CS225 Assignment 6 Solution Set

EECS, Oregon State University

Summer, 2012

4.3 Recursive Definitions

Textbook 4.3

2. Find $f(1), f(2), f(3), f(4)$, and $f(5)$ if $f(n)$ is defined recursively by $f(0) = 3$ and for $n = 0, 1, 2, \ldots$
Solution:

a). $f(n+1) = -2f(n)$.

$$\begin{aligned} f(1) &= -2f(0) = -2 * 3 = -6, \\ f(2) &= -2f(1) = -2 * (-6) = 12, \\ f(3) &= -2f(2) = -2 * (12) = -24, \\ f(4) &= -2f(3) = -2 * (-24) = 48, \\ f(5) &= -2f(4) = -2 * (48) = -96. \end{aligned}$$

b). $f(n+1) = 3f(n) + 7$.

$$\begin{aligned} f(1) &= 3f(0) + 7 = 3 * 3 + 7 = 16, \\ f(2) &= 3f(1) + 7 = 3 * 16 + 7 = 55, \\ f(3) &= 3f(2) + 7 = 3 * 55 + 7 = 172, \\ f(4) &= 3f(3) + 7 = 3 * 172 + 7 = 523, \\ f(5) &= 3f(4) + 7 = 3 * 523 + 7 = 1576. \end{aligned}$$

8. Given a recursive definition of the sequence $\{a_n\}, n = 1, 2, 3, \ldots$ if
Solution:

a). $a_n = 4n - 2$.
Basis Step: Specify a_1 by $a_1 = 4 * 1 - 2 = 2$.
Recursive Step: Give a rule for finding a_{n+1} from a_n, for $n \geq 1$:

$$\begin{aligned} a_{n+1} &= 4(n+1) - 2 \\ &= 4n + 4 - 2 \\ &= 4n - 2 + 4 \\ &= a_n + 4 \end{aligned}$$

b). $a_n = 1 + (-1)^n$.
Basis Step: Specify a_1 by $a_1 = 1 + (-1)^1 = 0$.
Recursive Step: Give a rule for finding a_{n+1} from a_n, for $n \geq 1$:

$$\begin{aligned} a_{n+1} &= 1 + (-1)^{n+1} \\ &= 1 + (-1)^n(-1) \\ &= 1 + \left[((-1)^n + 1) - 1\right](-1) \\ &= 1 + (a_n - 1)(-1) \\ &= 2 - a_n. \end{aligned}$$

c). $a_n = n(n+1)$.
Basis Step: Specify a_1 by $a_1 = 1(1+1) = 2$.
Recursive Step: Give a rule for finding a_{n+1} from a_n, for $n \geq 1$:

$$\begin{aligned} a_{n+1} &= (n+1)[(n+1)+1] \\ &= n(n+1) + n + (n+1) + 1 \\ &= a_n + 2n + 2. \end{aligned}$$

d). $a_n = n^2$.
Basis Step: Specify a_1 by $a_1 = 1^2 = 1$.
Recursive Step: Give a rule for finding a_{n+1} from a_n, for $n \geq 1$:

$$\begin{aligned} a_{n+1} &= (n+1)^2 \\ &= n^2 + 2n + 1 \\ &= a_n + 2n + 1. \end{aligned}$$

24. Given a recursive definition of
Solution:

a). the set of odd positive integers.
Basis Step: $1 \in S$.
Recursive Step: If $x \in S$, then $x + 2 \in S$.

b). the set of positive integer power of 3.
Basis Step: $3 \in S$.
Recursive Step: If $x \in S$, then $3x \in S$.

c). the set of polynomials with integer coefficients.
Basis Step: $0 \in S$.
Recursive Step: If $p(x) \in S$, then $p(x) + cx^n \in S$, where $c \in Z, n \in Z$, and $n \geq 0$.
(Z is the set of integers.)

26. a) Let S be the subset of the set of ordered pairs of integers defined by
Basis Step: $(0,0) \in S$.
Recursive Step: If $(a,b) \in S$, then $(a+2, b+3) \in S$, and $(a+3, b+2) \in S$.

List the elements of S produced by the first five applications of the recursive definition.

