Solution to Homework 8

Problem 1: Prove by induction on lists

∀l, m : List Int. sum (l ++ m) = sum l + sum m

Choose an arbitrary m:List Int and do induction on List Int, l.

Base: Prove

sum ([ ] ++ m) = sum [ ] + sum m.

LHS= sum ([ ] ++ m) = sum m (by def of ++)
RHS = sum [ ] + sum m = 0 + sum m = sum m (by def of sum)

Base case is done.

Induction: Assume the property holds for an arbitrary l:List Int.

(I.H.) sum (l ++ m) = sum l + sum m

Show, for some x:Int,

sum (x:l ++ m) = sum (x:l) + sum m

LHS = sum (x:l ++ m) = sum (x : (l ++ m)) (by def of ++)
= x + (sum (l ++ m)) (by def of sum)
= x + sum l + sum m (by IH)
RHS = sum (x:l) + sum m = x + sum l + sum m (by def of sum)

Hence the property holds for all l.

Problem 2: Show by induction on BTrees that

∀t : BTree a. leafCount t = length (flatten t)

Prove by induction on t:BTree a.

Base: Show the property holds for Leaf x, where x:a.

leafCount (Leaf x) = length (flatten (Leaf x))
1 = length ([ x ]) (by def of leafCount, def of flatten)
1 = 1 (by def of length)

Hence the base case holds.

Induction: Assume the property holds for t1 and t2, both arbitrary BTree a.

(I.H.) leafCount (t1) = length (flatten t1)
leafCount (t2) = length (flatten t2)

Show

leafCount (Fork t1 t2) = length (flatten (Fork t1 t2))

LHS = leafCount t1 + leafCount t2 (by def of leafCount)
RHS = length (flatten t1 ++ flatten t2) (by def of flatten)
= length flatten t1 + length flatten t2 (by def of length, and proof in class)
= leafCount t1 + leafCount t2 (by IH)

Hence the property holds for all BTrees.