Computational Semantics and Pragmatics 2012

Raquel Fernández – ILLC, University of Amsterdam

Homework #1b Due: 12/11/2012, 10:00AM

Select four out of the six exercises below. If you already knew some functional programming before, please do not select the easier exercises. Submit a Haskell file with the code for all exercises you selected. Include a couple of sample runs for each exercise as comments within the same Haskell file.

Exercise 1. Give your own definition of the function `length`, which computes the length of a string. Call it `len` to distinguish it from the built-in function `length`.

Exercise 2. Do exercise 3.12 from chapter 3 of *Computational Semantics with Functional Programming*.

Exercise 3. Do exercise 3.13 from chapter 3 of *Computational Semantics with Functional Programming*.

Exercise 4. Do exercise 3.14 from chapter 3 of *Computational Semantics with Functional Programming*.

Exercise 5. Define a function that takes a text (given in the form of a string) and returns those words in the text that contain two consecutive vowels (such as the words *'moon'* or *'break'*). Apply your function to Shakespeare's sonnets 18 and 73. The text of the sonnets is included in the Haskell file FPH.hs of *Computational Semantics with Functional Programming*.

Exercise 6. Implement a boolean function `most` that takes a list and a property as input and returns the truth value `True` if and only if more than half of the elements in the list have the given property. Example:

```
most [1,2,3,4,5,6,7] (>3)
True
```