Ejercicios Extra Parte 1: Expresiones Cuantificadas y Programación Funcional

Algoritmos y Estructuras de Datos I 2^{do} cuatrimestre 2016

Demostración

1. Dada la definición recursiva de la función

demostrar que la siguiente propiedad

$$P.n: fac.n = \langle \sum i : 0 \le i < n : i * fac.i \rangle + 1$$

vale para todo $n \in Nat$.

2. Dada la siguiente definición recursiva para la funcion $repetir: Nat \to Nat \to [Nat]$, que dada una cantidad n y un valor x construye una lista de longitud n con elementos repetidos x:

$$\begin{array}{ccc} \operatorname{repetir.0.x} & \doteq & [\;] \\ \operatorname{repetir.}(n+1).x & \doteq & x \rhd \operatorname{repetir.n.x} \end{array}$$

Demostrá que es válido #(repetir.n.x) = n.

3. Dada la siguiente definición recursiva

$$rev: [A] \to [A]$$

$$rev.[] \doteq []$$

$$rev.(x \triangleright xs) \doteq rev.xs + [x]$$

escriba en lenguaje natural que hace la función y demuestre que la propiedad

$$P.xs.ys : rev.(xs + ys) = rev.ys + rev.xs$$

vale para toda lista xs y ys.

4. Para las funciones $sum : [Num] \to Num$ y $duplica : [Num] \to [Num]$ definidas recursivamente como:

Demostrá que se satisface sum.(duplica.xs) = 2 * sum.xs.

5. Especificar utilizando cuantificadores las funciones abren : $[Char] \rightarrow Nat$ y cierran : $[Char] \rightarrow Nat$, que devuelven la cantidad de paréntesis "(" y ")" respectivamente, que ocurren en una lista que contiene solo estos caracteres.

Luego, verificar que la siguiente definición recursiva satisface la especificación:

$$abren.[] \doteq 0$$

$$abren.(x \triangleright xs) \doteq (x = `(` \rightarrow abren.xs + 1)]$$

$$\square x \neq `(` \rightarrow abren.xs)$$

Especificación y Derivación

- 6. Expresar utilizando cuantificadores las siguientes sentencias del lenguaje natural:
 - a) El elemento e ocurre un número par de veces en la lista xs.
 - b) El elemento e ocurre en las posiciones pares de la lista xs.
 - c) El elemento e ocurre únicamente en las posiciones pares de la lista xs.
 - d) Si e ocurre en la lista xs, entonces l ocurre en alguna posicion anterior en la misma lista.
 - e) Existe un elemento de la lista xs que es estrictamente mayor a todos los demás.
 - f) Cualquier valor x que anula la funcion f (es decir que f.x = 0) ocurre en la lista xs.
 - g) En la lista xs solo ocurren valores que anulan la funcion f.
- 7. Derivar funciones recursivas a partir de cada una de las especificaciones que escribió para el ejercicio 6.
- 8. Sea fib la definición recursiva estándar para la función de Fibonacci. Calculá la función de Fibolucci, $Fbl: Nat \rightarrow Nat$, especificada como:

$$Fbl.n = \langle \sum i \, : \, 0 \leq i < n \, : \, \mathit{fib.i} \times \mathit{fib.}(n-i) \rangle$$

- 9. Derivar una función recursiva a partir de la especificacion del ejercicio 10c del práctico 2: La lista xs es un segmento final de la lista ys.
- 10. Especificá formalmente utilizando cuantificadores cada una de las siguientes funciones descriptas informalmente. Luego, derivá soluciones algorítmicas para cada una.
 - a) listas_iguales : $[A] \rightarrow [A] \rightarrow Bool$, que determina si dos listas son iguales, es decir, contienen los mismos elementos en las mismas posiciones respectivamente.
 - b) divide: $Nat \rightarrow Nat \rightarrow Bool$, que determina si el primer parámetro divide al segundo.
 - c) primo?: $Nat \rightarrow Bool$, que determina si un numero es primo.
- 11. Especificar y derivar una función que calcula la cantidad de apariciones de un segmento en una lista.
- 12. Derivar las funciones a partir de las especificaciones en el ejercicio 13 del práctico 2.