Final Llenguatges de Programació

Grau en Enginyeria Informàtica
Temps estimat: 2h 45m

14 Febrer 2013

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 (1,5 punts): *Arbres generals i ordre superior.* Considereu la següent definició per arbres generals:

```
data Arbre a = Arbre a [Arbre a]
```

Apartat a) Usant el `map` feu una funció que donat un arbre d'enters ens retorna l'arbre resultant de sumar 1 a tots els enters de l'arbre. No podeu usar cap funció auxiliar (no predefinida) i la definició ha de ser en una línia completant el següent

```
inc (Arbre x la) = (Arbre ...
```

Recordeu que el tipus del `map` és `map :: (a -> b) -> [a] -> [b]`

Apartat b) Usant el `foldl` (o el `foldr`) i lambda expressions feu una funció que calculi el nombre de nodes de l'arbre. No podeu usar cap funció auxiliar (no predefinida) i la definició ha de ser en una línia completant el següent

```
mida (Arbre x la) = 1+(fold...
```

Recordeu que els tipus del `foldl` i el `foldr` són

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

Problema 2 (1,5 punts): *Sumes sèries.* Definiu una funció `infsum` que donat una enter més gran que 0 genera la llista infinita estrictament creixent d'enters que són suma d'una sèrie de números consecutius d'x. Per exemple `(infsum 3)` retorna

```
[3,7,12,18,25,33,42,52,...
```

Problema 3 (2 punts): *Robot Pintor.* Considereu la versió simple de la pràctica del robot pintor, amb les següents definicions:

```
teColor :: Mapa -> (Int,Int) -> Bool
-- que ens diu si una posició del mapa té color (no negre)

pintar :: Mapa -> (Int,Int) -> Color -> Mapa
-- que ens retorna el mapa resultant de pintar la posició indicada amb el color indicat
```

```
data Direccio = Nord | Sud | Est | Oest
     deriving (Eq)
data Color = Negre | Blanc | Vermell | Blau
     deriving (Eq)
data Accio = Pinta Color | Res
-- el robot o pinta o no fa res (no existeix l'acció d'esborrar)

data Robot = Robot (Int,Int) Direccio Accio
```

Noteu que, amb les operacions que tenim no ens cal conèixer la definició de `Mapa`.
Apartat a) Feu l'operació `avanca:: Robot -> Mapa -> (Robot,Mapa)`, de manera que satisfaci les restriccions de la pràctica de modelat (PEF).
Apartat b) Feu una funció que donada una llista de parells (Direccio,Color), un Robot i un Mapa ens retorna una parella (Robot,Mapa) resultant d'avançar (amb acció pintar) en la direcció i color de cada element de la llista (posició a posició) mentre no aparegui el color Negre o s'acabi la llista.

Problema 4 (4 punts): *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 `fun1`:

   ```
   fun1 f x y = f (x,y)
   ```

2. Assumint que `sum :: Num a => [a] -> a` i `(+) :: Num a => a -> a -> a`, inferiu el tipus més general de `fun2`:

   ```
   fun2 (x:l) = let s= (sum x) in s+ fun2 l
   ```

3. Assumint que `0:Int` i que `(==):Eq a => a -> a -> Bool`,
   inferiu el tipus més general de `fun3`:

   ```
   fun3 f (x:xs) = if (f x) == 0 then x else (fun3 f xs)
   ```

Els tipus de `(:)` i `(,)` són els habituals.

Problema 5 (1 punt): *Conceptes de llenguatges de programació.*

1. Com sol ser el sistema de tipus dels llenguatges de scripting? (indiqueu si és cert o fals)
   a) són dèbilment tipats x b) són type safe
   c) no tenen tipus d) fan comprovació estàtica de tipus

2. Indiqueu les propietats del sistema de tipus del llenguatge que us va tocar en el Treball Dirigit (TD) de Competències Transversals.

3. Indiqueu un exemple en Haskell que acaba gràcies a que s'usa *lazy evaluation* i expliqueu les raons.