
Examen de Prácticas - 25 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)

Define una función `operate` cuyo tipo sea:

```
operate :: (Int -> Int -> Int) -> [Int] -> [Int] -> [Int]
```

La función `operate`, dadas una función de tipo `(Int -> Int -> Int)` y dos listas de enteros de igual longitud, devuelve una lista cuyos elementos son el resultado de aplicar la función primer argumento a los enteros almacenados en las mismas posiciones de las listas segundo y tercero argumento.

Requisitos: NO se permite usar ninguna función predefinida de Haskell. Debe resolverse mediante recursión o lista intensional.

Ejemplos de uso:

```
*Main> operate (-) [10, 2, 15] [4, 5, 6]
[6, -3, 9]
*Main> operate (*) [1, 2, 3] [4, 5, 6]
[4, 10, 18]
```

SOLUCIÓN CON RECURSIÓN DIRECTA:

```
operate :: (Int -> Int -> Int) -> [Int] -> [Int] -> [Int]
operate f [] [] = []
operate f (x:xs) (y:ys) = (f x y) : operate f xs ys
```

SOLUCIÓN CON LISTAS INTENSIONALES:

```
operate :: (Int -> Int -> Int) -> [Int] -> [Int] -> [Int]
operate f xs ys = [f (xs!!i) (ys!!i) | i <- [0..length xs - 1] ]
```

Pregunta 2 – Haskell (3.25 puntos)

Considera el siguiente código, donde se define el tipo de datos **Sphere** (que representa la figura geométrica tridimensional Esfera) y la clase de tipos **Shape3D** (que representa las figuras geométricas tridimensionales, en general):

```
type Radius = Float
data Sphere = Sphere Radius
class Shape3D a where
    area    :: a -> Float
    volume  :: a -> Float
```

Se pide añadir el código necesario para que **Sphere** instancie **Shape3D**.

Ayuda: Las fórmulas para calcular el área y el volumen de una esfera de radio **r** son:

$$area = 4 \pi r^2 \qquad volume = (4/3) \pi r^3$$

Ejemplos de uso:

```
*Main> area (Sphere 2)
50.265484
*Main> volume (Sphere 2)
33.510323
```

SOLUCIÓN:

```
instance Shape3D Sphere where
    area (Sphere r) = 4 * pi * r**2
    volume (Sphere r) = (4/3) * pi * r**3
```

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(crimewave, 1985).
movie(spies_like_us, 1985).
/* director(M, D), M is a movie directed by D */
director(barton_fink, ethan_coen).
director(the_big_lebowski, joel_coen).
director(fargo, joel_coen).
director(crimewave, sam_raimi).
director(spies_like_us, john_landis).
/* actor(M, A, R), the actor A played the role of R in the movie M */
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, john_turturro, jesus_quintana).
actor(crimewave, joel_coen, reporter_at_execution).
actor(spies_like_us, joel_coen, drive_in_security).
actor(spies_like_us, sam_raimi, drive_in_security).
```

1.- Define un predicado directorActor que permita encontrar los directores que hayan sido actores en alguna película. Ejemplo de uso: `?- directorActor(D).`

```
D = joel_coen ;
D = sam_raimi.
```

SOLUCIÓN:

```
directorActor(D) :- director(_, D), actor(_, D, _).
```

2.- Define un predicado superstar que permita encontrar los actores que hayan salido en al menos dos películas. Ejemplo de uso: `?- superstar(A).`

```
A = john_turturro ;
A = john_goodman ;
A = joel_coen.
```

SOLUCIÓN:

```
superstar(A) :- actor(M1, A, _), actor(M2, A, _), M1 @< M2.
```