Publié à l'aide de Google Docs
Signaler un abus
En savoir plus

Harry Potter
y el Parcial de Funcional

Se pide desarrollar un programa Haskell que ayude a regular el consumo de las pociones que se enseñan a los alumnos del colegio Hogwarts de Magia y Hechicería.

Las pociones, producidas al combinar ingredientes exóticos que causan efectos diversos, pueden ser consumidas con el fin de alterar los niveles de suerte, inteligencia y fuerza de quien las bebe.

Para representar este modelo contamos con las siguientes definiciones:

```
data Persona = Persona {
  nombrePersona :: String,
  suerte :: Int,
  inteligencia :: Int,
  fuerza :: Int
} deriving (Show, Eq)

data Pocion = Pocion {
  nombrePocion :: String,
  ingredientes :: [Ingrediente]
}

type Efecto = Persona -> Persona

data Ingrediente = Ingrediente {
  nombreIngrediente :: String,
  efectos :: [Efecto]
}

nombresDeIngredientesProhibidos = [
 "sangre de unicornio",
 "veneno de basilisco",
 "patas de cabra",
 "efedrina"]

maximoSegun :: Ord b => (a -> b) -> [a] -> a
maximoSegun _ [ x ] = x
maximoSegun  f ( x : y : xs)
  | f x > f y = maximoSegun f (x:xs)
  | otherwise = maximoSegun f (y:xs)
```

Se pide resolver los siguientes puntos utilizando los conceptos aprendidos del paradigma funcional: composición, aplicación parcial y orden superior.

1. Dada una persona definir las siguientes funciones para cuantificar sus niveles de suerte, inteligencia y fuerza **sin repetir código**:
    a. **sumaDeNiveles** que suma todos sus niveles.
    b. **diferenciaDeNiveles** es la diferencia entre el nivel más alto y más bajo.
    c. **nivelesMayoresA** n, que indica la cantidad de niveles de la persona que están por encima del valor dado.

2. Definir la función **efectosDePocion** que dada una poción devuelve una lista con los efectos de todos sus ingredientes.

3. Dada una lista de pociones, consultar:
    a. Los **nombres de las pociones hardcore**, que son las que tienen al menos 4 efectos.
    b. La **cantidad de pociones prohibidas**, que son aquellas que tienen algún ingrediente cuyo nombre figura en la lista de ingredientes prohibidos.
    c. Si **son todas dulces**, lo cual ocurre cuando todas las pociones de la lista tienen algún ingrediente llamado “azúcar”.

4. Definir la función **tomarPocion** que recibe una poción y una persona, y devuelve como quedaría la persona después de tomar la poción. Cuando una persona toma una poción, se aplican todos los efectos de esta última, en orden.

5. Definir la función **esAntidotoDe** que recibe dos pociones y una persona, y dice si tomar la segunda poción revierte los cambios que se producen en la persona al tomar la primera.

6. Definir la función **personaMasAfectada** que recibe una poción, una función cuantificadora (es decir, una función que dada una persona retorna un número) y una lista de personas, y devuelve a la persona de la lista que hace máxima el valor del cuantificador. Mostrar un ejemplo de uso utilizando los cuantificadores definidos en el punto 1.