Ejercicio 1

Defina inductivamente (mediante cláusulas base, inductivas y extrema), c/u de los siguientes conjuntos:

a) El conjunto **Z** de los números enteros.
b) El conjunto **M** de todos los naturales múltiplos de 3.
c) El conjunto **O** de todos los naturales múltiplos de 4.
d) El conjunto **P** de todas las potencias de 2 (1, 2, 4, 8, 16, 32, 64, etc.).
e) El conjunto **J** de todos los naturales m tales que $m = 2n$, siendo n un natural impar.
f) El conjunto **K** de todos los naturales p tales que $p = n + (n/2)$ siendo n un natural par.
g) El conjunto **L** de todos los naturales x tales que $x = 4n + 1$, siendo n un natural cualquiera.
h) El conjunto **Q** de pares ordenados de naturales tales que $(x,y) \in Q \Leftrightarrow y = x+1$.
i) El conjunto **R** de pares ordenados de naturales tales que $(x,y) \in R \Leftrightarrow y = 2x$.
j) El conjunto **S** de pares ordenados de naturales tales que $(x,y) \in S \Leftrightarrow y = 2^x$.
k) El conjunto **T** de pares ordenados de naturales tales que $(x,y) \in T \Leftrightarrow y = x!$

Ejercicio 2

Dadas las siguientes funciones definidas por concordancia de patrones:

Alfa: N x N → N
Alfa (0, 0) = 1
Alfa (0, s(m)) = Alfa (0, m) + 2
Alfa (s(n), m) = Alfa (n,m) + Alfa (m,n)

Beta: Sec (A) → Sec (A)
Beta (nil) = nil
Beta (cons (x, nil)) = nil
Beta (cons (x, cons (y, s))) = cons (y, Beta (s))

a) Indique cuántos pasos base y cuántos pasos recursivos tiene cada una.
b) Calcule Alfa (2,1) aplicando paso a paso la definición de la función Alfa.
c) Calcule Beta ([5, 7, 1, 6, 3]) aplicando paso a paso la definición de la función Beta.
d) ¿Son funciones totales o parciales? Justifique su respuesta.

Ejercicio 3

Defina por concordancia de patrones las siguientes funciones sobre números naturales. Indique además si son totales, inyectivas, sobreyectivas, justificando sus respuestas.

a) ***EsPar: N → Boolean*** devuelve true si el número es par, false en caso contrario. Se pide definir esta función sin utilizar el operador módulo (resto de la división).
b) ***SumaHastaN: N → N*** dado $n \in N$, devuelve la suma de todos los naturales entre 0 y n.
c) ***SumaPares: N → N*** dado $n \in N$, devuelve la suma de todos los naturales pares entre 0 y n.
d) ***Potencia: N×N → N*** dados dos naturales n, m devuelve el resultado de elevar n^m. Se pide definir esta función sin utilizar el operador de potencia.
e) ***Max: N×N → N*** dados dos naturales n, m devuelve el valor más grande de entre ellos dos. Se pide definir esta función sin utilizar resta ni operadores de comparación.
f) ***SumaCuad: N → N*** dado un natural n, devuelve la suma de los cuadrados de todos los naturales que hay entre n y 0. Por ejemplo: SumCuad (3) = $3^2 + 2^2 + 1^2 + 0^2$.
g) ***SumaPots2: N → N*** dado un natural n, devuelve la suma de todas las potencias de 2 que hay entre 0 y n. Por ejemplo: SumPot (3) = $2^3 + 2^2 + 2^1 + 2^0$. Para resolverlo, hacer uso de la función *Potencia* definida en la parte (d).

Ejercicio 4

Defina por concordancia de patrones las siguientes funciones sobre secuencias. Indique además si son totales, inyectivas, sobreyectivas, justificando sus respuestas.

a) ***Esvacia: Sec(A) → Boolean*** devuelve true si y sólo si la secuencia es vacía.

b) ***Primero: Sec(A) → A*** dada una secuencia no vacía, devuelve su primer elemento.

c) ***Resto: Sec(A) → Sec(A)*** dada una secuencia no vacía, devuelve la secuencia que resulta de quitarle su primer elemento.

d) ***ContarDistintos: Sec(A) x A → N*** dada una secuencia y un elemento, cuenta la cantidad de ocurrencias de elementos en la secuencia que son diferentes al elemento dado.

e) ***ContarPosImpares: Sec(A) → N*** dada una secuencia, cuenta la cantidad de elementos que ocupan posiciones impares en ella (suponga que las posiciones se numeran a partir de 1).

f) ***Pertenece: Sec(A) x A → Boolean*** dada una secuencia y un elemento, devuelve true si y sólo si el elemento recibido pertenece a la secuencia, false en caso contrario.

