

Lenguajes de Programación

Auxiliar N°5

Auxiliar: Fabián Mosso

15/04/2016

1. Defina nats como la lista infinita de los naturales
2. Implemente la función `myrepeat` que toma un elemento a y crea una lista infinita con ese elemento
3. Implemente la función `consec` que convierta una lista l en una lista de pares (N,e) donde N sea la cantidad de veces consecutivas que e se encuentra en la lista. Hint: Utilice la función `group`. Ejemplo:

```
> consec "aaabbcccaaaa"  
[(3,'a'),(2,'b'),(3,'c'),(4,'a')]
```
4. Considere la siguiente definición para arboles en Haskell: `data Tree a = Empty | Node a (Tree a) (Tree a)`
 - (a) Defina un árbol infinito de unos
 - (b) Implemente la función `takeTree :: Int -> Tree` que toma los primeros n niveles de un Tree contando desde la raíz
 - (c) Implemente la función `zipWithTree` que toma una operación para operar sobre dos arboles, similar a la función `zipWith`
 - (d) Usando `zipWithTree` defina el arbol infinito `levelTree` que posee en cada nivel su profundidad
 - (e) Usando `zipWithTree` defina el arbol infinito `doubleTree` que duplica el valor por cada nivel de altura
5. Implemente en Haskell un interprete para el lenguaje con `with` visto en las primeras semanas de clase.
 - (a) Defina el tipo de dato `WAE` para representar el AST del lenguaje.
 - (b) Defina el tipo de dato `Env` para representar el enviroment.

- (c) Defina la función `envLookup` para obtener el valor de un identificador en el `environment`
 - (d) Defina la función `interp`, que dado una expresión de tipo `WAE` y un ambiente (de tipo `Env`) retorna un número
 - (e) Extienda `WAE` a `FWAE` realizando adiciones y cambios a los tipos y funciones definidas anteriormente
6. Preguntele al auxiliar sobre la tarea.