B501 Assignment 1
Enrique Areyan

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

1. Prove by mathematical induction that

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

Solution: Base case: $n = 0 \Rightarrow 2^0 = 1 = 2^{0+1} - 1$. It holds.

We want to prove that:

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1 \Rightarrow \sum_{i=0}^{n+1} 2^i = 2^{(n+1)+1} - 1$$

Proof:

$$\sum_{i=0}^{n+1} 2^i = \sum_{i=0}^{n} 2^i + 2^{n+1} = 2^{n+1} - 1 + 2^{n+1} = 2 \cdot 2^{n+1} - 1 = 2^{(n+1)+1} - 1$$

Q.E.D

2. Prove by mathematical induction that

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

Solution: A number x is divisible by 7 if $x = 7 \cdot k$, for some integer k
Base case: $n = 0 \Rightarrow 13^0 - 6^0 = 1 - 1 = 0 = 7 \cdot 0$. It holds.

We want to prove that:

$$13^n - 6^n \text{ is divisible by } 7 \Rightarrow 13^{n+1} - 6^{n+1} \text{ is divisible by } 7$$

Alternatively,

$$13^n - 6^n = 7 \cdot k \Rightarrow 13^{n+1} - 6^{n+1} = 7 \cdot l$$

where k and l are both integers.

Proof:

$13^{n+1} - 6^{n+1} = 13 \cdot 13^n - 6^{n+1} = 13 \cdot (7 \cdot k + 6^n) - 6 \cdot 6^n = 13 \cdot 7 \cdot k + 13 \cdot 6^n - 6 \cdot 6^n = 13 \cdot 7 \cdot k + 7 \cdot 6^n =$

$= 7 \cdot (13 \cdot k + 6^n) = 7 \cdot l,$ where l is an integer (both k and n are integers)

Q.E.D

3. Prove by mathematical induction that

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

Solution: Base case: $n = 2 \Rightarrow 1 + 2^2 = 1 + 4 = 5 < 9 = 3^2$. It holds.

We want to prove that:

$$1 + 2^n < 3^n \Rightarrow 1 + 2^{n+1} < 3^{n+1}$$

Proof:

$1 + 2^n < 3^n$	hypothesis
$2 + 2^{n+1} < 2 \cdot 3^n$	multiply by 2 the hypothesis
$2 + 2^{n+1} < 2 \cdot 3^n < 3 \cdot 3^n$	new upper bound still holds $(3 > 2)$
$1 + 2^{n+1} < 2 + 2^{n+1} < 2 \cdot 3^n < 3 \cdot 3^n$	new lower bound still holds
$1 + 2^{n+1} < 3^{n+1} = 3 \cdot 3^n$	follows from previous statement

Q.E.D

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

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

Solution: The function $\texttt{sum}(n)$ can be written as $\sum_{i=0}^{n} i$.

Base case: $n = 0 \Rightarrow \sum_{i=0}^{0} i = 0 = \frac{0(0+1)}{2}$. It holds.

We want to prove that:

$$\sum_{i=0}^{n} i = \frac{n(n+1)}{2} \Rightarrow \sum_{i=0}^{n+1} i = \frac{(n+1)((n+1)+1)}{2} = \frac{n^2+3n+2}{2}$$

Proof:

$$\sum_{i=0}^{n+1} i = \sum_{i=0}^{n} i + (n+1) = \frac{n(n+1)}{2} + (n+1) = \frac{n(n+1)+2n+2}{2} = \frac{n^2+3n+2}{2}$$

Q.E.D

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.)

   Solution:
   Let us first define $\cdot$ to be a tree with a single node, $T \in \mathcal{B}$ to be any binary tree according to the definition (T_1 being its left child and T_2 being its right), and the following two functions:

   $\#f : \mathcal{B} \mapsto \mathcal{N}$, number of full nodes defined as:
   1. $\#f(\cdot) = 0$
   2. $\#f(T) = 1 + \#f(T_1) + \#f(T_2)$

   $\#l : \mathcal{B} \mapsto \mathcal{N}$, number of leaves defined as:
   1. $\#l(\cdot) = 1$
   2. $\#l(T) = \#l(T_1) + \#l(T_2)$

   We want to show that the following property holds:

   $$\forall T \in B : \#f(T) = \#l(T) - 1$$

   Base case: $\#f(\cdot) = 0 = 1 - 1 = \#l(\cdot) - 1$. It holds.

   Proof:

   $$\begin{aligned} \#f(T) &= 1 + \#f(T_1) + \#f(T_2) && \text{(definition of } \#f) \\ &= 1 + \#l(T_1) - 1 + \#l(T_2) - 1 && \text{(hypothesis)} \\ &= \#l(T_1) + \#l(T_2) - 1 && \text{(by simple algebra)} \\ &= \#l(T) - 1 && \text{(by definition of } \#l)\text{. Q.E.D} \end{aligned}$$

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.

Solution:

First, let us define the function Replace (R) as follow (in a mathematical sense):
$R : E \mapsto E$, such that:
if $e = n$, a positive integer, then, $R(e) = 1$,
if $e = (e_1 + e_2)$ then, $R(e) = (R(e_1) + R(e_2))$,
if $e = (e_1 * e_2)$ then, $R(e) = (R(e_1) * R(e_2))$

Now, in pseudo code:

```
function replace(e in E): E
{
  if e>0 return 1
  else if e == e_1+e_2 return (R(e_1) + R(e_2))
  else return (R(e_1) * R(e_2))
}
```

Now, we want to prove a property of the members of this set. But before I do this, let us define yet another function:

$V : E \mapsto Z^+$ (V stands for value):
if $e = n,$ a positive integer, then, $V(e) = n$,
if $e = (e_1 + e_2)$ then, $V(e) = (V(e_1) + V(e_2))$,
if $e = (e_1 * e_2)$ then, $V(e) = (V(e_1) * V(e_2))$

The property we want to prove by structural induction is:

$$\forall e \in E : V(e) \geq V(R(e))$$

Base case: $e = n \Rightarrow V(e) = n$, by definition, and $V(R(n)) = V(1) = 1$.
Thus $n \geq 1$, the property holds.

Proof:
Let $e = (e_1 + e_2)$, then

$$\begin{aligned} V(e) = &\quad V(e_1) + V(e_2) && \text{(definition of } V\text{)} \\ \geq &\quad V(R(e_1)) + V(R(e_2)) && \text{(hypothesis)} \\ = &\quad V(R(e_1) + R(e_2)) && \text{(by definition of } V\text{, and the fact that we can consider } R(e_1) \text{ and} \\ & && R(e_2) \text{ to be just another two expressions in E.} \\ = &\quad V(R(e)) && \text{(by definition of } R\text{).} \end{aligned}$$

A similar proof follows for the operation $*$. Q.E.D