CPSC 449 Assignment #4 Written Report

Fall 2020

Ali Akbari

30010402

Tutorial #3

Question #2)

Definition:

```
map f[] = []                                               (map.1)
map f(x:xs)= (f x : map f xs)                              (map.2)
map (f.g)xs = (map f.map g)xs                              (map.3)
map f (ys++zs) = map f ys ++ map f zs                      (map.4)

concat[[a]]= [a]                                           (concat.1)
concat [] = []                                             (concat.2)
concat (x:xs) = x ++ concat xs                             (concat.3)
concat = foldr (++) []                                     (concat.4)

foldr f s (x:xs)= f x (foldr f s xs)                       (foldr.1)

Question/Exercise 11.34:
concat (map (map f) xs) = map f (concat xs)
```

```
a) Principle of Structural Induction for concat exercise 11.34
   To prove that 11.34 holds for all finite lists xs and function f
   prove the following:
   concat (map (map f) xs) = map f (concat xs)

   Proof Goals:
     1) concat (map (map f) []) = map f (concat [])        (Base.1)
   Assume:
     2) concat (map (map f) xs) = map f (concat xs)        (hyp.1)
   Prove:
     3) concat (map (map f) (x:xs)) = map f (concat (x:xs))      (Ind.1)

b) Base Case:
   concat (map (map f) []) = map f (concat [])

   LHS:
   concat (map(map f)[])
   = concat []                                             (map.1)
   = []                                                    (concat.2)

   RHS:
   map f (concat [])
   = map f []                                              (concat.2)
   = []                                                    (map.1)
   LHS = RHS
   Base case holds.
```

c) Induction Step:

```
concat (map (map f) xs) = map f (concat xs)                   (hyp.1)
concat (map (map f) (x:xs)) = map f (concat (x:xs))           (Ind.1)

LHS:
concat (map (map f) (x:xs))
= concat (map f x : map (map f) xs)                           (map.2)
= foldr (++) [](map f x : map (map f) xs)                     (concat.4)
= (map f) x ++ foldr (++) [] (map (map f) xs)                 (foldr.1)
= map f x ++ foldr (++) [] (map (map f) xs)                   (associative)

RHS:
map f (concat (x:xs))
= map f (foldr (++)[] (x:xs))                                 (concat.4)
= map f (x ++ foldr (++) [] xs)                               (foldr.1)
= map f (x ++ concat xs)                                      (concat.4)
= map f x ++ map f (concat xs)                                (map.4)
= map f x ++ concat (map (map f) xs)                          (hyp.1)
= map f x ++ foldr (++) [] (map (map f) xs)                   (concat.4)

LHS = RHS
Thus this holds for the induction step.
End of Proof.
```