Guia 2

2do cuatrimestre 2024 — Algoritmos y Estructuras de Datos I

Integrante	LU	Correo electrónico
Federico Barberón	112/24	jfedericobarberonj@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - Pabellón I
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Argentina
Tel/Fax: (54 11) 4576-3359
http://exactas.uba.ar

Índice

1. Guia 2

1.1. Ejercicio 1

Nombrar los siguientes predicados sobre enteros:

a) `pred esCuadrado` (x: $\mathbb{Z}$) {

$(\exists c : \mathbb{Z})\ (c > 0 \wedge (c * c = x))$

}

b) `pred esPrimo` (x: $\mathbb{Z}$) {

$(\forall n : \mathbb{Z})\ ((1 < n < x) \rightarrow_L (x \mod n \neq 0))$

}

1.2. Ejercicio 2

Escriba los siguientes predicados sobre números enteros en lenguaje de especifiación:

a) Que sea verdadero si y sólo si x e y son coprimos.

`pred sonCoprimos` (x, y: $\mathbb{Z}$) {

$(\forall i : \mathbb{Z})\ (i > 1 \rightarrow_L \neg(x \mod i = 0 \wedge y \mod i = 0))$

}

b) Que sea verdadero si y es el mayor primo que divide a x.

`pred mayorPrimoQueDivide` (x, y: $\mathbb{Z}$) {

$(esPrimo(y) \wedge_L x \mod y = 0) \wedge \neg(\exists i : \mathbb{Z})\ (esPrimo(i) \wedge_L (x \mod i = 0 \wedge i > y))$

}

1.3. Ejercicio 3

Nombre los siguientes predicados auxiliares sobre secuencias de enteros:

a) `pred todoPositivos` (s: $seq\langle\mathbb{Z}\rangle$) {

$(\forall i : \mathbb{Z})\ ((0 \leq i < |s|) \rightarrow_L s[i] \geq 0)$

}

b) `pred todosDistintos` (s: $seq\langle\mathbb{Z}\rangle$) {

$(\forall i : \mathbb{Z})\ ((0 \leq i < |s|) \rightarrow_L (\forall j : \mathbb{Z})\ ((0 \leq j < |s| \wedge i \neq j) \rightarrow_L (s[i] \neq s[j])))$

}

1.4. Ejercicio 4

Escriba los siguientes predicados auxiliares sobre secuencias de enteros, aclarando los tipos de los parámetros que recibe:

a) *esPrefijo*, que determina si una secuencia es prefijo de otra.

`pred esPrefijo` (s1, s2: $seq\langle\mathbb{Z}\rangle$) {

$(|s1| \leq |s2|) \wedge_L (\forall i : \mathbb{Z})\ ((0 \leq i < |s1|) \rightarrow_L (s1[i] = s2[i]))$

}

b) *estáOrdenada*, que determina si la secuencia está ordenada de menor a mayor.

`pred estáOrdenada` (s: $seq\langle\mathbb{Z}\rangle$) {

$(\forall i : \mathbb{Z})\ ((0 \leq i < |s| - 1) \rightarrow_L (s[i] \leq s[i+1]))$

}

c) *hayUnoParQueDivideAlResto*, que determina si hay un elemento par en la secuencia que divide a todos los otros elementos de la secuencia.

`pred divideA` (d, n: $\mathbb{Z}$) {

$(d \neq 0) \wedge_L n \mod d = 0$

}

`pred hayUnoParQueDivideAlResto` (s: $seq\langle\mathbb{Z}\rangle$) {

$(\exists i : \mathbb{Z})\ ((0 \leq i < |s|) \wedge_L esPar(s[i]) \wedge (\forall j : \mathbb{Z})\ ((0 \leq j < |s|) \wedge_L divideA(s[i], s[j]))))$

}

d) *enTresPartes*, que determina si en la secuencia aparecen (de izquieda a derecha) primero 0s, despúes 1s y por último 2s. Por ejemplo $\langle 0, 0, 1, 1, 1, 1, 2\rangle$ cumple, pero $\langle 0, 1, 3, 0\rangle$ o $\langle 0, 0, 0, 1, 1\rangle$ no. ¿Cómo modificaría la expresión para que se admitan cero apariciones de 0s, 1s y 2s (es decir, para que por ejemplo $\langle 0, 0, 0, 1, 1\rangle$ o $\langle\rangle$ sí cumplan)?

`pred tieneSoloCeroUnoYDos` (s: $seq\langle\mathbb{Z}\rangle$) {

$(\forall i : \mathbb{Z})\ ((0 \leq i < |s|) \rightarrow_L (s[i] = 0 \vee s[i] = 1 \vee s[i] = 2))$

}

`pred enTresPartes` (s: $seq\langle\mathbb{Z}\rangle$) {

$tieneSoloCeroUnoYDos(s) \wedge estaOrdenada(s)$

}

1.5. Ejercicio 5

Sea s una secuencia de elementos de tipo $\mathbb{Z}$. Escribir una expresión (utilizando sumatoria y productoria) tal que:

a) Cuente la cantidad de veces que aparece el elemento e de tipo $\mathbb{Z}$ en la secuencia s.

