Quizlet 1: Haskell expressions | recursions

Kevin Liang Ling 185A Due: 08/09/2024, 11:59 PM PDT

Your name: Ricardo Varela Tellez

Total: 20 points

1 Haskell expressions

1. Evaluate the following expressions step by step in the same way as we did in class.

(a) `((\x -> (\y -> x + (2 * y))) 5) 1` 2 points

(\y → 5 + (2*y)) 1

5 + (2*1)

5+2

7

(b) `(\x -> (\y -> x + (2 * y)) 5) 1` 2 points

(\y → 1 + (2*y)) 5

1+ (2*5)

1+10

11

2. Construct a closed expression whose value depends on a shape represented by the variable s and a number represented by the variable n, such that: 6 points

- if the shape s is `Rock`, then this expression evaluates to the number n; and
- if the shape s is `Paper`, then this expression evaluates to the number n plus 1; and
- if the shape s is `Scissors`, then expression evaluates to the number n plus 2.

Shape s
number n

data Shape = Rock | Paper | scissors deriving Show

case s of {Rock → n , Paper → n+1 ; scissors → n+2}

2 Recursion

Not built-in type

1. Assume the type IntList from lecture (Section 2.3). Write a recursive function isLengthEven::IntList -> Bool that returns True if the list contains an even number of integers, and False otherwise. If you want, you can write it in Haskell syntax; but if you want to take some time to get more comfortable working with Haskell, you can also use other ways of representing the same idea (e.g., write a recursive function in mathematical terms).

10 points

```
data Intlist = Empty | NonEmpty Int IntList deriving Show
isLengthEven:: IntList → Bool
isLengthEven Empty = True
isLengthEven (NonEmpty x Empty) = False
isLengthEven (NonEmpty x rest) = not (isLengthEven rest)
```

peel off first element

Check: [1,2,3]

NonEmpty 1(NonEmpty 2 (NonEmpty 3 Empty)) false ✓

isLengthEven (NonEmpty x rest) = not (isLen~~gthEven~~ true 2,3)

isLengthEven (NonEmpty x rest) = not (isLen~~gthEven~~ false 3)

isLengthEven (NonEmpty x Empty) = False

[1,2]

NonEmpty 1 (NonEmpty 2 Empty) True ✓

isLengthEven (NonEmpty x rest) = not (isLen~~gthEven~~ false 2)

isLengthEven (NonEmpty x Empty) = False