MA 151: Homework #5

due Friday September 16

Written problems

In each of these, simplify the expressions step-by-step to get the final value. If there is an error, say exactly what the problem is. If the function gives an infinite loop, explain in general terms what the output will be. You should show enough detail to make it clear that you know what is going on. In all cases, you should be able to check your answer by typing the expressions into GHCi.

Assume that the following definitions have been loaded into GHCi:

\[
\begin{align*}
f (x:xs) \\
| \quad x \leq 4 & = 1 + f xs \\
| \quad \text{otherwise} & = 2 \times f xs \\
\end{align*}
\]

\[
g 0 = 1 \\
g x = 2^x + g (x-1)
\]

Now evaluate:

1. \(f [3,4,5]\)
2. \(g [3,4,5]\)
3. \(f 5\)
4. \(g 5\)

Programming problems

For all these functions, use recursion. Don’t use list comprehensions.

- Write a function called summa which takes a list of Int and adds together the evens and subtracts all the odds. For example:
  \[\text{summa} [4,1,4,6,9] \text{ is } 4 - 1 + 4 + 6 - 9 = 4.\]

- Write a function called sticker which takes a list of strings and appends them all together. (This is the same as the prelude function concat.) For example:
  \[\text{sticker} ["sometimes","i","talk","too","fast"] \text{ is } "sometimes i talk too fast"\]

- Write a function called wordList which takes a big string with spaces and splits it into a list of the strings which were separated by the spaces. (There is a similar prelude function called words.) For example:
  \[\text{wordList} "take off every zig" \text{ is } ["take", "off", "every", "zig"]\]
  (Hint: make two helping functions which either take all initial non-space characters, or drop all initial non-space characters.)
• Write a function called `allTrue` which takes a list of Bools and returns `True` if everything in the list is `True`, and `False` otherwise. Make your definition recursive (don’t just do something like `not (elem False l)`.)

• Write a function called `parsel` which doubles any lowercase 's' in a string. For example:
  
parsel "esther sleeps soundly" is "essther ssleepss ssoundly"