## CS350, 2022-23: Homework 1

August 22, 2022

## Contents

1. [Lists] Write a Haskell function TakeAlternate which takes two arguments. The first argument is a natural number N and the second is a list of type [a]. It should return a list of N elements from the positions  $0, 2, \ldots, 2N$  if the list is sufficiently long.

In cases where the list is too short, it should return as many elements as it can. [10]

2. [Lists] Write a Haskell function Last which takes a list of type [a] and returns the last element.

Handle empty lists correctly. [10]

- [Lists] Write a Haskell function Merge of type [a] → [a] → [a] which merges two sorted lists and produces a merged sorted list with no duplicates. (all in the same ordering) [10]
- 4. a. [Lists] Write a Haskell function Zip of type  $[a] \rightarrow [b] \rightarrow [(a,b)]$ which takes two lists of equal length and produces a list of tuples - the first element from each tuple comes from the first list, and the second comes from the second list. The output obeys the input ordering. [10]

b. [Higher Order Programming] Write a Haskell function ZipWith of type  $(a \to b \to c) \to [a] \to [b] \to [c]$  which takes a function f of type  $(a \to b \to c)$ , and two lists  $[x_1, x_2, \ldots]$  and  $[y_1, y_2, \ldots]$  and produces the list  $[(fx_1y_1), (fx_2y_2), \ldots]$ . [10]

[Higher Order Programming] Write a right-associative fold, called foldR.
[10]

Implement map using foldR. [10]

- 6. Produce an infinite stream of numbers which are multiples of 2, 3 or 5, in strictly ascending order,
  - a. using list comprehension [10]
  - b. using self-referential streams [15]