Section 4.2

4.

a) It’s true for 18, 19, 20, 21 because 18 = 7 + 7 + 4, 19 = 4 + 4 + 7 + 4, 20 = 4 + 4 + 4 + 4 + 4, 21 = 7 + 7 + 7.

b) Inductive hypothesis: Assume that for some k we have that \( P(i) \) is true for all i s.t. \( 18 \leq i \leq k \).

(Note that “usually” the inductive hypothesis in a strong induction proof would be that \( P(i) \) is true for all \( i \leq k \), but that actually only applies to proofs of theorems where the base case is \( k = 0 \). Here the base case is \( k = 18 \) and hence we can only assume that \( P(i) \) is true for \( i \) between 18 and \( k \).)

c) What we need to prove is that \( P(k + 1) \) holds, i.e. that \( k + 1 \) can be represented as a sum of 7’s and 4’s, assuming the inductive hypothesis on \( k \), as stated above.

d) The proof is simple: Since \( k + 1 = k' + 4 \) if \( k' = k - 3 \) then if you can represent \( k' = k - 3 \) as a sum of 7’s and 4’s then you can do the same with \( k + 1 \). In other words \( P(k - 3) \) implies \( P(k + 1) \). But is \( P(k - 3) \) implied by the inductive assumption stated in part (b)? Yes, because \( k \geq 21 \) and therefore \( k - 3 \geq 18 \), and at the same time obviously \( k - 3 \geq k \), so therefore the inductive assumption, that \( P(i) \) holds for all \( i \) between 18 and \( k \), covers the case \( i = k - 3 \), and therefore it implies \( P(k - 3) \). Since we’ve seen that \( P(k - 3) \) implies \( P(k + 1) \), this implies \( P(k + 1) \), as needed.

e) This finishes the proof because part (a) covered the cases \( n = 18, 19, 20, 21 \), and part (d) showed that for any \( k \geq 21 \), if \( P(i) \) holds for all \( i \) between 18 and \( k \) then it holds for \( k + 1 \), and this implies that \( P(n) \) holds for all \( n \geq 21 \). Thus \( P(n) \) holds for all \( n \geq 18 \).

10. The answer is that exactly \( n - 1 \) breaks are needed. Let \( P(n) \) state that every \( n \)-squared rectangular bar needs exactly \( n - 1 \) breaks to be broken to single pieces. The base case: \( P(1) \) is obviously true. The (strong) inductive case: Let \( k \) be any integer \( k \geq 1 \). Assume that \( P(i) \) holds for all \( i \) s.t. \( 1 \leq i \leq k \). We’ll show that \( P(k + 1) \) holds as well. Why? Assume you have a \((k + 1)\)-squared rectangular bar. Assume you broke it somehow using a single break, and hence now you have two rectangular bars, with \( n_1 \) and \( n_2 \) pieces each s.t. \( n_1 + n_2 = k + 1 \). Moreover we have \( 1 \leq n_1, n_2 \leq k \). Therefore the inductive assumption kicks in for both \( n_1 \) and \( n_2 \), and therefore we know that it takes exactly \( n_1 - 1 \) steps to break the first bar and exactly \( n_2 - 1 \) steps to break the second bar. (Note that you couldn’t do this step that easily using standard induction!!) Therefore it took \((n_1 - 1) + (n_2 - 1) + 1 = (n_1 + n_2) - 1 = (k + 1) - 1\) steps to break the main bar, which was what we needed to show.

12. Let \( P(n) \) say that natural \( n \) can be represented as a sum of distinct powers of two. Base case: \( P(1) \) is true since \( 1 = 2^0 \). Base case: Let \( k \) by any integer \( k \geq 1 \). Assume \( P(i) \) holds for all \( i \) s.t. \( 1 \leq i \leq k \). We’ll show that \( P(k + 1) \) holds as well.

First consider the case that \( k + 1 \) is odd. Then \( k \) is even, and hence divisible by 2. By the inductive assumption \( k \) can be represented as a sum of distinctive powers of 2. Also, these powers must all be non-zero or otherwise \( k \) would be not be divisible by 2. Since \( 1 = 2^0 \) and \( k \) can be represented as a sum of distinct non-zero powers of 2 we get that \( k + 1 \) can be represented as a sum of distinct powers of 2 (one of them is a zero-th power...).

Now consider the case that \( k + 1 \) is even. Therefore \( k + 1 = 2k' \) for some integer \( k' \). Moreover, since \( k + 1 \geq 2 \), we have \( k' \geq 1 \). Therefore \( P(k') \) holds by the inductive assumption, and hence you can represent \( k' \) as a sum of distinct powers of 2. But this means that you can represent \( k + 1 = 2k' \) also as a sum of distinct powers of 2: Just increase each power by one: If they were distinct, then after incrementing each of them by one they remain distinct!

