Ejercicios de Coloquio - IntroAlg 2021

1.

c) *mayoresQue* : $Int \to [Int] \to [Int]$, que dado un entero n y una lista de enteros xs devuelve una lista sólo con los números mayores que n contenidos en xs,

Por ejemplo: $mayoresQue.2.[3, 0, -2, 12] = [3, 12]$

2.

2. (a) **[15 pto(s)]** Definir la función recursiva *tamaños* : $[[a]] \to [Num]$ que dada una lista de listas de elementos de tipo a, retorna la lista que contiene el tamaño de cada una de esas listas. Ejemplos:

   (I) $tamaños.[[11, 2], [\,]] = [2, 0]$

3.

c) *multiplica* : $Int \to [Int] \to [Int]$, que dado un número n y una lista, multiplica cada uno de los elementos por n.

Por ejemplo: $multiplica.3.[3, 0, -2] = [9, 0, -6]$

4.

e) *ultimo* : $[A] \to A$, que devuelve el último elemento de una lista.

Por ejemplo: $ultimo.[10, 5, 3, 1] = 1$

5.

f) *repetir* : $Num \to Num \to [Num]$, que dado un número n mayor o igual a 0 y un número k arbitrario construye una lista donde k aparece repetido n veces.

Por ejemplo: $repetir.3.6 = [6, 6, 6]$

6.

3. (a) **[15 pto(s)]** Definir una función recursiva *hacerA* : $String \to String$ que dada una string cambia todas sus vocales por 'a'. Ejemplos:

   (I) *hacerA*."¡Pero, che!" = "¡Para, cha!"

   (II) *hacerA*."Famaf"= "Famaf"

7.

2. (a) **[15 pto(s)]** Definir la función recursiva *siguientes* : $[Num] \to [(Num, Num)]$, que dada una lista de números retorna la lista resultante de armar un par con cada uno de ellos y sus respectivos números siguientes. Ejemplos:

   (I) $siguientes.[11, 7] = [(11, 12), (7, 8)]$

   (b) **[5 pto(s)]** Evaluar manualmente la función utilizando el ejemplo (I). Justificar cada paso.

8.

12. Considerando la función $quitarCeros : [Num] \to [Num]$ definida de la siguiente manera

$$\begin{array}{lcll} quitarCeros.[] & \doteq & [] & \\ quitarCeros.(x \triangleright xs) & \doteq & (& x \neq 0 \to x \triangleright quitarCeros.xs \\ & & \Box & x = 0 \to quitarCeros.xs \\ & &) & \end{array}$$

demostrá que

$$sum.(quitarCeros.xs) = sum.xs$$

9.

[**25 pto(s)**] Dadas las siguientes funciones $invertir : [Num] \to [Num]$, y $sum : [Num] \to Num$

$$\begin{array}{rcl} invertir.[\,] & \doteq & [\,] \\ invertir.(x \triangleright xs) & \doteq & (-x) \triangleright (invertir.xs) \end{array} \qquad \begin{array}{rcl} sum.[\,] & \doteq & 0 \\ sum.(x \triangleright xs) & \doteq & x + sumxs \end{array}$$

demostrar por inducción la siguiente propiedad

$$sum.(invertir.xs) = -sum.xs$$

10.

4. [**25 pto(s)**] Dadas las siguientes funciones

$$\begin{array}{rcl} stutter.[\,] & \doteq & [\,] \\ stutter.(x \triangleright xs) & \doteq & x \triangleright (x \triangleright stutter.xs) \end{array} \qquad \begin{array}{rcl} \#[\,] & \doteq & 0 \\ \#(x \triangleright xs) & \doteq & 1 + \#xs \end{array}$$

demuestre por inducción la siguiente propiedad

$$\#(stutter.xs). = 2 * \#xs$$

11.

13. Considerando la función $soloPares : [Num] \to [Num]$ definida de la siguiente manera

$$\begin{array}{lcll} soloPares.[] & \doteq & [] & \\ soloPares.(x \triangleright xs) & \doteq & (& x \bmod 2 = 0 \to x \triangleright soloPares.xs \\ & & \Box & x \bmod 2 \neq 0 \to soloPares.xs \\ & &) & \end{array}$$

demostrá que

$$soloPares.(xs +\!\!+ ys) = soloPares.xs +\!\!+ soloPares.ys$$

12.

