

Práctico 10: Inducción - Derivación

***Ejercicio 1.** Demostrar que la concatenación de listas es asociativa (dar la definición de $\#$)

- $(xs \# ys) \# zs = xs \# (ys \# zs)$

Ejercicio 2. Demostrar la siguiente propiedad (definir map y \circ)

- $\text{map } (f \circ g) = (\text{map } f) \circ (\text{map } g)$

Ejercicio 3. Demostrar la siguiente propiedad (definir reversa y $++$)

- $\text{reversa } (xs \# ys) = \text{reversa } ys \# \text{reversa } xs$

***Ejercicio 4.** Demostrar la siguiente propiedad (definir reversa y $++$)

- $\text{reversa } (\text{reversa } xs) = xs$

***Ejercicio 5.** Demostrar la siguiente propiedad (definir lenght)

- $\text{lenght } (xs \# ys) = \text{lenght } xs + \text{lenght } ys$

Ejercicio 6. Especificar y derivar la siguiente función.

- $f \text{ xs}$ dice si todos los elementos son iguales.

***Ejercicio 7.** Derivar una función que dada una lista determina si los elementos de esta lista están ordenados de forma creciente.

Ejercicio 8. Sea $m : [Num] \rightarrow Num$ una función que devuelve el mínimo de una lista dada. Especificar y derivar m .

Ejercicio 9. Especificar y derivar una función que dada una lista determina si existe un elemento en ella que sea igual a la suma del resto de los elementos de la lista.

Ejercicio 10. Derivar un programa usando la siguiente especificación:

$$P.xs.ys = \langle \exists as, bs :: ys = as \# xs \# bs \rangle,$$

que dadas dos listas determina si la primera es subsegmento de la segunda.

Ejercicio 11. Calcular la cantidad de números pares e impares de una lista dada, recorriendo la lista una sola vez (Ayuda: utilizar tuplas.)

***Ejercicio 12.** Implementar todas las funciones obtenidas de las derivaciones dadas en el teórico y en el práctico.