The University of Hong Kong

COMP3258: Functional Programming

Assignment 4

Deadline: 23:59, May 5, 2017 (HKT)

---

1 Structural Induction

Problem 1. (20 pts.) Prove that `(++)` and `length` on lists satisfy the following property.

```
length (xs ++ ys) = length xs + length ys
```

Problem 2. (20 pts.) Use the above property, now try to prove that the length of a list is equal to that of itself being reversed.

```
length (reverse l) = length l
```

The `reverse` function is defined as follows

```
reverse :: [a] -> [a]
reverse []     = []
reverse (x:xs) = reverse xs ++ [x]
```

Problem 3. (20 pts.) Prove that `(++)` and `reverse` on lists satisfy the following property.

```
reverse (xs ++ ys) = reverse ys ++ reverse xs
```

Problem 4. (20 pts.) For functions `g` and `h`, and a value `w`, if `f` and `v` satisfy the following:

```
v         = h w
f x (h y) = h (g x y)
```

then the following equation holds:

```
h . foldr g w = foldr f v
```

This is the so-called fusion property of `foldr`. Try to prove that by induction on lists.

Problem 5. (20 pts.) Prove that

```
filter p . filter q = filter (and p q)
```

where

```
and p q x = p x && p y
```

Important:

There are lots of places that may cause mark deduction. The main places are:

- Missing base cases (corner cases) will be penalized
- Missing steps in proof will be penalized.
- Unjustified steps will be penalized.
- Wrong logic will get 0 mark.
- Reduction rules that do not make sense.

Look at the solution of Tutorial 9, for example, to prove the following

```
map f (map g xs) = map (f . g) xs
```

We proceed by induction on `xs`, the proof is as follow:

```
Base case: xs = []

map f (map g xs)
= {xs = []}
map f (map g [])
= {Definition of map}
[]
= {Definition of map}
```

```
map (f . g) []
= {xs = []}
map (f . g) xs

Inductive case: xs = y:ys

map f (map g xs)
= {xs = y:ys}
map f (map g (y:ys))
= {Definition of map}
map f (g y : map g ys)
= {Definition of map}
f (g y) : map f (map g ys)
= {Induction hypothesis}
f (g y) : map (f . g) ys
= {Definition of map and .}
map (f . g) (y:ys)
= {xs = (y:ys)}
map (f . g) xs
```

Note that every `=` is accompanied by a brief explanation.

Code style and submission

You should write your answers in a file named as A4_XXX.txt, with XXX replaced by your UID. Please submit your solution on Moodle before the deadline.

Plagiarism

Please do this assignment on your own; if, for a small part of an exercise, you use something from the Internet or were advised by your classmate, please mark and attribute the source in a comment. Do not use publicly accessible code sharing websites for your assignment to avoid being suspected of plagiarism.