In either case we showed that \( P(k + 1) \) is true, which ends the proof. Note that the first case (i.e. odd \( k + 1 \)) would work in the same if you attempted to prove this using standard induction, but you’d run into problems in the second case!

26.

a) For all nonnegative even integers.

b) For all nonnegative integers which are divisible by 3.

c) For all nonnegative integers.

d) For all integers \( n \) s.t. \( n = 0 \) or \( n \geq 2 \).
30. The error is in the proof of the inductive case: This proof fails for \( k = 0 \), because while it’s true (by the base case) that \( a^k = a^0 = 1 \), it’s not true that \( a^{k-1} = a^{-1} = 1 \). This is a classic case of failing to use the inductive assumption correctly. If the inductive assumption is that \( P(i) \) is true for all \( i \) between \( n_0 \) (the base case) and \( k \) for some \( k \), then you have to make sure that in your proof of \( P(k+1) \) you don’t attempt to use the (unproven) assumption that \( P(i) \) is true for some \( i < n_0 \).

Section 4.3

6.  
   a) Good definition. \( f(n) = (-1)^n \). You can easily prove it by standard induction.
   b) Good definition. \( f(n) = 2^{n/3} \) for all \( n \) s.t. \( n = 0 \) mod 3, \( f(n) = 0 \) for all \( n \) s.t. \( n = 1 \) mod 3, and \( f(n) = 2^{(n+1)/3} \) for all \( n \) s.t. \( n = 2 \) mod 3. You can prove each of the above by standard induction.
   c) Bad definition. Attempt to define \( f(n) \) in terms of \( f(n+1) \), i.e. values of \( f \) on higher points than \( n \).
   d) Bad definition because \( f(1) \) is defined as 1 in the base case and as \( 2 \ast f(0) = 2 \ast 0 = 0 \) in the inductive case.
   e) It’s good, although it’s odd because \( f(n) \) is defined “twice”, i.e. using two different rules, in the case that \( n \) is odd and greater than 1. However, these two rules actually define the same thing so this definition is redundant but it does define a (unique) function on nonnegative integers. The function is \( f(n) = 2^{(n/2)+1} \). The base case \( f(0) = 2 \) matches, it’s true that \( f(n) = f(n-1) \) for odd nonnegative \( n \), and it’s also true that \( f(n) = 2f(n-2) \) for all \( n \geq 2 \).

8.  
   a) For example, \( a_1 = 2 \) and \( a_n = a_{n-1} + 4 \) for \( n \geq 2 \).
   b) For example, \( a_1 = 0 \), \( a_2 = 2 \), and \( a_n = a_{n-2} \) for all \( n \geq 3 \).
   c) For example, \( a_1 = 2 \) and \( a_n = a_{n-1} + 2n \) for all \( n \geq 2 \).
   d) For example, \( a_1 = 1 \) and \( a_n = a_{n-1} + 2n - 1 \) for all \( n \geq 2 \).

20. First define a function \( \max_2 \) which takes the max of a pair of numbers, i.e. \( \max_2(a, b) \) is defined as \( a \) if \( a \geq b \) and \( b \) if \( b < a \). Then define function \( \max \) recursively as follows: In the base case, \( n = 2 \), we define \( \max(a_1, a_2) = \max_2(a_1, a_2) \). In the inductive case, when \( n \geq 3 \), we define \( \max(a_1, a_2, ..., a_n) = \max_2(\max(a_1, \max(a_2, ..., a_n))) \).

   You can also define it without the “auxiliary” function like \( \max_2 \), but it’s more convenient to do it this way. For comparison, let’s define the minimum function without introducing any auxiliary functions: In the base case, \( n = 2 \), we define \( \min(a_1, a_2) \) as \( a_1 \) if \( a_1 \leq a_2 \) and \( a_2 \) otherwise (i.e. if \( a_2 < a_1 \)). In the inductive case, \( n \geq 3 \), we define \( \min(a_1, a_2, ..., a_n) \) as \( a_1 \) if \( a_1 \leq \min(a_2, ..., a_n) \), and as \( \min(a_2, ..., a_n) \) otherwise.

24.  
   a) We can define the set \( S \) of odd positive integers for example as follows: The basis step: Let \( 1 \in S \). The inductive step: If \( n \in S \) then \( n + 2 \in S \).
   b) We can define the set \( S \) of powers of 3 as follows: The basis step: Let \( 1 \in S \). The inductive step: If \( n \in S \) then \( 3n \in S \).
   c) We can define the set \( S \) of polynomials with integer coefficients as follows: The basis step: Let \( 0 \in S \) and \( 1 \in S \). The inductive step: If \( p(n) \in S \) then \( p'(n) \in S \) where \( p'(n) = a + n \ast p(n) \) for some integer \( a \).

