

CI4251 - Programación Funcional Avanzada
Tarea 1

Ernesto Hernández-Novich
86-17791
<emhn@usb.ve>

Mayo 6, 2010

1. Monoids (1 punto)

Estudie la definición del *typeclass* `Monoid`, provea la instanciación del tipo `[a]` como un `Monoid` y escriba al menos tres ejemplos de uso de las funciones estándar. ¿Que representa cada función del `Monoid`?

2. Monad Writer (2 puntos)

Estudie las definiciones de los *typeclass* asociados a la construcción del `Monad Writer` y provea la implantación estándar de la clase `Monad` justificando las funciones `return` y `>>=`, así como la implantación estándar de la clase `MonadWriter` justificando las funciones `pass`, `listen` y `tell`.

Note que el *typeclass* utiliza un `Monoid` para el registro de “eventos” y por lo tanto resultaría natural emplear una lista para ello. ¿Por qué es una **mala idea** desde el punto de vista de la eficiencia el empleo de listas?

3. Uso de Monad State (2 puntos)

Se tiene un archivo de texto de longitud desconocida del cual debe seleccionarse una línea al azar, con probabilidad uniforme. Esto es, si el archivo tiene n líneas, debe escogerse una cualquiera con probabilidad $1/n$. Sin embargo, como el archivo puede ser muy largo, se desea recorrer el archivo **una sola vez**.

Hay un algoritmo muy simple (y justo) para resolver este problema: la k -ésima línea se selecciona con probabilidad $1/k$. Escriba un programa Haskell que reciba el nombre de un archivo como un argumento de la línea de comandos, lo abra, lo procese línea por línea recorriéndolo **una sola vez**, y una vez procesado todo el archivo muestre la línea seleccionada. El proceso de selección de la línea debe hacerse con estilo monádico, usando el algoritmo descrito. **No** implante el `Monad State` manualmente; simplemente diseñe el tipo de datos para representar el estado mutable y aproveche `Control.Monad.State`.