Solution:

$$\begin{array}{llllllll}
1): & (2,3), & (3,2); \\
2): & (4,6), & (5,5), & (6,4); \\
3): & (6,9), & (7,8), & (8,7), & (9,6); \\
4): & (8,12), & (9,11), & (10,10), & (11,9), & (12,8); \\
5): & (10,15), & (11,14), & (12,13), & (13,12), & (14,11), & (15,10).
\end{array}$$

28. Give a recursive definition of each of these sets of ordered pairs for positive integers. [*Hint:* Plot the points in the set in the plane and look for lines containing points in the set].
Solution:

a). $S = \{(a, b) | a \in \mathbf{Z}^+, b \in \mathbf{Z}^+, \text{ and } a + b \text{ is odd.}\}$
Solution:
Basis Step: $(1, 2) \in S$.
Recursive Step: If $(a, b) \in S$, then $(a, b + 2) \in S$, $(a + 2, b) \in S$, and $(a + 1, b + 1) \in S$.

All elements put in S satisfy the condition, because $(1, 2)$ has an odd sum of coordinates, and if (a, b) has an odd sum of coordinates, then so do $(a + 2, b), (a, b + 2)$, and $(a + 1, b + 1)$.

Conversely, we show by induction that if $a+b$ is odd, then $(a, b) \in S$. If the sum is 3 (3 is the basis step because it is the smallest possible sum of a and b), then $(a, b) = (1, 2)$, and the basis step put (a, b) in to S. Otherwise the sum is at least 5, 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 odd number smaller than $a + b$, and therefore must be in S. Then one application of the recursive step shows that $(a, b) \in S$.

32.a) Give a recursive definition of the function $ones(s)$, which counts the number of ones in a bit string s. (A bit string is a string of zeros and ones.)
Solution:
Let $\Sigma = \{0, 1\}$. The function is defined by:
Basis Step: $ones(\lambda) = 0$, where λ is the empty string containing no symbols.
Recursive Step: If $x \in \Sigma$, and $w \in \Sigma^*$, then $ones(wx) = ones(w) + x$, where x is a bit: either 0 or 1.
(Σ^* is the set of *strings* over the alphabet Σ and Σ^* contains λ. Note that λx is equal to x.)

4.4 Structural Induction

Textbook 4.3

12. Prove that $f_1^2 + f_2^2 + \cdots + f_n^2 = f_n f_{n+1}$ when n is a positive integer.
Solution:
Let $P(n)$ be $f_1^2 + f_2^2 + \cdots + f_n^2 = f_n f_{n+1}$.
Basis Step: $f_1^2 = f_1 * f_2$ is true, because $f_1 = 1, f_2 = 1$, and $f_1^2 = 1 = 1 \cdot 1 = f_1 * f_2$.

Recursive Step: We use mathematical induction. Assuming $P(n)$ is true, we need to prove $P(n+1)$ is true. That is, assuming $f_1^2 + f_2^2 + \cdots + f_n^2 = f_n f_{n+1}$, we want to show $f_1^2 + f_2^2 + \cdots + f_n^2 + f_{n+1}^2 = f_{n+1} f_{n+2}$. We show that

$$\begin{array}{rcll}
f_1^2 + f_2^2 + \cdots + f_n^2 + f_{n+1}^2 & = & (f_1^2 + f_2^2 + \cdots + f_n^2) + f_{n+1}^2 & \\
& = & f_n f_{n+1} + f_{n+1}^2 & \text{Inductive Hypothesis} \\
& = & f_{n+1}(f_n + f_{n+1}) & \text{Factorization} \\
& = & f_{n+1} f_{n+2} & \text{Definition of Fibonacci number}
\end{array}$$

26. c) Let S be the subset of the set of ordered pairs of integers defined by

Basis Step: $(0,0) \in S$.

Recursive Step: If $(a,b) \in S$, then $(a+2, b+3) \in S$, and $(a+3, b+2) \in S$.

Use structural induction to show that $5|a+b$ when $(a,b) \in S$.

Solution:

Basis Step: This holds for the basis step because $5|(0+0=0)$.

Recursive Step: In the recursive step, we need to show that if $5|a+b$ holds, then this also holds for the elements obtained from (a,b). Suppose $a+b=5k$ for some integer k. Then $5|(a+2)+(b+3)$, because $(a+2)+(b+3) = a+b+5 = 5k+5 = 5(k+1)$, where $k+1$ is also an integer. Similarly, $5|(a+3)+(b+2)$, because $(a+3)+(b+2) = a+b+5 = 5k+5 = 5(k+1)$, where $k+1$ is also an integer. This completes our structural induction proof.

