Homework 1 CS 557 Section 1.1

Soumya Banerjee[§]
[§] Department of Computer Science, University of New Mexico

I. Problem 1.1 Higher-order functions over lists

A. Problem 1.1.1 Filter

filter p (filter q xs) = filter (p &&& q) xs
where p &&& q =
x -> p x && q x
We prove this by induction.

Base Case -
LHS = filter p (filter q xs) where xs = []
= filter p (filter q [])
= { filter q [] = [], i.e. fiter on an emptyset returns the emptyset }
filter p ([])
= { filter p [] = [], i.e. fiter on an emptyset returns the emptyset}
[]

RHS = filter (p &&& q) xs where xs = []
= filter (p &&& q) []
= { filter x [] = [], i.e. fiter on an emptyset returns the emptyset }
= []
LHS = RHS. Hence the base case is proved.

Inductive Hypothesis -
filter p (filter q xs) = filter (p &&& q) xs

Inductive Step - x = x':xs' and prove that filter p (filter q x':xs') = filter (p &&& q) x':xs'
LHS = filter p (filter q x':xs')

= { By definition of filter - filter p y:ys = if p y then y : filter p ys else filter p ys }

if q x' == True then
filter p (x' : filter q xs')
= { By definition of filter - filter p y:ys = if p y then y : filter p ys else filter p ys }
if p x' == True
x' : filter p (filter q xs')
= { By Inductive Hypothesis}
x' : filter (p &&& q) xs'
else { if p x' == False}
filter p (filter q xs')
= { By Inductive Hypothesis}
filter (p &&& q) xs'
else { if q x' == False}
filter p (filter q xs')
= { By Inductive Hypothesis}
filter (p &&& q) xs'

RHS = filter (p &&& q) x':xs'
= { By definition of filter - filter p y:ys = if p y then y : filter p ys else filter p ys }
if (p &&& q) x' == True

x' : filter (p &&& q) xs'
{ Since the condition (p &&& q) x' == True is equivalent to p x' == True && q x' == True (by definition of the function &&&) we see that LHS = RHS}

else { (p &&& q) x' == False }
filter (p &&& q) xs'

{ We note that the condition (p &&& q) x' == False is equivalent to p x' == False OR even q x' == False (by definition of &&& function). We note that LHS = RHS }

Hence LHS = RHS

Hence we have proved by induction that filter p (filter q xs) = filter (p &&& q) xs where p &&& q =
x -> p x && q x

B. Problem 1.1.2 Map and concat

concat (map (map f) xs) = map f (concat xs)
We prove this by induction

Base Case - xs = []
LHS = concat (map (map f) xs) where xs = []
= concat (map (map f) [])
= { Let map f = z }
concat (map z [])
= {By definition of map on empty list - map q [] = [] }
concat ([])
= {By definition of concat on empty list - concat [] = [] }
= []

RHS = map f (concat xs) where xs = []
= map f (concat [])
= {By definition of concat on empty list - concat [] = [] }
map f []
= {By definition of map on empty list - map q [] = [] }
= []

Hence LHS = RHS for the base case

Inductive Hypothesis - concat (map (map f) xs) = map f (concat xs)

Inductive Step - Prove concat (map (map f) x':xs') = map f (concat x':xs') where x = x':xs'

LHS = concat (map (map f) x':xs')
= { By definition of map on the outer map we have - map q x':xs' = (qx') : (map q xs') }
concat ((map f x') : (map (map f) xs'))
= { By definition of concat, we have concat x:xs = x ++ (concat xs) }
(map f x') ++ concat (map (map f) xs')
= { By Inductive Hypothesis }
(map f x') ++ map f (concat xs)

RHS = map f (concat x':xs')
= { By definition of concat, we have concat x:xs = x ++ (concat xs) }
map f (x' ++ (concat xs'))
= { By definition of map, we have map f (x ++ xs) = map f x ++ map f xs }
map f x' ++ map f (concat xs')

Hence LHS = RHS

Hence we have proved by induction that concat (map (map f) xs) = map f (concat xs)

II. Pledge

I pledge my honor that in the preparation of this assignment I have complied with the University of New Mexico Board of Regents Policy Manual. Soumya Banerjee . Date - 6th October, 2008