1 foldl

1. What is the type of the function foldl (1 point)

2. Give the de\_nition of the function foldl (1 point)

2 Tupling dropWhile and takeWhile

(2 points)

The function dropWhile :: (a ! Bool ) ! [a ] ! [a ] drops the longest pre\_x of a list in which all

elements obey the passed predicate, whereas takeWhile ::(a ! Bool ) ! [a ] ! [a ] returns the part that

is not returned by dropWhile. Write a function takeAndDropWhile :: (a ! Bool ) ! [a ] ! ([a ]; [a ])

which computes the combined result of this function and applies the predicate only once to elements

of the list.

3 Type inference

What is the type of the following expressions:

1. foldr map (1 point)

2. map foldr (1 point)

4 IO

(2 points)

Write a function table :: IO () which reads a number (say 4) from the terminal (you may use the

function read : String ! Int), and which prints the lines "1\*4 = 4" upto "10\*4=40".

5 List based functions

1. Write the function splits :: [a ] ! [(a; [a ])] which splits a list into all possible combinations of a

single element and the rest of the elements. (1 point)

2. Use this fuction to write a function permute which returns all possible permutations of a list.

(1 point)