdots + f_n^2 + f_{n+1}^2 = f_n f_{n+1} + f_{n+1}^2 = f_{n+1}(f_n + f_{n+1}) = f_{n+1} f_{n+2}$, as desired.

Extra Problem 1:

We induct on w. The basis step is $(x \cdot \lambda)^R = \lambda x^R = \lambda^R \cdot x^R$. For the inductive step assume $w = w_1 y$, where w_1 is a string of length one less than the length of w. Then $(x \cdot w)^R = (x \cdot w_1 y)^R = y(x \cdot w_1)^R = y \cdot w_1^R \cdot x^R = (w_1 y)^R \cdot x^R = w^R \cdot x^R$.

Section 4.4

14.

The recursive algorithm will need to keep track not only of what the mode actually is, but also of how often the mode appears. We will describe this algorithm in words rather than in pseudocode. The input is a list $a_1, a_2, \ldots, a_n$ of integers. Call this list L. If $n = 1$, then the output is that the mode is a_1 and it appears 1 time. For the recursive case, form a new list L' by deleting from L the term a_n and all terms in L equal to a_n. Let k be the number of terms deleted. If $k = n$, then the output is that the mode is a_n and it appears n times. Otherwise, apply the algorithm recursively to L', obtaining a mode m, which appears t times. Now if

$t \geq k$, then the output is that the mode is m and it appears t times, otherwise the output is that the mode is a_n and it appears k times.

18.

We use mathematical induction on n. If $n = 0$, we know that $0! = 1$ by definition, so the if clause handles this basis step correctly. Now fix $k \geq 0$ and assume the inductive hypothesis that the algorithm correctly computes $k!$. Consider what happens with input $k + 1$. Since $k + 1 > 0$, the else clause is executed, and the answer is whatever the algorithm gives as output for input k, which by inductive hypothesis is $k!$, multiplied by $k+1$. But by definition, $k! \cdot (k+1) = (k+1)!$, so the algorithm works correctly on input $k + 1$.

32.

This is very similar to the recursive procedure for computing the Fibonacci numbers. Note that we can combine the three base cases into one.

```
procedure sequence(n: nonnegative integer)
if n<3 then sequence(n):=n+1
else sequence(n):=sequence(n−1)+sequence(n−2)+sequence(n−3)
```

34. The iterative algorithm is much more efficient here. If we compute with the recursive algorithm, we end up computing the small values over and over again.

44. Step1: 34251867, Step2: 23451678, Step3: 12345678.

46.

(c) From the analysis given before the statement of Lemma1, it follows that the number of comparisons is $m + n - r$, where the lists have m and n elements, respectively, and r is the number of elements remaining in one list at the point the other list is exhausted. In this exercise $m = n = 5$ and $r = 2$. Thus the answer is 8.

50. Five levels of recursion are required. 123578946, 123457896, 123456789, 123456789, 123456789.