

Final Llenguatges de Programació

Grau en Enginyeria Informàtica

Temps estimat: 2h 45m

17 de Juny de 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. Si l'enunciat no ho impedeix, podeu usar les funcions auxiliars que us calguin.

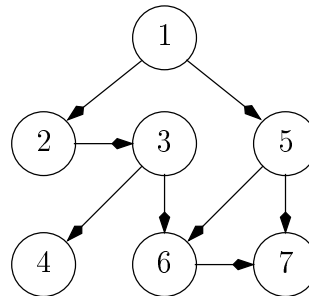
Problema 1 (1,5 punts): *Divisors no repetits.*

Definiu una funció que donada una llista de nombres primers generi en ordre creixent tots el nombres positius que només són divisibles per nombres primers que pertanyen a la llista. Per exemple, per la llista [3,5,7] el resultat és la llista infinita [1,3,5,7,9,15,21,25,...].

Problema 2 (2.5 punts): *Graphs llistes.*

Apartat a). Definiu el tipus Graph dirigit amb llistes d'adjacència seguint l'exemple i feu una operació que, donat un graph dirigit acíclic i un element, ens retorni la seva accessibilitat (en qualsevol ordre i sense repetits), és a dir la llista amb tots els nodes als que pot arribar amb 0 o més passos.

```
gr1 = Graph [(1,[2,5]),
             (3,[6,4]),
             (2,[3]),
             (4,[]),
             (6,[7]),
             (5,[6,7]),
             (7,[]) ]
```



Per exemple, per 3, la llista d'accessibilitat és [3,6,4,7], i la del 1 és [1,2,5,3,6,4,7].

Important: el graph pot ser de qualsevol cosa, no només d'enters. Indiqueu el tipus més general de la funció.

Apartat b). Modifiqueu el tipus per a que es puguin afegir pesos a les arestes. Podeu assumir que els pesos són enters. Feu una operació que donat un node i un pes i un graf dirigit amb pesos, ens retorna la llista dels nodes accessibles sense superar el pes donat. El pes d'un camí és la suma dels pesos de les arestes per les que es passa.

Problema 3 (1.5 punts): *Robot Pintor 2.* Considereu la versió 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)
dimensions :: Mapa -> (Int,Int)
-- que retorna les dimensions del mapa
```

```

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

data Direccio = Amunt | Avall | Esquerra | Dreta
    deriving (Eq)
data Color = Blanc | Groc | Blau | Vermell
    deriving (Eq)

```

Noteu que, amb les operacions que tenim no ens cal conèixer la definició de **Mapa**. **No hi ha cartutxos al mapa**, però sí que el robot té una certa quantitat de tinta de cada color, com a la pràctica.

Considereu dues formes de pintar. La normal, d'un color (com a la pràctica) i la *multicolor* on pinta les caselles consecutivament amb tots els colors, començant pel Blanc i acabant pel Vermell (seguint l'ordre de la definició) i tornant al Blanc (una casella de cada color) repetidament. Si no li queda tinta del color que toca pintar avança normalment sense pintar i passa al següent color.

Doneu una definició del tipus robot, que permeti simular les dues formes de pintar (normal i multicolor), a més de la de no pintar. Feu l'operació `avanca :: Robot -> Mapa -> (Robot, Mapa)`, de manera que satisfaci les restriccions de la pràctica de modelat (PEF), amb les simplificacions i modificacions considerades en l'enunciat.

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. Assumint que `(,) :: a -> b -> (a,b)`, inferiu el tipus més general de `fun1`:

```
fun1 f x = let (y,z) = (f x) in (z,y)
```

2. Assumint que `max :: Ord a => a -> a -> a`, `null :: [a] -> Bool` i `(:) :: a -> [a] -> [a]`, inferiu el tipus més general de `fun2`:

```
fun2 f (x:l) = if (null l) then x else (max (f x) (fun2 f l))
```

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

```
fun3 (x:xs) (y:ys) = if (x == (0 :: Int)) then y:(fun3 xs ys)
                    else x:(fun3 xs ys)
```

Problema 5 (0.5 punt): Conceptes de llenguatges de programació.

1. Indiqueu les propietats del sistema de tipus de Haskell.
2. Indiqueu si és o no de scripting i quins paradigmes inclou el llenguatge que us va tocar en el Treball Dirigit (TD) de Competències Transversals. Si té extensions importants, indiqueu quins paradigmes incorporen.