Final Llenguatges de Programació

Grau en Enginyeria Informàtica
Temps estimat: 2h 30m

11 Juny 2012

Es valorarà l'ús que es faci de funcions d'ordre superior predefinides. Ara bé, en principi només s'han d'usar les de l'entorn Prelude, és a dir no hauria de caldre cap import. Si voleu usar una funció que requereixi import consulteu-ho abans amb el professor.

Excepte en el problema 1 podeu usar les funcions auxiliars que us calguin.

Problema 1: *Lambdes i ordre superior.* Usant el `foldl` (o el `foldr`) i lambda expressions usant `(+)` i `( ^ )` feu una funció que calculi la suma dels quadrats del elements d'una llista d'enters. Recordeu que els tipus del `foldl` i el `foldr` són

```
foldl :: (a -> b -> a) -> a -> [b] -> a
foldr :: (a -> b -> b) -> b -> [a] -> b
```

No podeu usar funcions auxiliars ni definicions recursives.

Problema 2: *Infinits.* Definiu una funció que donada una llista d'enter més grans que 0 genera la llista infinita de tots els nombres més grans o iguals que 2 que no són divisibles per cap dels elements de la llista d'entrada. Per exemple, si la llista d'entrada és `[3,7,19]` ens retorna `[2,4,5,8,10,11,13,16,17,20,...]`

Problema 3: *Robot.* Considereu la versió simple de la pràctica del Robot, amb les següents definicions:

```
data TipusTerreny = Aigua | Buit | Bloc | Forat
     deriving (Eq, Enum)

data Direccio = Nord | Sud | Est | Oest
     deriving (Eq)

obtenirTT :: Mapa TipusTerreny -> Int -> Int -> TipusTerreny
-- Donat un Mapa de TipusTerreny i dos enters ens indica el
-- TipusTerreny de la posició indicada pels dos enters.

type Posicio = (Int,Int)

data Robot = Robot Posicio Direccio CSalut
```

Noteu que, com que tenim l'operació `obtenirTT`, no ens cal conèixer la definició de `Mapa`.

Feu les següents operacions, de manera que satisfacin el que es demana a (l'apartat 2 de) la pràctica de modelat (PEF).

```
gNord :: Robot -> Robot
gSud :: Robot -> Robot
gEst :: Robot -> Robot
gOest :: Robot -> Robot
avanca :: Robot -> (Mapa TipusTerreny) -> Robot
```

Feu, a més, una funció que donada una llista de Direcció i un Robot ens retorna un Robot que mostra l'estat del robot després d'aplicar seqüencialment les orientacions indicades a la llista i aplicar avançar després de cada una d'elles.

Problema 4: *Inferència de tipus.* Cal escriure l'arbre decorat de les expressions i generar les restriccions de tipus. Resoleu-les per obtenir la solució.

1. Inferiu el tipus més general de `flip`:

   ```
   flip f x y = f y x
   ```

2. Inferiu el tipus més general de `tf`:

   ```
   tf (x:xs) f = let y = f x in
                 if y then x:(tf xs f)
                 else []
   ```

3. Assumint que `0,1:Int`, `(==):Eq a => a -> a -> Bool` i que `(+):Int -> Int -> Int`, inferiu el tipus més general de `pos`:

   ```
   pos (x:xs) y = if (x==y) then 0
                  else (pos xs y)+1
   ```

Problema 5: *Conceptes de llenguatges de programació.*

1. Què compleix el llenguatge `C++`?
   a) és dèbilment tipat b) és type unsafe
   c) no té tipus d) fa comprovació dinàmica de tipus

2. Indiqueu quines propietats compleix el sistema de tipus de Haskell (baseu-vos en les opcions de l'anterior pregunta, però indica correctament tot el que satisfà).

3. Indiqueu tres característiques que satisfan la majoria dels llenguatge de scripting.