

Parcial Llenguatges de Programació

Grau en Enginyeria Informàtica

27 Novembre 2013

Per accedir al racó aneu a <https://examens.fib.upc.es>

Cal que lliureu via racó el codi amb els comentaris que considereu necessaris en un arxiu “.hs” executable en l’entorn ghci i que satisfaci els diferents enunciats que es llisten a continuació.

Cal que al començar la solució de cada problema afegiu una línia comentada indicant el problema (i subapartat, si en té) que ve a continuació. Per exemple,

```
-- Problema 2
-- Problema 3.b
```

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 ha de caldre cap import. Podeu definir funcions auxiliars si us calen.

Problema 1: *Generació*

Feu una funció **genPairs**, que donades 3 llistes (que podeu assumir que no tenen elements repetits) retorna la llista amb tots els parells amb el primer element de la primera llista, el segon de la segona i on un i només un dels dos pertany a la tercera.

```
genPairs [2,5,6] [3,1,4] [2,5,3]
[(2,1),(2,4),(5,1),(5,4),(6,3)]
```

Es valorarà que ho feu usant una definició amb llistes per comprensió.

Problema 2: *Repetits*

Definiu una funció **nodup** que elimina els elements repetits i es queda la primera aparició de cada element.

```
nodup [3,5,4,5,3,1]
[3,5,4,1]
```

Es valorarà que useu funcions d’ordre superior predefinides y lambda expressions.

Problema 3: *Arbres infinits*

Cosidereu la següent definició del tipus arbre binari

```
data Arbre a = Node a (Arbre a) (Arbre a)
              | Abuit
```

Apartat 3.1: Afegiu a la definició d'arbre el mínim que calgui per poder mostrar els resultats (com a l'exemple final).

Apartat 3.2: Feu una funció `ttake` que donat un arbre binari *a* i un enter no negatiu *n* retorna un arbre binari que és la part superior de l'arbre *a* fins la profunditat *n* (similar al `take` de les llistes).

Apartat 3.3: Feu una funció `inftree` (que no té paràmetres) i que és un arbre infinit on totes les branques són infinites i tenen com a nodes els enters estrictament positius en ordre creixent 1, 2, 3, 4, 5, 6 ...

```
ttake 3 inftree
Node 1 (Node 2 (Node 3 Abuit Abuit) (Node 3 Abuit Abuit)) (Node 2 (Node 3 Abuit Abuit) (Node 3 Abuit Abuit))
```

Problema 4: *Llistes amb errors*

Volem representar el tipus (genèric) llista amb errors. Que inclou una llista i un nivell d'error, que és un enter (que podem assumir no negatiu). En aquest tipus, dues llistes són iguals si, comparant les llistes d'esquerra a dreta, el nombre d'elements diferents és inferior o igual als nivells d'error de les dues llistes. Noteu que si una llista és més curta que l'altre, tots els elements de diferència es consideren errors. Com a exemples tenim que, si la primera llista amb errors conté 3,1,2,5 i té nivell d'error 2 i la segona conté 3,4,2 i té nivell d'error 3, llavors són iguals, però si, amb els mateixos elements, la primera té nivell d'error 2 i la segona 1 llavors són diferents.

Definiu un nou tipus (genèric) `ErrorList` que representi les llistes amb errors i indiqueu que `ErrorList` és una *instance* de la classe `Eq` on (`==`) és la igualtat descrita anteriorment.