[**20 pto(s)**] Dada la función recursiva *cuantos0y1* : $[Num] \to Num$ que cuenta la cantidad de 0s y 1s en una lista, y *swap0y1* : $[Num] \to [Num]$ que cambia cada 0 por un 1 y vice-versa, definidas como:

$$
\begin{array}{rcl}
cuantos0y1.[\,] & \doteq & 0 \\
cuantos0y1.(x \triangleright xs) & \doteq & (\ x = 0 \to 1 + cuantos0y1.xs \\
 & & \Box x = 1 \to 1 + cuantos0y1.xs \\
 & & \Box (x \neq 0) \wedge (x \neq 1) \to cuantos0y1.xs \\
 & &)
\end{array}
$$

$$
\begin{array}{rcl}
swap0y1.[\,] & \doteq & [\,] \\
swap0y1.(x \triangleright xs) & \doteq & (\ x = 0 \to 1 \triangleright swap0y1.xs \\
 & & \Box x = 1 \to 0 \triangleright swap0y1.xs \\
 & & \Box (x \neq 0) \wedge (x \neq 1) \to x \triangleright swap0y1.xs \\
 & &)
\end{array}
$$

demuestre por inducción que $cuantos0y1.xs = cuantos0y1.(swap0y1.xs)$

13.

5. [**25 pto(s)**] Dada las siguientes funciones recursivas *cuantos* : $Num \to [Num] \to Num$ y *agrega0si1* : $[Num] \to [Num]$, definidas como:

$$
\begin{array}{rcl}
cuantos.n.[\,] & \doteq & 0 \\
cuantos.n.(x \triangleright xs) & \doteq & (\ x = n \to 1 + cuantos.n.xs \\
 & & \Box (x \neq n) \to cuantos.n.xs \\
 & &)
\end{array}
$$

$$
\begin{array}{rcl}
agregaSiguiente.[\,] & \doteq & [\,] \\
agregaSiguiente.(x \triangleright xs) & \doteq & x \triangleright ((x+1) \triangleright agregaSiguiente.xs)
\end{array}
$$

demuestre por inducción que $cuantos.1.(agregaSiguiente.xs) = cuantos.0.xs + cuantos.1.xs$

(Sugerencia: Considerar los casos en que la lista empiece con 0, con 1, o con otro número diferente.)

14.

g) *De Morgan para la conjunción*: $\neg(p \wedge q) \equiv \neg p \vee \neg q$

15.

h) *Ley de absorción*: $p \wedge (p \vee q) \equiv p$

16.

e) *Debilitamiento para* $\vee$: $p \Rightarrow p \vee q$.

17.

f) *Modus Ponens* $p \wedge (p \Rightarrow q) \equiv p \wedge q$.

18.

h) *Contrarreciproca* $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$.

19.

b) *Monotonía de la conjunción*: $(p \Rightarrow q) \Rightarrow (p \wedge r \Rightarrow q \wedge r)$.

20.

a) **[15 pto(s)]** $p \Rightarrow q \equiv q \Rightarrow p \equiv p \equiv q$

21.

1. Demostrar que las siguientes fórmulas son teoremas del Cálculo Proposicional. En cada paso de la demostración indique que axioma o teorema se utiliza, y subraye la subfórmula involucrada. Se pueden utilizar, sin demostrar, los axiomas y teoremas dados en el Digesto Proposicional.

   a) **[15 pto(s)]** $(p \Rightarrow q) \Rightarrow (p \wedge r \Rightarrow q)$.

   b) **[15 pto(s)]** $(\neg q \Rightarrow \neg p) \vee r \equiv p \vee q \vee r \equiv q \wedge r \equiv q \equiv r$.

22.

a) **[15 pto(s)]** $p \Rightarrow (q \Rightarrow r) \equiv \neg r \Rightarrow (p \Rightarrow \neg q)$

23.

b) **[15 pto(s)]** $\neg p \wedge q \equiv \neg(p \wedge \neg q \equiv q) \equiv (p \equiv \mathit{False})$.

