1999 Paper 13 Question 10

Introduction to Functional Programming

Define the higher order function `foldl` ("fold left") such that

`foldl (op+) (0,[1,2,3])` evaluates to `6`, and
`foldl (op^) ("doh",["ray","me"])` evaluates to `"dohrayme"`

(Recall that `^` is a function to concatenate two strings.) [3 marks]

Define the higher order function `foldr` ("fold right") such that

`foldr (op+) (0,[1,2,3])` evaluates to `6`, and
`foldr (op^) ("doh",["ray","me"])` evaluates to `"raymedoh"`

[3 marks]

Use either `foldl` or `foldr` to write the following functions.

(*a*) The function `append`, such that the expression

```
append([1,2],[3,4])
```

evaluates to `[1,2,3,4]`. [2 marks]

(*b*) The function `length`, such that the expression

```
length([1,6,9,15])
```

evaluates to `4`. [4 marks]

(*c*) The function `map`, such that the expression

```
map (fn x => x+1) [1,2,3]
```

evaluates to `[2,3,4]`. [4 marks]

In some cases, `foldl` and `foldr` can be interchanged, i.e. the theorem

`foldl f (e,xs)` = `foldr f (e,xs)`

holds. Assuming that the list `xs` is finite, give two conditions concerning `f` and `e` that are sufficient for this theorem to be true. [Hint: you may find it helpful to consider the expansion of the expressions `foldl (op+) (0,[1,2,3])` and `foldr (op+) (0,[1,2,3])`.] [4 marks]