

Facultad de Ciencias UNAM

Lógica Computacional

Práctica 1

Profesor: Francisco Hernández Quiroz

Ayudante: Valeria Garcia Landa

Ayudante de laboratorio: Sara Doris Montes Incin

Entrega: 7 de febrero de 2020 antes de las 11:59 p.m

1 Definiciones

Consideremos las siguientes definiciones:

```
data Natural = Cero | Suc Natural deriving Show
```

```
data ListaNat = Nil | Cons Natural ListaNat
```

```
data BTree a = Void | Node (BTree a) a (BTree a) deriving Show
```

```
data ListaSnoc a = Empty | Snoc (ListaSnoc a) a deriving Show
```

2 Ejercicios

1. Implementa una función que dados dos naturales nos dice si el primero es mayor que el segundo.

Firma de la función:

```
mayorQue :: Natural -> Natural -> Bool
```

2. Implementa una función que reste dos naturales. Es decir, restaNat(a,b) = a-b, teniendo cuidado que b < a.

Firma de la función:

`restaNat :: Natural -> Natural -> Natural`

3. Implementa una función que multiplique dos números naturales.

Firma de la función:

`mulNat :: Natural -> Natural -> Natural`

4. Implementa una función que regrese la reversa de una lista

Firma de la función:

`reversa :: ListaNat -> ListaNat`

5. Implementa una función que dadas dos listas de naturales regrese la concatenación de ambas.

Firma de la función:

`concatena :: ListaNat -> ListaNat -> ListaNat`

6. Función recursiva que indica si un elemento pertenece a una lista.

7. Implementa una función que convierta un árbol binario en una lista utilizando un recorrido inorden.

Firma de la función:

`inOrden :: BTree a -> [a]`

8. Implementa una función que agregue un elemento e a un árbol ordenado, preservando el orden.

Firma de la función:

`agregaOrden :: (Ord a) => a -> BTree a -> BTree a`

9. Implementa una función que dada una lista snoc devuelve la lista sin el primer elemento,

Firma de la función:

`tailSnoc :: ListaSnoc a -> ListaSnoc a`

10. Función que implementa mapeo sobre listas snoc. Firma de la función:

`MapSnoc :: (a -> b) -> ListaSnoc a -> ListaSnoc b`

3 Puntos Extra

1. Implementar una función recursiva que devuelva la longitud de un número entero. No es válido hacer conversiones a cadena o usar funciones que resuelvan directamente el ejercicios.

Firma de la función:

`longitud :: Int -> Int`

2. Implementar la función que regrese una lista con los n primeros números de tribonacci iniciando con 0, 0, 1. Usar map para implementar esta función.

Firma de la función:

`tribonaccies :: Int -> [Int]`

¡Éxito en su práctica 1!