COMP 681 Formal Methods Spring 2008 Recitation 14—<u>Solutions</u>

1. Prove the following by simple (weak) induction.

a. $\sum_{i=1}^{n} \frac{1}{i(i+1)} = \frac{n}{n+1}$

Answer

<u>Basis</u>: $n = 1$

$$LHS = \sum_{i=1}^{1} \frac{1}{i(i+1)} = \frac{1}{1(1+1)} = \frac{1}{2}$$

$$RHS = \frac{1}{1+2} = \frac{1}{2}$$

So $LHS = RHS$

<u>Induction step</u>

Assume that

$$\sum_{i=1}^{k} \frac{1}{i(i+1)} = \frac{k}{k+1}$$

and show that

$$\sum_{i=1}^{k+1} \frac{1}{i(i+1)} = \frac{k+1}{k+2}$$

Now,

$$\sum_{i=1}^{k+1} \frac{1}{i(i+1)} = \sum_{i=1}^{k} \frac{1}{i(i+1)} + \frac{1}{(k+1)(k+2)}$$

$$= \frac{k}{k+1} + \frac{1}{(k+1)(k+2)} \qquad \text{by the induction hypothesis}$$

$$= \frac{k(k+2)}{(k+1)(k+2)} + \frac{1}{(k+1)(k+2)}$$

$$= \frac{k^2 + 2k + 1}{(k+1)(k+2)}$$

$$= \frac{(k+1)(k+1)}{(k+1)(k+2)}$$

$$= \frac{k+1}{k+2} \qquad \text{Q.E.D.}$$

b. The number of n-bit (nonnegative) binary numbers is 2^n.

Answer

Basis $n = 1$

The 1-bit binary numbers are 0 and 1.
So there are $2 = 2^1$ 1-bit binary numbers (as required).

Induction step

Assume there are 2^k k-bit binary numbers (and show that there are 2^{k+1} (k+1)-bit binary numbers).

Note that the first k bits of a (k+1)-bit binary number correspond to a k-bit binary number. Now, all and only the odd (k+1)-bit binary numbers are generated by $2 \times b + 1$, where b ranges over all k-bit binary numbers. By the induction hypotheses, then, there are 2^k odd (k+1)-bit binary numbers. Similarly, all and only the even (k+1)-bit binary numbers are generated by $2 \times b + 0$, where b ranges over all k-bit binary numbers. Again by the induction hypotheses, there are 2^k even (k+1)-bit binary numbers. Since the (k+1)-bit binary numbers are partitioned into the odd (k+1)-bit binary numbers and the even (k+1)-bit binary numbers, the total number of (k+1)-bit binary numbers is $2^k + 2^k = 2 \cdot 2^k = 2^{k+1}$. Q.E.D.

2. Prove the following by strong induction.

A binary tree with n nodes has height at least $\lfloor \lg n \rfloor$.

Answer

Basis: $n = 1$

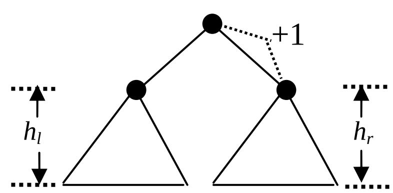
(Note that this does not cover the case $n = 0$, i.e., the empty binary tree, for which the height is not defined.)
A binary tree with 1 node has just a root thus has height 0, and $\lfloor \lg 1 \rfloor = \lfloor 0 \rfloor$, as required.

Induction step

Assume that, for all i, $1 \le i \le k$, a binary tree with i nodes has height at least $\lfloor \log i \rfloor$ (and show that a binary tree of height k+1 has height at least $\lfloor \lg (k+1) \rfloor$).
We're considering a lower bound on the height, so we want to consider the partition of non-root nodes between left and right subtrees that results in the smallest height of the binary tree. Since the height of a binary tree is the maximum of the heights of its subtrees plus 1 (see the figure at right), the partition in question divides the k non-root nodes as equally as possible between the two subtrees.

There are two cases depending on whether k is even or odd.

The height of a binary tree



Where h is the height of the binary tree, h_l is the height of the left substree, and h_r is the height of the right sub tree,
$h = \max(h_l, h_r) + 1$

Case of k even:

Then, since k+1 is odd, k+1 is not a power of 2, and $\lfloor \lg (k+1) \rfloor = \lfloor \lg k \rfloor$. Then the non-root nodes can be partitioned evenly, with k/2 nodes in each subtree. By the induction hypothesis, the height of each subtree is at least $\lfloor \lg k/2 \rfloor = \lfloor \lg k - \lg 2 \rfloor$. =

$\lfloor \lg k - 1 \rfloor$. So the maximum height of the subtrees is $\lfloor \lg k - 1 \rfloor$, and the height of the overall binary tree is $\lfloor \lg k - 1 \rfloor + 1 = \lfloor \lg k \rfloor = \lfloor \lg (k+1) \rfloor$ (as required).

Case of *k* odd:

Then the non-root nodes cannot be partitioned evenly. The best we can do is have $(k-1)/2$ nodes in one subtree and $(k+1)/2$ in the other. The latter is at least as high as the former and, by the induction hypothesis, has height at least $\lfloor \lg ((k+1)/2) \rfloor = \lfloor \lg (k+1) - \lg 2 \rfloor = \lfloor \lg (k+1) - 1 \rfloor$, which is the maximum of the heights of the subtrees. (If $k-1 = 0$, then there is only one non-empty subtree, and the maximum is over only one height, giving the same result.) The height of the overall binary tree, then, is at least $\lfloor \lg (k+1) - 1 \rfloor + 1 = \lfloor \lg (k+1) \rfloor$. Q.E.D.