24.

Formalizar las siguientes propiedades escritas en lenguaje natural, en el lenguaje de la lógica de predicados:

a) **[10 pto(s)]** *"Todas las figuras de xs que tiene tamaño mayor a 5, son cuadrados".*
Ejemplos: Las listas $[(\mathit{Triangulo}, \mathit{Rojo}, 3)]$ y $[(\mathit{Triangulo}, \mathit{Rojo}, 5), (\mathit{Cuadrado}, \mathit{Verde}, 6)]$ satisfacen la propiedad. La lista $[(\mathit{Rombo}, \mathit{Rojo}, 20)]$ no la satisface.

25.

Formalizar las siguientes propiedades escritas en lenguaje natural, en el lenguaje de la lógica de predicados:

a) **[10 pto(s)]** *"Hay algún cuadrado en xs de tamaño menor a 10"*
Ejemplo: La lista $xs = [(Triangulo, Rojo, 5), (Cuadrado, Azul, 9)]$ satisface la propiedad y la lista $xs = [(Triangulo, Rojo, 5)]$ no la satisface.

26.

Demostrar la siguiente propiedad:

b) $\langle \exists x : : \mathit{cuadrado}.x \rangle \wedge \langle \forall y : : \mathit{amarillo}.y \rangle \Rightarrow \langle \exists x : : \mathit{cuadrado}.x \wedge \mathit{amarillo}.x \rangle$.

27.

2. Formalizar las siguientes propiedades escritas en lenguaje natural, en el lenguaje de la lógica de predicados:

   a) **[10 pto(s)]** *"Ningún círculo en xs es rojo".*
   Ejemplos: Las listas $[(\mathit{Rombo}, \mathit{Rojo}, 3)]$ y $[(\mathit{Circulo}, \mathit{Azul}, 3)]$ satisfacen la propiedad. La lista $[(\mathit{Circulo}, \mathit{Rojo}, 2)]$ no la satisface.

28.

b) **[10 pto(s)]** *"Hay un único cuadrado en xs y es rojo".*
Ejemplos: Las listas $[(\mathit{Cuadrado}, \mathit{Rojo}, 3)]$ y $[(\mathit{Cuadrado}, \mathit{Rojo}, 2), (\mathit{Rombo}, \mathit{Azul}, 1)]$ satisfacen la propiedad. Las listas $[(\mathit{Rombo}, \mathit{Azul}, 1)]$ y $[(\mathit{Cuadrado}, \mathit{Rojo}, 1), (\mathit{Cuadrado}, \mathit{Azul}, 2)]$ no la satisfacen.

29.

2. Formalizar las siguientes propiedades escritas en lenguaje natural, en el lenguaje de la lógica de predicados:

   a) [**10 pto(s)**] *"Todas las figuras de xs son triangulos rojos o cuadrados".*
   Ejemplos: Las listas $[(Triangulo, Rojo, 3)]$ y $[(Triangulo, Rojo, 5), (Cuadrado, Verde, 6)]$ satisfacen la propiedad. La listas $[(Rombo, Rojo, 2)]$ y $[(Triangulo, Rojo, 5), (Triangulo, Verde, 6)]$ no la satisfacen.

30.

b) [**10 pto(s)**] *"Existe un único elemento de xs que es mayor estricto que cero".*
Ejemplos: Las listas $[-1, 0, 3]$ y $[-4, -3, 7, -1]$ satisfacen la propiedad. Las listas $[6, 5]$ y $[9, 6, 7]$ no la satisfacen.

31.

2. Formalizar las siguientes propiedades escritas en lenguaje natural, en el lenguaje de la lógica de predicados:

   a) [**10 pto(s)**] *"Hay una triángulo en xs con tamaño menor a 5".*

   b) [**10 pto(s)**] *"El último elemento de xs está en ys".*

32.

