

Final Llenguatges de Programació

Grau en Enginyeria Informàtica

Temps estimat: 2h i 45m

25 de juny de 2014

Es valorarà l'ús que es faci de funcions d'ordre superior predefinides i la simplicitat de la solució. Només s'han d'usar funcions de l'entorn Prelude. Si useu funcions auxiliars, cal que les implementeu.

Problema 1 (1 punt): *Pi*.

La suma de sèrie de Leibniz, $\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$, convergeix a $\frac{\pi}{4}$. Feu una funció `mpi`, que donat un enter positiu m , calcula la aproximació de π usant els m primers element de la sèrie de Leibniz. Per exemple, `mpi 10`, és $4 * (\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \frac{1}{15} + \frac{1}{17} - \frac{1}{19})$

Problema 2 (1.5 punt): *Recorregut en amplada de llistes*. Feu una funció `wlist` que donada una llista de llistes ll retorna la llista que conté el recorregut per nivells de ll . Per exemple, `wlist [[1,5,8],[2,6],[],[3,7,9],[4]]` retorna `[1,2,3,4,5,6,7,8,9]`

Problema 3 (2 punts): *Llistes amb punt d'interès*. Volem definir un nou tipus de dades genèric `DList` que representi les llistes amb punt d'interès. Per això usarem el constructor `Empty` per representar la llista buida i `DList` per representar les demés llistes. Com a exemple, la llista que conté els elements 3,6,1,2,-3,8 amb punt d'interès al 2 volem que es representi (obligatòriament) amb `DList 2 [1,6,3] [-3,8]`.

Apartat a) A més de definir el tipus, definiu les següents funcions:

- `cur`, que donada una llista (no buida) amb punt d'interès, ens retorna l'element que està en el punt d'interès.
- `hasleft` i `hasright`, que donada una llista (no buida) amb punt d'interès, ens diu si hi ha algun element a l'esquerra (o a la dreta respectivament) del punt d'interès.
- `left`, que donada una llista (no buida) amb un punt d'interès que té algun element a l'esquerra, mou el punt d'interès a l'element de l'esquerra del punt d'interès actual. Per exemple, `left (DList "va" ["et","Com"] ["la","vida","Joan"]` és `left (DList "et" ["Com"] ["va","la","vida","Joan"]`
- `right`, com l'anterior, però movent el punt d'interès a la dreta.
- `dadd`, que afegeix un element donat a l'esquerra del punt d'interès (si en té) i el posa com a nou punt d'interès. Per exemple,
`dadd 12 (DList 2 [1,6,3] [-3,8])` és `DList 12 [1,6,3] [2,-3,8]`
Si la llista és buida, el resultant conté només l'element afegit com a punt d'interès.

Apartat b) Indiqueu que `DList` és instance de la classe `Eq` on `(==)` és la igualtat de les llistes sense tenir en compte el punt d'interès, és a dir, si té els mateixos element en el mateix ordre. Per exemple, `DList 12 [1,6,3] [2,-3,8]`, `DList -3 [2,12,1,6,3] [8]` i `DList 3 [] [6,1,12,2,-3,8]` són iguals.

Problema 4 (2.5 punts): *Inferència de tipus.* Cal escriure l'arbre decorat de les expressions i generar les restriccions de tipus. Resoleu-les per obtenir la solució. Assenyaleu el resultat final amb un requadre.

1. Inferiu el tipus més general de `fun1`:

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

2. Assumint que `fst :: (a,b) -> a` `snd :: (a,b) -> b` i que `(==) :: Eq a => a -> a -> Bool`, inferiu el tipus més general de `fun2`:

```
fun2 y (x:xs) = if fst x == y then [] else (snd x):(fun2 y xs)
```

Problema 5 (2.5 punts): *Python.*

1. Feu una funció en Python que donada una llista `l` retorna una llista de parells que indica la freqüència amb la que apareixen els elements de la llista. El primer valor del parell és l'element i el segon és la freqüència. Per exemple, si la llista és `[3,5,4,7,3,4,5,1,6,7,7,11,2,1]`, el resultat ha de ser `[(1, 2), (2, 1), (3, 2), (4, 2), (5, 2), (6, 1), (7, 3), (11, 1)]` (en qualsevol ordre).
2. Considereu la següent definició de la classe `Restaurant`

```
class Restaurant:
    def __init__(self,nom):
        self.nom = nom
        self.adr = ""
    def afegir_nom(self,nom):
        self.nom = nom
    def afegir_adr(self,adr):
        self.adr = adr
```

Definiu una subclasse `TRestaurant` de la classe `Restaurant`, que permeti afegir un nombre indefinit de telèfons per a un restaurant. Heu de guardar els telèfons en un atribut de tipus llista dins de l'objecte. Per això, afegiu l'operació `afegir_tel` que rep un telèfon i l'afegeix a la llista. Modifiqueu només les operacions que calguin. La classe `TRestaurant` també ha de tenir una operació `mostrar_restaurant` que permeti mostrar tots els camps (separats per blancs) en una línia.

Problema 6 (0.5 punt): *Conceptes de llenguatges de programació.*

1. Indiqueu si el llenguatge que us va tocar en el Treball Dirigit (TD) de Competències Transversals, és compilat o interpretat i si té "garbage collection" o no.
2. Indiqueu quins paradigmes de programació admet el llenguatge que us va tocar en el Treball Dirigit (TD) de Competències Transversals.