

Programación Declarativa 2016-2

Práctica 1: HASKELL^{*}

Selene Linares Arévalo

Lourdes González Huesca

7 de febrero de 2016
Facultad de Ciencias UNAM

Instrucciones:

Para cada una de las siguientes secciones y cada mini-proyecto, utiliza la agrupación de código mediante módulos: `module SeccionFoo where`

Un archivo con los módulos correspondientes debe llamarse `practica01Apellido.hs` donde el apellido de identificación es de algún miembro del equipo. La primer línea del archivo debe contener el nombre de cada miembro del equipo.

El archivo debe ser enviado a `luglzhuesca@ciencias.unam.mx` a más tardar el **viernes 19 de febrero de 2016**.

Booleanos

1. Las operaciones booleanas posibles entre dos valores booleanos son 16. En el preludio de HASKELL están definidas: `&&` (and), `||` (or) y `not`. Define el resto de las operaciones.

Números

1. Define la función `max` que devuelve el número mayor dados dos números.
2. Define la función `maxthree` que devuelve el número mayor dados tres números.
3. Define una función que calcule el promedio de tres números.
4. Define la función `gtaveragethree` que recibe tres enteros y devuelve el número que es mayor al promedio de los tres.

Listas

1. Define una función `atN` que toma una lista y un número `n` y devuelve el elemento en la n -ésima posición de la lista.

^{*}<http://wiki.haskell.org/Haskell>

<http://www.haskell.org/tutorial/index.html>

2. Da la definición de una función que selecciona los k primeros elementos de una lista, donde k es el elemento más pequeño de la lista.
3. Define una función que dada una lista $[a_0, a_1, a_2, \dots, a_m, a_n, a_o, a_p]$ devuelva una lista de pares cuyos elementos son (a_0, a_p) (a_1, a_o) (a_2, a_n) \dots ¿Cómo tratar el caso en que la lista dada tenga longitud impar? Incluye esta observación en tu implementación.

Listas por comprensión

1. Reescribe las siguientes listas como listas por comprensión:

- a) $[0, 1, 3, 7, 15, 31, 63, \dots]$
- b) $[(3, 4), (7, 8), (11, 12), (15, 16), \dots]$
- c) $[51, 58, 65, 72, 79, 86, 93, 100]$
- d) ternas pitagóricas

Funciones de orden superior

1. Define una función que devuelva una lista con los n primeros números de tribonacci iniciando con 1, 1, 1.

Mini-proyectos

Para cada mini-proyecto se debe incluir documentación que explique las funciones y las ideas o soluciones implementadas.

Conjuntos Considera la siguiente definición que caracteriza conjuntos:

```
data Set a = ESet | CSet a (Set a) deriving (Read, Show)
```

Define las siguientes funciones: `listToSet`, `SettoList`, `isEmpty`, `inSet`, `addtoSet`, `unionS`, `intersectionS`, `diffS`, `subSets`, `equalsS`.

Árboles Proporciona una definición de tipo de dato para los árboles binarios que contienen información sólo en las hojas. Incluye las definiciones de las siguientes funciones: `addleaf`, `leafs`, `depth`, `size`, `mapA`.