4. [**20 pto(s)**] Demostrar que la siguiente fórmula es teorema del Cálculo de Predicados. En cada paso de la demostración indique que axioma o teorema se utiliza, y subraye la subfórmula involucrada. Se pueden utilizar, sin demostrar, los axiomas y teoremas dados en el Digesto Proposicional y en el Digesto de Predicados.

$$\langle \forall x : \, : \neg(P.x \Rightarrow Q.x) \rangle \equiv (\langle \forall x : \, : P.x \rangle \wedge \langle \forall x : \, : \neg Q.x \rangle).$$

Demostrar que las siguientes fórmulas son teoremas del Cálculo de Predicados. En cada paso de la demostración indique que axioma o teorema se utiliza, y subraye la subfórmula involucrada. Se pueden utilizar, sin demostrar, los axiomas y teoremas dados en el Digesto Proposicional y en el Digesto de Predicados.

a) [**15 pto(s)**] $\langle \forall x : \, : T.x \rangle \Rightarrow \langle \forall x : R.x : T.x \rangle$

33.

[**20 pto(s)**] Demostrar que la siguiente fórmula es teorema del Cálculo de Predicados. En cada paso de la demostración indique que axioma o teorema se utiliza, y subraye la subfórmula involucrada. Se pueden utilizar, sin demostrar, los axiomas y teoremas dados en el Digesto Proposicional y en el Digesto de Predicados.

$$(\langle \exists x : R.x : T.x \rangle \wedge \langle \forall x : R.x : S.x \rangle) \Rightarrow \langle \exists x : R.x : T.x \wedge S.x \rangle$$

34. Demostrar que la siguiente fórmula es teorema del Cálculo de predicados.

$$\langle \forall x : \, : P.x \Rightarrow Q.x \rangle \Rightarrow (\langle \exists x : \, : P.x \rangle \Rightarrow \langle \exists x : \, : Q.x \rangle).$$

35.

36.

4. [**20 pto(s)**] Demostrar que la siguiente fórmula es teorema del Cálculo de Predicados. En cada paso de la demostración indique qué axioma o teorema se utiliza, y subraye la subfórmula involucrada. Se pueden utilizar, sin demostrar, los axiomas y teoremas dados en el Digesto Proposicional y en el Digesto de Predicados.

$$\langle \forall x : R.x : T.x \rangle \wedge \langle \forall x : \neg T.x : R.x \rangle \equiv \langle \forall x : \; : T.x \rangle$$

37.

Dada la definición de la función *algunCuadrado*:

$$\begin{array}{l} algunCuadrado : [Figura] \to Bool \\ algunCuadrado.[\,] \doteq False \\ algunCuadrado.(x \triangleright xs) \doteq cuadrado.x \vee algunCuadrado.xs \end{array}$$

demostrá por inducción la siguiente fórmula

$$algunCuadrado.xs \equiv \langle \exists x : x \in_\ell xs : cuadrado.x \rangle.$$

38.

Dada la definición de la función *soloCeros*:

$$\begin{array}{l} soloCeros : [Num] \to Bool \\ soloCeros.[\,] \doteq True \\ soloCeros.(x \triangleright xs) \doteq x = 0 \wedge soloCeros.xs \end{array}$$

demostrá por inducción la siguiente fórmula

$$soloCeros.xs \equiv \langle \forall x : x \in_\ell xs : x = 0 \rangle.$$

39.

5. [**20 pto(s)**] Dada la definición de la función *hayTR*:

$$\begin{array}{l} hayTR : [Figura] \to Bool \\ hayTR.[\,] \doteq False \\ hayTR.(x \triangleright xs) \doteq (triangulo.x \wedge rojo.x) \vee hayTR.xs \end{array}$$

demostrar por inducción la siguiente fórmula

$$hayTR.xs \equiv \langle \exists y : y \in_\ell xs : triangulo.y \wedge rojo.y \rangle.$$

40.

5. **[25 pto(s)]** Dada la definición de la función $hayCoT$:

$$
\begin{aligned}
&hayCoT : [Figura] \rightarrow Bool\\
&hayCoT.[\,] \doteq False\\
&hayCoT.(x \triangleright xs) \doteq (circulo.x \vee triangulo.x) \vee hayCoT.xs
\end{aligned}
$$

demostrá por inducción la siguiente fórmula

$$hayCoT.xs \equiv \langle \exists y : y \in_{\ell} xs : circulo.y \vee triangulo.y \rangle.$$