99 questions/Solutions/13

From HaskellWiki

< 99 questions | Solutions

(**) Run-length encoding of a list (direct solution).

Implement the so-called run-length encoding data compression method directly. I.e. don't explicitly create the sublists containing the duplicates, as in problem 9, but only count them. As in problem P11, simplify the result list by replacing the singleton lists (1 X) by X.

```
encode' :: Eq a => [a] -> [(Int,a)]
encode' = foldr helper []
    where
      helper x [] = [(1,x)]
      helper x (y@(a,b):ys)
        | x == b    = (1+a,x):ys
        | otherwise = (1,x):y:ys

encodeDirect :: Eq a => [a] -> [ListItem a]
encodeDirect = map encodeHelper . encode'
    where
      encodeHelper (1,x) = Single x
      encodeHelper (n,x) = Multiple n x
```

First of all we could rewrite the function `encode` from problem 10 in a way that is does not create the sublists. Thus, I decided to traverse the original list from right to left (using **`foldr`**) and to prepend each element to the resulting list in the proper way. Thereafter we only need to modify the function `encodeModified` from problem 11 to use `encode'`.

Alternative direct writing without significant external functions:

```
encodeDirect :: (Eq a) => [a] -> [ListItem a]
encodeDirect [] = []
encodeDirect (x:xs) = encodeDirect' 1 x xs
encodeDirect' n y [] = [encodeElement n y]
encodeDirect' n y (x:xs) | y == x    = encodeDirect' (n+1) y xs
                         | otherwise = encodeElement n y : (encodeDirect' 1 x xs)
encodeElement 1 y = Single y
encodeElement n y = Multiple n y
```

Yet another solution:

```
encodeDirect :: (Eq a)=> [a] -> [ListItem a]
encodeDirect [] = []
encodeDirect (x:xs)
    | count==1  = (Single x) : (encodeDirect xs)
    | otherwise = (Multiple count x) : (encodeDirect rest)
    where
        (matched, rest) = span (==x) xs
        count = 1 + (length matched)
```

As a wrapper, with a helper:

```
encodeDirect :: Eq a => [a] -> [ListItem a]
encodeDirect []=[]
encodeDirect (x:xs) = encodeDirectHelper 1 x xs

encodeDirectHelper :: Eq a => Int->a->[a]->[ListItem a]
encodeDirectHelper n x [] = [encodeHelper(n,x)]
encodeDirectHelper n x xs = if x==(head xs)
            then encodeDirectHelper (n+1) x (tail xs)
            else [encodeHelper(n,x)] ++ (encodeDirect xs)

encodeHelper :: (Int, a)-> ListItem a
encodeHelper (1,x)= Single x
encodeHelper (n,x)= Multiple n x
```

Retrieved from "https://wiki.haskell.org/index.php?title=99_questions/Solutions/13&oldid=57421"

Category:

- Programming exercise spoilers

- This page was last modified on 18 January 2014, at 19:32.
-