Name : Adarsha Kanel
UCID: 30049820
CPSC 449
Tutorial T03

Assignment 4
Theory component

[Thompson] exercise 11.34. Use the following definition of the function concat:
concat = foldr (++) []
You may also use the axiom(map++)on page 261 (under Exercise 11.31)

concat(map (map f) xs) = map f (concat xs)

1. Definitions

concat = foldr (++) [] (c.1)

foldr f z [] = z (f.1)
foldr f z (x:xs) = f x (foldr f z xs) (f.2)

map f [] = [] (m.1)
map f (x:xs) = f x : map f xs (m.2)
map f (ys++zs) = map f ys ++ map f zs (m.3)

2. Proof goals

We want to prove the two goals of the induction proof:

For base case, we have to prove:

concat(map (map f) []) = map f (concat []) (BASE)

Then we are going to prove the induction step

concat(map (map f) xs) = map f (concat xs) (IND)

On the assumption that:

map f (ys++zs) = map f ys ++ map f zs

3. Proving the base case

concat(map (map f) xs) = map f (concat [])

Left hand-side
concat(map (map f) [])

= foldr (++) [] (map (map f) [])	by (c.1)
= foldr (++) [] []	by (m.1)
= []	by (f.1)

Right hand-side
map f (concat [])

= map f (foldr (++) [] [])	by (c.1)
= map f []	by (f.1)
= []	by (m.1)

So, LHS = RHS

4. Proving the inductive case

concat(map (map f) (x:xs)) = map f (concat (x:xs))

Left hand-side(LHS)
concat(map (map f) (x:xs))

= foldr (++) [] (map (map f) (x:xs))	by (c.1)
= foldr (++) [] (map f x : map (map f) xs)	by (m.2)
= (++) (map f x) (foldr (++) [] (map(map f) xs))	by (f.2)
= (++) (map f x) (concat (map(map f) xs)))	by (c.1)
= (++) (map f x) (map f (concat xs))	by (HYP)
= map f (x) ++ map f (concat xs)	pre-fix To infix

Right-hand side (RHS)
map f (concat (x:xs)

= map f (foldr (++) [] (x:xs))	by (c.1)
= map f ((++) x (foldr (++) [] xs))	by (f.2)
= map f (x ++ (foldr (++) [] xs))	prefix to infix
= map f (x) ++ map f (foldr (++) [] xs)	by (m.3)
= map f (x) ++ map f (concat xs)	by (c.1)

So LHS = RHS

[■]