32.  
   a) Recall that strings are defined recursively as follows. Basis case: \( \lambda \in S \). Inductive case: If \( s \in S \) then \( s0 \in S \) and \( s1 \in S \). Therefore we need to define any function, likes \( \text{ones} \), on strings by defining it for the same two cases. Here’s the definition of \( \text{ones} \) that meet this criterion. In the basis case, \( \text{ones}(\lambda) = 0 \). In the inductive case, \( \text{ones}(s0) = \text{ones}(s) \) and \( \text{ones}(s1) = \text{ones}(s) + 1 \).
   b) Let’s denote the string-concatenation function explicitly as “\( \cdot \)” and bit-concatenation by just putting a bit after a string, with no operator between them. The claim is that \( \text{ones}(s \cdot t) = \text{ones}(s) + \text{ones}(t) \) for
any strings $s, t$. Recall that the concatenation function $s \cdot t$ is defined recursively as follows. Basis case: $s \cdot \lambda = s$. Inductive case: $s \cdot (tb) = (s \cdot t)b$.

Therefore we can prove the claim using structural induction as follows. Base case: The claim we need to show is that $\text{ones}(s \cdot \lambda) = \text{ones}(s) + \text{ones}(\lambda)$, but this is true because $\text{ones}(\lambda) = 0$ by definition of $\text{ones}$. The inductive case: The claim we need to show is that $\text{ones}(s \cdot (tb)) = \text{ones}(s) + \text{ones}(tb)$. This claim is true because:

$$\text{ones}(s \cdot (tb)) = \text{ones}((s \cdot t)b) \quad \text{by inductive definition of string concatenation}$$

$$= \text{ones}(s \cdot t) + \text{ones}(b) \quad \text{by inductive definition of the 'ones' function}$$

$$= \text{ones}(s) + \text{ones}(t) + \text{ones}(b) \quad \text{by the (structural) inductive assumption}$$

$$= \text{ones}(s) + \text{ones}(tb) \quad \text{by the inductive definition of the 'ones' function}$$

**44.** Claim: For every tree $T$, $l(T) = i(T) + 1$. This was proven in the lecture on Friday. The basis case is when $T$ is a single vertex, in which case $l(T) = 1$, $i(T) = 0$, and thus $l(T) + i(T) + 1$ holds. In the inductive case $T$ is made from two other trees $T_1, T_2$, which is denoted $T = T_1 \cdot T_2$. Here’s why the claim is true in this case as well:

$$l(T_1 \cdot T_2) = l(T_1) + l(T_2) \quad \text{by inductive definition of the 'leaves' function}$$

$$= (i(T_1) + 1) + (i(T_2) + 1) \quad \text{by the (structural) inductive assumption}$$

$$= i(T_1) + i(T_2) + 1$$

$$= i(T) + 1 \quad \text{by inductive definition of the 'internal nodes' function}$$

**48.**

$$A(1, 0) = 0,$$

$$A(0, 1) = 2 \cdot 1 = 2,$$

$$A(1, 1) = 2,$$

$$A(2, 2) = A(1, A(2, 1)) = A(1, 2) = A(0, A(1, 1)) = 2A(1, 1) = 4.$$  

**50.**

Basis case: $A(1, 1) = 2$ by part (48) above.

Inductive case: Assume that $A(1, n) = 2^n$ for some $n \geq 1$. Then $A(1, n + 1) = A(0, A(1, n))$ because $n + 1 > 2$, so the last rule kicks in. But $A(0, A(1, n)) = 2A(1, n)$ by the first rule, and $2A(1, n) = 2 \cdot 2^n = 2^{n+1}$ by the inductive assumption, so we get that $A(1, n + 1) = 2^{n+1}$, as needed.

**58.**

a) This is not a good definition because $F(1)$ is defined twice: Once in the base case as 1 and once in the inductive case as a function of $F(1 - 0) = F(0)$, but $F(0)$ is undefined.

b) This is not a good definition for the similar reason: $F(2)$ is defined once in the base case as 3 and again in the inductive case in terms of $F(2 - 3) = F(-1)$, but $F(-1)$ is undefined.

c) This definition again “double-defines” $F(2)$, but actually that’s fine because it defines it consistently: In the base case $F(2)$ is defined as 2 and in the inductive case $F(2) = 1 + F(1) = 1 + 1 = 2$ again. However, the inductive part does not define the function for odd $n$. For example $F(3)$ is defined in terms of $F(3/2)$, which is undefined.

d) This is bad because $F(1)$ is defined twice. First in the base case as 1, and second in the inductive case as a function of $F(1 - 1) = F(0)$, which is undefined.

e) This is bad because $F(n)$ for odd $n \geq 3$ are defined in terms of values of $F(n')$ for $n'$ larger than $n$.  

Page 3 of 3