



PUNTUACION: **1(2.5) 2(1+1) 3(2.5) 4(1.5+1.5)**

1 Sabiendo que $id = foldr (\:) []$, demuestra la siguiente propiedad para listas

$$\forall as, bs :: [a] . foldr (\:) as bs \equiv foldr (\:) [] (bs ++ as)$$

2.A Escribe la expresión $map f (filter p xs)$ usando listas por comprensión

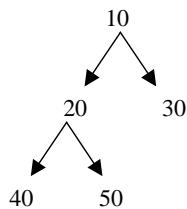
2.B Escribe la expresión $[p (fx) / x <- xs, p x]$ usando las funciones *filter* y *map*

3 Consideremos las siguientes funciones:

```
unfold p h t x
/p x      = []
/otherwise = h x : unfold p h t (t x)
desdeHasta n m = [n .. m]
```

Da un definición de la función *desdeHasta* usando *unfold*.

4 Una rama es una lista con los nodos desde la raíz de un árbol hasta una de sus hojas. Para el árbol que aparece a continuación las distintas ramas son [10,20,40], [10,20,50] y [10,30]:



4.A Para el tipo

```
data ÁrbolB a = VacíoB | NodoB (ÁrbolB a) a (ÁrbolB a) deriving Show
```

define una función *ramas* :: *ÁrbolB a* -> *[a]* que devuelva todas las ramas de un árbol binario.

4.B Resuelve el mismo problema pero para un árbol general:

```
data Árbol a = Vacío | Nodo a [Árbol a] deriving Show
```