

Programming Paradigms 2024

Session 8 : Catching up

Hans Hüttel

5 November 2024

Our plan for today

1. Problem no. 1
2. Problem no. 2
3. Problem no. 3
4. Break
5. Problem no. 4
6. Problem no. 5
7. We evaluate today's session – please stay until the end!

Problem 1 - From types to expressions (15 minutes)

Find Haskell expressions that have the types

1. $(\text{Ord } a, \text{Num } a) \Rightarrow a \rightarrow a \rightarrow [[\text{Bool}]] \rightarrow \text{Bool}$
2. $\text{Num } a \Rightarrow (t \rightarrow a, t) \rightarrow a \rightarrow a$
3. $\text{Fractional } t1 \Rightarrow (t2 \rightarrow t1) \rightarrow (t2 \rightarrow t1) \rightarrow (t1 \rightarrow t3) \rightarrow t2 \rightarrow t3$

Problem 2 – Triples (20 minutes)

A former minister of science and education now wants to get a masters degree and is learning Haskell. The minister is trying to construct a function `triples` that takes a list of tuples (each tuple has exactly 3 elements) and converts that list of tuples into a tuple of lists.

`triples [(1,2,3) , (4, 5, 6), (7, 8, 9)]` should produce
([1,4,7], [2, 5, 8], [3, 6, 9]).

The minister wrote the following piece of code and a type specification but ran into problems. What seems to be wrong?

```
triples :: Num a => [(a,a,a)] -> ([a],[a],[a])
triples [] = ()
triples [(a,b,c)] = ([a],[b],[c])
triples (x:xs,y:ys,z:zs) = [x,y,z] : Triples
    [(xs,ys,zs)]
```

Problem 3 - Declaring your own type class (20 minutes)

From linear algebra we know that a vector space with inner product is one for which the operations of vector sum and dot product are defined. Given two vectors v_1 and v_2 , the sum $v_1 + v_2$ is again a vector, and the inner product $v_1 \cdot v_2$ is a number. *Please note:* In linear algebra there is much more to the definition of inner product spaces than these two requirements, but in this problem, please ignore that. Also assume that the inner product is a number of type `Int`.

Define a typeclass `InVector` whose instances are types that can be seen as inner product spaces, where vector sum is called `&&&` and inner product is called `***`. *Hint:* Which section in the text for today do you need here?

Find out how to declare `Bool` as an instance of `InVector`. *Hint:* You have to find a definition of vector sum and inner product for truth values.

Problem 4 – Frequencies (20 minutes)

The goal of this problem is to define a function `frequencies` that, given a string `s`, creates a list of pairs `[(x1,f1) ,....(xk,fk)]` such that if the character `xi` occurs a total number of `fi` times throughout the list `s`, then the list of pairs will contain the pair `(xi, fi)`.

As an example of this,

```
frequencies "regn timer"
```

should return the list

```
[( 'r' ,2) ,( 'e' ,2) ,( 'g' ,2) ,( 'n' ,2) ,( 'i' ,1) ]
```

1. What should the type of the function be?
2. Use *recursion* to give a definition of `frequencies`.

Problem 5 – Encyclopedias again (20 minutes)

The type `Encyclopedia` is given by the definition

```
data Encyclopedia a = Entry String a [
    Encyclopedia a]
```

An encyclopedia is *layered* if it holds that all values at the same level of the encyclopedia are larger than the values in the levels above. As an example, t_2 shown in the problem set is layered, since 8 and 9 at level 3 are greater than the values 3, 4 and 5 at level 2 – which are greater than the value 1 at level 1.

Define a function `layered` that can tell us if an encyclopedia is layered. *Hint:* The higher-order functions `all` and `map` are useful.

Evaluation

Did this help? What else can we do?