B501 Assignment 1

Due Date: January 18, 2012
Due Time: 11:00pm

1. Prove by mathematical induction that

$$\forall n \geq 0 : \sum_{i=0}^{n} 2^i = 2^{n+1} - 1.$$

2. Prove by mathematical induction that

$$\forall n \geq 0 : 13^n - 6^n \text{ is divisible by } 7$$

3. Prove by mathematical induction that

$$\forall n \geq 2 : 1 + 2^n < 3^n$$

4. Consider the following function `sum` from the natural numbers to the natural numbers. The natural numbers are denoted by N in this function.

```
function sum(n in N): N
{
  if n==0 return 0
  else return n + sum(n-1)
}
```

Prove by mathematical induction that

$$\forall n \geq 0 : \texttt{sum}(n) = \frac{n(n+1)}{2}$$

5. Define the set $\mathcal{B}$ of *binary trees* as follows:

   (a) A tree with a single node r is in $\mathcal{B}$; and

   (b) If r is a node and T_1 and T_2 are binary trees, i.e., $T_1 \in \mathcal{B}$ and $T_2 \in \mathcal{B}$, then the tree $T = (r, T_1, T_2)$ is a binary tree, i.e., T is in $\mathcal{B}$. You should view T as a tree with root r with r having as left child the tree T_1 and as right child the tree T_2.

Define a node of a binary tree to be a *full* if it has both a non-empty left and a non-empty right child. Prove by structural induction that the number of full nodes in a binary tree is 1 fewer than the number of leaves. (Hint: Consider binary trees as defined in class.)

6. Let E denote the set of arithmetic expressions. The recursive definition for E is as follows:

   - if n is a **positive** integer then n is in E;
   - if e_1 and e_2 are in E, then $(e_1 + e_2)$ is in E;
   - if e_1 and e_2 are in E, then $(e_1 * e_2)$ is in E.

   Write a recursive function `Replace` using appropriate pseudo-code which takes as input an expression in e in E and returns an expression in E wherein each number is replaced by the number 1.

   For example, if e is the expression

   $$((((2 + 3) * 3) * (5 + (3 * 5))))$$

   then `Replace`(e) is the expression

   $$((((1 + 1) * 1) * (1 + (1 * 1))))$$

   Then prove by structural induction on the recursive definition of the expressions in E that the value of an expression e in E is at least the value of `Replace`(e).

   For example, the value of

   $$((((2 + 3) * 3) * (5 + (3 * 5))))$$

   is 300, whereas the value of

   $$((((1 + 1) * 1) * (1 + (1 * 1))))$$

   is 4.