Programming Paradigms 2024
Session 2: First steps in Haskell

Problems for solving and discussing

Hans Hüttel

24 September 2024

1 Problems that everyone should solve during the session

1. Use the functions in Section 2.4 to define a function allbutsecond that will, when given a list list , return the list consists of all but the second element of the list. As examples of what this function should do, we expect that

   ```
   allbutsecond [1,4,5,6]
   ```

   should give us [1,5,6] and that

   ```
   allbutsecond ["some,-"bizarre",-"mango'']
   ```

   will give us ["some","mango"]. How can you become more certain that your solution is correct?

2. ***(Pair programming*** Use the functions in Section 2.3 to define a function midtover that will, when given a list list of length n, return a pair of two lists (list1 , list2) such that list1 consists of the first $\lfloor \frac{n}{2} \rfloor$ elements from list1 and such that list2 consists of the remaining elements. *Hint:* Please use integer division – this is the ‘div‘ function – please remember the backquotes if you want to use it as an infix operator; the prefix version is called (div).

   As examples of what this function should do, we expect that

   ```
   midtover [1,4,5,6]
   ```

   should give us [(1,4],[5,6]) and that

   ```
   midtover ["this,-"is",-"actually '', "a'',''fairly '',''long '',''
       list '']
   ```

   - - - -

   will give us (["this,-"is",-"actually "],["a ","" fairly ","" long ","" list "])

   How can you become more certain that your solution is correct?

3. Something is wrong in the following tiny piece of code. But what is wrong? Explain. Then repair the code such that it works.

```
N = a 'div' length xs
     where
       a = 10
      xs = [1,2,3,4,5]
```

More problems to solve at your own pace

a. There is a function called reverse in the Haskell prelude that allows us to reverse any list. Use reverse to give a definition of the function last that returns the last element of a given list.

b. Try to modify the definition of qsort from the file simple.hs such that it sorts the elements in *descending order* – that is,

   ```
   qsort [2,5,6,3,8]
   ```

should give us

```
[8 ,6 ,5 ,3 ,2]
```

Will the function call qsort ["kpst ",'' ding '','' bop '','' plip ''] make sense? Why/why not?

c. Suppose we changed the definition of qsort from the file simple.hs such that we replaced <= to <. What would happen then?