COMP 302 Assignment 2 Question 1

Yang David Zhou, ID 260517397

October 1, 2014

Problem 1. *Prove reflect (reflect t)= t.*

Proof. Proof by induction:

```
Base case:
reflect Empty = Empty
reflect (reflect Empty) = Empty

Induction hypothesis:
Assume reflect (reflect someNode) = someNode for left and right
in someNode = Node (x, left, right)

let someTree = Node (x, left, right)
reflect someTree = Node (x, reflect right, reflect left)

reflect (reflect someTree)
 = Node (x, reflect (reflect left), reflect (reflect right))
 = Node (x, left, right)   //from induction hypothesis
 = someTree
```

□

Problem 2. *Prove that for all m:'a tree, size m = size' m 0.*

Proof. Auxiliary proof by induction, then show that the statement is a special case:

```
Begin by proving size' someTree a = size someTree + a

Base case:
size' Empty a
 = a = 0 + a
 = size Empty + a

Induction hypothesis:
Assume size' someNode a = size someNode + a for left and right
in someNode = Node(x, left, right)
```

```
let someTree = Node (x, left, right)

size’ someTree a
 = size’ left (size’ right (a + 1))  //by function definition
 = size left + size’ right (1 + a)   //from induction hypothesis
 = size left + size right + 1 + a    //apply IH again
 = size Node (x’, left, right) + a   //by function definition

The value of x’ is irrelevant so we can set it to x to obtain:

 = size someTree + a

Now that I have proven size’ someTree a = size someTree + a,
I can show that there is the special case when a = 0

size m = size’ m 0
```

□