43. Use structural induction to show that $n(T) \geq 2h(T)+1$, where T is a full binary tree, $n(T)$ equals the number of vertices of T, and $h(T)$ is the height of T.

Solution:

Basis Step: For the full binary tree consisting of just a root the result is true, because $n(T) = 1$ and $h(T) = 0$, and $1 \geq 2 \cdot 0 + 1$.

Recursive Step: We use strong induction. Assume the result holds for all full binary trees smaller than T. We need to show that $n(T) \geq 2h(T)+1$ for the full binary tree T. By the recursive definition of a full binary tree, T is formed by two subtrees T_1 and T_2 with the addition of a root node, where T_1 and T_2 are smaller than T. By the induction hypothesis, we know that the inductive hypothesis holds for T_1 and T_2, i.e. $n(T_1) \geq 2h(T_1)+1$ and $n(T_2) \geq 2h(T_2)+1$. By the recursive definition of $n(T)$ and $h(T)$(*pp. 306*), we have $n(T) = 1 + n(T_1) + n(T_2)$ and $h(T) = 1 + \max(h(T_1), h(T_2))$. Therefore,

$$\begin{array}{rll}
n(T) & = 1 + n(T_1) + n(T_2) & \\
& \geq 1 + 2h(T_1) + 1 + 2h(T_2) + 1 & \text{Inductive Hypothesis} \\
& \geq 1 + 2\max(h(T_1), h(T_2)) + 2 & \text{By } 2h(T_1) + 2h(T_2) \geq 2\max(h(T_1), h(T_2)) \\
& = 1 + 2(\max(h(T_1), h(T_2)) + 1) & \text{Factorization} \\
& = 1 + 2h(T) & \text{Recursive definition of full binary tree}
\end{array}$$

This completes our induction proof.

44. Use structural induction to show that $l(T)$, the number of leaves of a full binary tree T, is 1 more than $i(T)$, the number of internal vertices of T.

Solution:

Basis Step: The smallest full binary tree is a single root r. This has no internal vertices and the root itself is a leaf. So $l(T) = 1 = 1 + i(T)$.

Recursive Step: We use strong induction. Assume the result holds for all full binary trees smaller than T. We need to show that the result holds for T. By the recursive definition of a full binary tree, T is formed by two subtrees T_1 and T_2 with the addition of a root node, where T_1 and T_2 are smaller than T. By the induction hypothesis, we know that the result holds for T_1 and T_2, i.e., $l(T_1) = i(T_1) + 1$ and $l(T_2) = i(T_2) + 1$. We also know that $l(T) = l(T_1) + l(T_2)$ and $i(T) = i(T_1) + i(T_2) + 1$, where the addition of 1 comes from the root node. Hence,

$$\begin{array}{rcll} l(T) & = & l(T_1) + l(T_2) & \text{Recursive definition of full binary tree} \\ & = & i(T_1) + 1 + i(T_2) + 1 & \text{Induction Hypothesis} \\ & = & i(T) + 1 & \text{Recursive definition of full binary tree} \end{array}$$

This completes our inductive step.

58.(pp.332) The set B of all **balanced strings of parentheses** is defined recursively by $\lambda \in B$, where λ is the empty string; $(x) \in B, xy \in B$ if $x, y \in B$.

Use induction to show that if x is a balanced string of parentheses, then the number of left parentheses equals the number of right parentheses in x.

Solution:

Basis Step: The result holds for the basis step because the empty string does not have left or right parentheses. That is λ has equal number of left and right parentheses, which is 0.

Recursive Step: Assume the results hold for $x, y \in B$,i.e., both x and y have equal numbers of left parentheses and right parentheses, we need to show that the result holds for all elements obtained from $x, y \in B$. Let l_x and l_y be the number of left parentheses of x and y respectively. Let r_x and r_y be the number of right parentheses of x and y respectively. Then we know that $l_x = r_x$ and $l_y = r_y$.

$(x) \in S$ has 1 more left parentheses and 1 more right parentheses than x. That is, it has $l_x + 1$ left parentheses and $r_x + 1$ right parentheses. Since $l_x + 1 = r_x + 1$. (x) has the same number of left and right parentheses.

The number of left parentheses in string xy is $l_x + l_y$, while the number of right parentheses is $r_x + r_y$. By inductive hypothesis, we know that $l_x + l_y = r_x + r_y$. Hence, xy also has the same number of left parentheses and right parentheses. This completes our induction proof.