g) ***TodosIguales: Sec(A) → Boolean*** dada una secuencia y un elemento, devuelve true si y sólo si todos sus elementos son iguales entre sí, false en caso contrario.

h) ***SecIguales: Sec(A) x Sec(A) → Boolean*** dadas dos secuencias, determina si son iguales.

i) ***Elemsec: Sec(A) x N → A*** dada una secuencia y un número natural *n* > 0, devuelve el elemento que ocupa la posición *n* de la secuencia (suponga que las posiciones se numeran a partir de 1).

j) ***Ultimo: Sec(A) → A*** dada una secuencia no vacía, devuelve su último elemento.

k) ***BorrarUltimo: Sec(A) → Sec(A)*** dada una secuencia no vacía, devuelve la secuencia resultante de borrarle su último elemento.

l) ***Tomar: Sec(A) x N → Sec(A)*** dada una secuencia y un número natural *n*, devuelve otra secuencia que resulta de tomar los primeros *n* elementos de la secuencia.

m) ***Borrar: Sec(A) x N → Sec(A)*** Dada una secuencia y un número natural *n*, devuelve otra secuencia que resulta de borrar los primeros *n* elementos de la secuencia.

n) ***FiltrarDistintos: Sec(A) x A → Sec(A)*** dada una secuencia y un elemento, devuelve otra secuencia formada por aquellos elementos en la secuencia original que son diferentes al elemento dado.

o) ***InsBack: Sec(A) x A → Sec(A)*** dadas una secuencia y un valor, devuelve la secuencia resultante de insertar el nuevo valor al final de la secuencia.

p) ***Repetidos: Sec(A) → Boolean*** dada una secuencia, determina si en ella hay o no algún valor repetido.

Ejercicio 5

Demuestre las siguientes propiedades por inducción estructural en los naturales:

a) $\forall n \in N$ se cumple que $(2n + 1)$ es impar.

b) $\forall n \in N$ se cumple que $2^n \leq 3^n$.

c) $\forall n \in N$ se cumple que $(5^n - 1)$ es múltiplo de 4.

d) $\forall n \in N$ se cumple que $(n^3 - n)$ es múltiplo de 3.

e) $\forall n \in N$, se cumple que $(4^n - 1)$ es múltiplo de 3.

f) $\forall n \in N$, $\forall m \in N$ se cumple que $a^n \times a^m = a^{(n+m)}$ (siendo $a \neq 0$).

g) $\sum_{i=0}^{n} i = n(n+1)/2$

Ejercicio 6

Dadas las siguientes funciones definidas sobre secuencias de elementos:

Largo : Sec(A) $\rightarrow$ N
Largo (nil) = 0
Largo (cons (x,s)) = 1 + Largo (s)

Clonar : Sec(A) $\rightarrow$ Sec(A)
Clonar (nil) = nil
Clonar (cons(x,s)) = cons (x, Clonar(s))

Duplicar : Sec(A) $\rightarrow$ Sec(A)
Duplicar (nil) = nil
Duplicar (cons(x,s)) = cons (x, cons (x, Duplicar(s)))

a) Aplique paso a paso la función Largo para calcular Largo ([a, b, c]).
b) Aplique paso a paso la función Clonar para calcular Clonar ([a, b, c]).
c) Aplique paso a paso la función Duplicar para calcular Duplicar ([a, b, c]).
d) Demuestre por inducción estructural que para toda s $\in$ Sec(A) se cumple que:
Largo (Clonar (s)) = Largo (s).
e) Demuestre por inducción estructural que para toda s $\in$ Sec(A) se cumple que:
Largo (Duplicar (s)) = 2 * Largo (s).

Ejercicio 7

Dadas las siguientes funciones definidas sobre secuencias de naturales:

Largo : Sec(N) $\rightarrow$ N
Largo (nil) = 0
Largo (cons (x,s)) = 1 + Largo (s)

Sumar : Sec(N) $\rightarrow$ N
Sumar (nil) = 0
Sumar (cons (x,s)) = x + Sumar (s)

Sustituir : Sec (N) x N $\rightarrow$ Sec (N)
Sustituir (nil, n) = nil
Sustituir (cons (x,s), n) = cons (n, Sustituir (s, n))

a) Calcule el resultado de Largo (Sustituir ([3, 2, 7], 5)), aplicando paso a paso las definiciones de las funciones Largo y Sustituir.
b) Calcule el resultado de Sumar (Sustituir ([3, 2, 7], 1)), aplicando paso a paso las definiciones de las funciones Sumar y Sustituir.
c) Demuestre por inducción estructural que para toda s $\in$ Sec(N) y para todo n $\in$ N se cumple que:
Largo (s) = Largo (Sustituir (s, n)).
d) Demuestre por inducción estructural que para toda s $\in$ Sec(A) se cumple que:
Largo (s) = Sumar (Sustituir (s, 1)).