Homework Assignment 4 – Solutions

October 20, 2013

1 Word sets again

```
ghci 1> let text = "Pierre Vinken , 61 years old , will join the board " ++
            "as a nonexecutive director Nov. 29 .\nMr. Vinken " ++
            "is chairman of Elsevier N.V. , the Dutch publishing group ."
```

A. Write a function to extract word sets from a text; the function should explicitly test whether a word is in the sentence or not, i.e., you shouldn't use a function that Haskell already provides to do this.

```
ghci 2> let {itemNotInList :: (Eq a) ⇒ a → [a] → Bool;
             itemNotInList x [ ] = True;
             itemNotInList x (y : ys)
                 | x ≡ y = False
                 | otherwise = itemNotInList x ys}
```

```
ghci 3> let {addItemIfNew :: (Eq a) ⇒ a → [a] → [a];
             addItemIfNew x ys =
               if itemNotInList x ys then x : ys else ys}
```

```
ghci 4> let {nub′ :: (Eq a) ⇒ [a] → [a];
             nub′ xs = foldr addItemIfNew [ ] xs}
```

B. Generate the word set for the above text using this function:

```
ghci 5> let ws_text = concat $ map words $ lines text
```

```
ghci 6> ws_text
["Pierre","Vinken",",","61","years","old",",","will","join","the","board","as",
"a","nonexecutive","director","Nov.","29",".","Mr.","Vinken","is","chairman",
"of","Elsevier","N.V.",",","the","Dutch","publishing","group","."]
```

```
ghci 7> nub' ws_text
["Pierre","61","years","old","will","join","board","as","a","nonexecutive",
"director","Nov.","29","Mr.","Vinken","is","chairman","of","Elsevier","N.V.",
",","the","Dutch","publishing","group","."]
```

```
ghci 8> let {countToken :: (Eq a,Num b) => a -> [a] -> b;
             countToken _ [] = 0;
             countToken x (y:ys)
               | x == y = 1 + countToken x ys
               | otherwise = countToken x ys
             }
```

```
ghci 9> let {countItemsInList :: (Eq a,Num b) => [a] -> [a] -> [(a,b)];
             countItemsInList [] _ = [];
             countItemsInList (x:xs) ys =
               (x,countToken x ys) : countItemsInList xs ys}
```

```
ghci 10> let {tokenCounts :: (Eq a,Num b) => [a] -> [(a,b)];
              tokenCounts xs = countItemsInList (nub' xs) xs}
```

```
ghci 11> tokenCounts ws_text
[("Pierre",1),("61",1),("years",1),("old",1),("will",1),("join",1),("board",1),
("as",1),("a",1),("nonexecutive",1),("director",1),("Nov.",1),("29",1),("Mr.",1),
("Vinken",2),("is",1),("chairman",1),("of",1),("Elsevier",1),("N.V.",1),(",",3),
("the",2),("Dutch",1),("publishing",1),("group",1),(".",2)]
```

C. Now write a function to extract word sets from a text using the data structures and functions in the *Data.Set* module

```
ghci 12> import qualified Data.Set as S
```

```
ghci 13> let setNub xs = S.toList $ S.fromList xs
```

D. Generate the word set for the above text using this function

ghci 14> *setNub ws_text*

```
[",",".","29","61","Dutch","Elsevier","Mr.","N.V.","Nov.","Pierre","Vinken",
"a","as","board","chairman","director","group","is","join","nonexecutive",
"of","old","publishing","the","will","years"]
```