Mini-Quiz 2

1. Write your name:

2. (Rosen, Sec. 1.5, Ex. 21) Let A and B be subsets of a universal set U. Show that $A \subseteq B$ if and only if $\overline{B} \subseteq \overline{A}$.

3. (Rosen, Sec. 1.6, Ex. 15) Determine whether each of the following functions is a bijection from $\mathbb{R}$ to $\mathbb{R}$.

   (a) $f(x) = 2x + 1$
   (b) $f(x) = x^2 + 1$
   (c) $f(x) = x^3$

Massachusetts Institute of Technology
6.042J/18.062J: Mathematics for Computer Science
Professors David Karger and Nancy Lynch

Handout 7
February 11, 2000

Tutorial 2 Problems

Problem 1. False Proofs

(a) Consider the following recursive definition of the set F of full binary trees:

1. A single node is in F.
2. If t and t' are in F, then the structure formed from t, t', and a single new node, with an arrow labeled "left" from the new node to t and another arrow labeled "right" from the new node to t', is also in F.

The recursive definition of the set F of full binary trees is similar to the one for binary trees, except that we only allow the recursive definition step that includes both a left and a right child.

Find the flaw in the following proof. Define the *height* of a binary tree to be 0 for a single node, otherwise one more than the larger of its two children's heights.

Claim. *Any height-n binary tree has exactly* $2^n - 1$ *nodes.*

Proof. By induction. The base case is clear. For the inductive step, assume it is true for height-n binary trees. Putting two of them together under a new node gives a height $n+1$ binary tree which has $2 * (2^n - 1) + 1 = 2^{n+1} - 1$ nodes. This proves the inductive step. ■

Solution:
The basic problem here is that we are building *some* height $n+1$ binary tree, but we are not showing the theorem for *all* height $n+1$ binary trees. There are height $n+1$ trees that are *not* assembled from two height n trees: for example, I could have a height n left subtree and a height zero right subtree.

This general flaw, of building some but not all of the possible "bigger instances," is known as *buildup error*.

(b) Claim. *For $n \in \mathbb{N}, n \geq 1$, any $2n \times 2n$ square grid can be tiled by L-shaped tiles such that any particular square is left untiled.*

Proof. The proof is by strong induction on n. Let $P(n)$ be the predicate: "Any $2n \times 2n$ square grid can be tiled by L-shaped tiles such that any particular square is left untiled".

Base Case:

The base case, for $n = 1$, is very simple, just use one tile.

Inductive Step:

Assume that the claim is true for all square grids smaller than $2n$, and we prove it for the $2n \times 2n$ square grid. This follows the proof of the $2^n \times 2^n$ case: Given a $2n \times 2n$ square grid, with one square designated to be left untiled, we construct a tiling (using L-shaped tiles) for all the other squares, as follows: Divide the square grid up into four quadrants, each $n \times n$. By the inductive hypothesis (since $n < 2n$ for all $n \geq 1$), we can tile each of these quadrants (with the L-shaped tiles) completely, except for one square of our choice. For the quadrant with the square to be left untiled, choose that square. For the others, choose the squares at the corner where the quadrants meet. This forms an L-shaped space at the center of the $2n \times 2n$ square grid, which can be filled by a single L-shaped tile. Thus, the entire $2n \times 2n$ square grid is tiled, except for the square designated.

■

Point out any logical errors in this proof (that is, not simply places where some steps were skipped, but where there is actually an error, though you are free to point out places where more steps should have been included for clarity). Then, either construct a correct proof, or show that the claim is false.

Solution:

The error in the proof is that we erroneously invoke the induction hypothesis. The inductive hypothesis states that we can tile all square grids of size $2m \times 2m$, for all $m \in \mathbb{N}, m < n$. That is, we can tile all square grids of even size that are smaller than the $2n \times 2n$ square grid. When we invoke the induction hypothesis, we use it to claim that we can tile an $n \times n$ square grid. The problem occurs when n is odd.

In fact, the claim is false. Consider $n = 3$, *i.e.*, a 6×6 square grid. The grid contains 36 squares which is divisible by 3 the number of squares of an L-shaped tile. Thus, it is not possible to tile such a grid so that exactly one particular square is left untiled.

Problem 2. Let F_n be the nth Fibonacci number, for all $n \in \mathbb{N}$. Recall that the Fibonacci numbers are defined recursively as follows: $F_0 = 0$, $F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$, for all $n \in \mathbb{N}, n \geq 2$.

(a) Prove that for all $n \in \mathbb{N}$, $\sum_{i=0}^{n} F_i = F_{n+2} - 1$.

(b) Find a formula for $\sum_{i=0}^{n} F_{2i}$ and prove that the formula is correct.

Solution:

(a) Let $P(n) : \sum_{i=0}^{n} F_i = F_{n+2} - 1$.

(b) Let $P(n) : \sum_{i=0}^{n} F_{2i} = F_{2n+1} - 1$.

Problem 3.

(a) Give the structural inductive definition of the set F of full binary trees.

(b) Define the function $numleaves(t)$, for $t \in F$, giving the number of "leaves" of the full binary tree t; that is, $numleaves(t)$, for $t \in F$, gives the number of nodes in t that do not have children. Your definition should be recursive, based on the recursive definition of the set F.

(c) Define the function $numinternal(t)$, for $t \in F$, giving the number of "internal" nodes of the full binary tree t; that is, $numinternal(t)$, for $t \in F$, gives the number of nodes in t that have children. Your definition should be recursive, based on the recursive definition of the set F.

(d) Prove by structural induction that for any full binary tree t in F, $numleaves(t) = 1 + numinternal(t)$.

Solution:

(a) *Carl: I am not sure whether this is what Nancy and David wanted as an answer here since we give the student's the recursive definition in problem 1a.*

Recall from lecture that we let $makeboth(t, t')$, for $t, t' \in F$, denote the full binary tree made up of attaching the full binary trees t and t' respectively to a node, as done in the recursive definition of F.

$\forall t \in F(P(t))$
1. (Base) $P(n)$, where n is the single-node tree.
2. (Inductive step) $\forall t, t' \in F, (P(t) \wedge P(t') \Rightarrow P(makeboth(t, t')))$
3. QED

(b) We define the function $numleaves(t)$, for $t \in F$, as follows:

$numleaves(\text{single node}) = 1$

$numleaves(makeboth(t, t')) = numleaves(t) + numleaves(t')$

(c) We define the function $numinternal(t)$, for $t \in F$, as follows:

$numinternal(\text{single node}) = 0$

$numinternal(makeboth(t, t')) = numinternal(t) + numinternal(t') + 1$

(d) This part is shown by structural induction on the recursion step.

Let $P(t) : numleaves(t) = 1 + numinternal(t)$.

Use the structural inductive definition of part (a) to show the property is true.