
Examen de Prácticas - 14 de enero de 2019
LTP (Tipo B)

ALUMNO: _____ GRUPO: _____

Instrucciones

- El alumno dispone de 60 minutos para resolver el examen.
- El examen consta de 3 preguntas que deberán responderse en el mismo enunciado, en los recuadros incluidos en cada pregunta.

Pregunta 1 – Haskell (3.25 puntos)

Considera disponible una función `member` cuyo tipo es:

```
member :: (Eq a) => a -> [a] -> Bool
```

La función `member` comprueba si un elemento está en una lista de elementos (del mismo tipo `a`, genérico, pero restringido a la clase de tipos `Eq`).

Define una función `union` cuyo tipo sea:

```
union :: (Eq a) => [a] -> [a] -> [a]
```

La función `union`, dadas dos listas de elementos del tipo `(Eq a) => a`, devuelve una lista que es la unión de las dos listas recibidas. Considera listas sin elementos repetidos.

Ejemplos de uso:

```
*Main> union [3,4,2,1,5] [6,5,2,8,1]
[3, 4, 6, 5, 2, 8, 1]
*Main> union [3,4,2] [6,5,2,8,1]
[3, 4, 6, 5, 2, 8, 1]
*Main> union [3,4,2] [6,5,8,1]
[3, 4, 2, 6, 5, 8, 1]
*Main> union ['a','b','k','h'] ['h','n','b']
"akhnb"
*Main> union [2.0, 3.1] [4.5, 3.1, 2.2]
[2.0, 4.5, 3.1, 2.2]
```

El orden de los elementos en la lista resultado no es relevante. Así, en cada ejemplo, sería un resultado válido cualquier lista que contuviera los mismos valores, pero en otro orden.

SOLUCIÓN CON RECURSIÓN DIRECTA:

```
union :: (Eq a) => [a] -> [a] -> [a]
union [] y = y
union (x:xs) y
  | member x y = union xs y
  | otherwise = x : union xs y
```

SOLUCIÓN CON LISTAS INTENSIONALES:

```
union :: (Eq a) => [a] -> [a] -> [a]
union x y = [z | z <- x, not(member z y)] ++ y
```

Pregunta 2 – Haskell (3.25 puntos)

Considera la definición del tipo de dato `BinTreeInt`:

```
data BinTreeInt = Void | Node Int BinTreeInt BinTreeInt deriving Show
```

Define una función `sizeTree` cuyo tipo sea:

```
sizeTree :: BinTreeInt -> Int
```

La función `sizeTree`, dado un árbol binario de enteros, devuelve su talla o número de enteros almacenados en los nodos del árbol.

Ejemplos de uso:

```
*Main> sizeTree Void
0
*Main> sizeTree (Node 4 Void Void)
1
*Main> sizeTree (Node 3 (Node 2 Void Void) (Node 7 (Node 4 Void Void) Void))
4
*Main> sizeTree (Node 3 (Node 2 Void Void) (Node 7 (Node 4 Void Void) (Node 9 Void Void)))
5
```

SOLUCIÓN CON RECURSIÓN DIRECTA:

```
sizeTree :: BinTreeInt -> Int
sizeTree Void = 0
sizeTree (Node x izq der) = 1 + sizeTree izq + sizeTree der
```

Pregunta 3 – Prolog (3.50 puntos)

Resolver los 2 ejercicios que se plantean, dada la siguiente base de conocimiento:

```
/* movie(M, Y), M is a movie released in the year Y */
movie(barton_fink, 1991).
movie(the_big_lebowski, 1998).
movie(fargo, 1996).
movie(lick_the_star, 1998).
movie(mission_impossible, 1996).
movie(fall, 1997).
/* director(M, D), M is a movie directed by D */
director(the_big_lebowski, joel_coen).
director(barton_fink, ethan_coen).
director(fargo, joel_coen).
director(lick_the_star, sofia_coppola).
director(mission_impossible, brian_de_palma).
/* actor(M, A, R), the actor A played the role of R in the movie M */
actor(mission_impossible, tom_cruise, ethan_hunt).
actor(mission_impossible, jon_voight, jim_phelps).
actor(barton_fink, john_turturro, barton_fink).
actor(barton_fink, john_goodman, charlie_meadows).
actor(the_big_lebowski, jeff_bridges, jeffrey_lebowski__the_dude).
actor(the_big_lebowski, john_goodman, walter_sobchak).
actor(the_big_lebowski, philip_seymour_hoffman, brandt).
actor(the_big_lebowski, john_turturro, jesus_quintana).
```

1.- Define un predicado moviesTwoActors que permita, dados dos actores, encontrar las películas en las que coincidieron. Ejemplo de uso: `moviesTwoActors(M, john_turturro, john_goodman).`
M = barton_fink ;
M = the_big_lebowski.

SOLUCIÓN:

```
moviesTwoActors(M, A1, A2) :- actor(M, A1, _), actor(M, A2, _), A1 \== A2.
```

2.- Define un predicado moviesYear que permita, dado un año, encontrar dos películas estrenadas ese año. Ejemplo de uso: `?- moviesYear(M1, M2, 1996).`
M1 = fargo ,
M2 = mission_impossible.

SOLUCIÓN:

```
moviesYear(M1, M2, Y) :- movie(M1, Y), movie(M2, Y), M1 @< M2.
```