PMT Additional Exercises (Week 4)

Nicholas Sim

February 1st, 2018

1 Tree Reversal

[2010 Q1b (C146)] Consider the following operations on trees and lists:

```
data Tree a = Leaf a | Node (Tree a) (Tree a)

flatten :: Tree a -> [a]
flatten Leaf x = [x]
flatten (Node t1 t2) = (flatten t1) ++ (flatten t2)

rotate :: Tree a -> Tree a
rotate Leaf x = Leaf x
rotate (Node t1 t2) = Node (rotate t2) (rotate t1)

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

Prove, *using structural induction*, that:

$$\forall \texttt{t:Tree a.}\ \texttt{flatten (rotate t) = rev (flatten t)}$$

In the proofs, state what is given, the induction hypothesis (if any), what is to be shown, and justify each step. You may use the property (P), where:

```
rev (xs ++ ys) = (rev ys) ++ (rev xs)
```

2 Stronger properties

[RAP 2018 PMT tutorial W4 Q2, adapted]

```
revB :: [a] -> [a] -> [a]
revB ys [] = ys
revB ys (x:xs) = revB (x:ys) xs
```

We want to prove that `(revB [])` reverses a list, i.e.:

$$\forall \texttt{xs:[a].}\ \texttt{revB [] xs = rev xs}$$

However, this fails in the inductive step using structural induction over `xs`. Why? What alternative lemma can we prove from which we can derive this?

Bonus question: complete the proof.

3 Solution to trees

A straightforward structural induction (to prove: as in question) on trees. Note that the type a does not matter in this question; consider it fixed.

Base case To show: `flatten (rotate (Leaf x)) = rev (flatten (Leaf x))`

`flatten (rotate (Leaf x))`	
= `flatten (Leaf x)`	by def. rotate
= `[x]`	by def. flatten
= `rev [x]`	by def. rev
= `rev (flatten (Leaf x))`	by def. flatten

Inductive step Fix `t = Node t1 t2`. Inductive Hypothesis:

$$\forall \texttt{flatten (rotate t1) = rev (flatten t1)}$$

$$\wedge\forall \texttt{flatten (rotate t2) = rev (flatten t2)}$$

To show:

`flatten (rotate (Node t1 t2)) = rev (flatten (Node t1 t2))`

`flatten (rotate (Node t1 t2))`	
= `flatten (Node (rotate t2) (rotate t1))`	by def. rotate
= `(flatten (rotate t2)) ++ (flatten (rotate t1))`	by def. flatten
= `(rev (flatten t2)) ++ (rev (flatten t1))`	by I.H
= `rev (flatten t1) ++ (flatten t2)`	by (P)
= `rev (flatten (Node t1 t2))`	by def. flatten

4 Solution to stronger properties

With `xs` fixed, our would-be inductive hypothesis would look like this:

`revB [] xs = rev xs`

To show:

$\forall$`x:Integer. revB [] (x:xs) = rev (x:xs)`

`revB [] (x:xs)`	
= `revB (x:[]) xs`	by def. revB
=???	

Prove this instead (structural induction over `xs` again):

$\forall$`xs:[Integer],ys:[Integer]. revB ys xs = (rev xs) ++ ys`