$\sum_{i=0}^{|s|-1} \text{if } s[i] = e \text{ then } 1 \text{ else } 0 \text{ fi}$

b) Sume los elementos en las posiciones impares de la secuencia s.

$\sum_{i=0}^{|s|-1} \text{if } \neg esPar(s[i]) \text{ then } s[i] \text{ else } 0 \text{ fi}$

c) Sume los elementos mayores a 0 contenidos en la secuencia s.

$\sum_{i=0}^{|s|-1} \text{if } s[i] > 0 \text{ then } s[i] \text{ else } 0 \text{ fi}$

d) Sume los inverso multiplicativos ($\frac{1}{x}$) de los elementos contenidos en la secuencia s distintos a 0.

$\sum_{i=0}^{|s|-1} \text{if } s[i] \neq 0 \text{ then } \frac{1}{s[i]} \text{ else } 0 \text{ fi}$

1.6. Ejercicio 6

Las siguientes especificaciones no son correctas. Indicar por qué y corregirlas para que describan correctamente el problema.

a) progresionGeometricaFactor2: Indica si la secuencia l representa una progresión geométrica factor 2. Es decir, si cada elemento de la secuencia es el doble del elemento anterior.

`proc progresionGeometricaFactor2` (in l: $seq\langle\mathbb{Z}\rangle$) : Bool

`requiere` $\{True\}$

asegura $\{res = True \leftrightarrow ((\forall i : \mathbb{Z})\ (0 \leq i < |l| \rightarrow_L l[i] = 2 * l[i-1]))\}$

El asegura se indefine con $i = 0$ pues trata de acceder a $l[-1]$.

Solucion:

proc progresionGeometricaFactor2 (in l: $seq\langle\mathbb{Z}\rangle$) : Bool

requiere $\{True\}$

asegura $\{res = True \leftrightarrow ((\forall i : \mathbb{Z})\ (0 < i < |l| \rightarrow_L l[i] = 2 * l[i-1]))\}$

b) minimo: Devuelve en res el menor elemento de l.

proc minimo (in l: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$

requiere $\{True\}$

asegura $\{(\forall y : \mathbb{Z})\ ((y \in l \land y \neq x) \rightarrow y > res)\}$

En el asegura se hace referencia a x que no está definida. La lista no puede estar vacía y res tiene que estar en la lista.

Solución:

proc minimo (in l: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$

requiere $\{|l| > 0\}$

asegura $\{res \in l \land (\forall y : \mathbb{Z})\ ((y \in l \land y \neq res) \rightarrow y > res)\}$

1.7. Ejercicio 7

Para los siguientes problemas, dar todas las soluciones posibles a las entradas dadas:

a) proc indiceDelMaximo (in l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{Z}$

requiere $\{|l| > 0\}$

asegura $\{0 \leq res < |l| \land_L ((\forall i : \mathbb{Z})\ (0 \leq i < |l| \rightarrow_L l[i] \leq l[res]))\}$

- I) $\langle 1, 2, 3, 4\rangle$ $res = 3$
- II) $\langle 15.5, -18, 4.215, 15.5, -1\rangle$ $res = 0 \lor res = 3$
- III) $\langle 0, 0, 0, 0, 0, 0\rangle$ $res \in [0, |l|)$

b) proc indiceDelPrimerMaximo (in l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{Z}$

requiere $\{|l| > 0\}$

asegura $\{0 \leq res < |l| \land_L ((\forall i : \mathbb{Z})\ (0 \leq i < |l| \rightarrow_L (l[i] \leq l[res] \lor (l[i] = l[res] \land i >= res))))\}$

- I) $\langle 1, 2, 3, 4\rangle$ $res = 3$
- II) $\langle 15.5, -18, 4.215, 15.5, -1\rangle$ $res = 0$
- III) $\langle 0, 0, 0, 0, 0, 0\rangle$ $res = 0$

c) ¿Para qué valores de entrada indiceDelPrimerMaximo y indiceDelMaximo tienen necesariamente la misma salida?

Tienen la misma salida sii el máximo de la lista no está repetido.

1.8. Ejercicio 8

Sea $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ definida como:

$$f(a,b) = \begin{cases} 2b & \text{si } a < 0 \\ b - 1 & \text{en otro caso} \end{cases}$$

Indicar cuáles de las siguientes especifiaciones son correctas para el problema de calcular $f(a, b)$. Para aquellas que no lo son, indicar por qué.

a) `proc f` (`in` a, b: $\mathbb{R}$) : $\mathbb{R}$

   `requiere` $\{True\}$

   `asegura` $\{(a < 0 \land res = 2 * b) \land (a \geq 0 \land res = b - 1)\}$

   Falla pues el asegura es siempre falso ya que, por la asociatividad, se tiene que $(a < 0 \land a \geq 0 \land \dots)$ lo cual es siempre falso.

b) `proc f` (`in` a, b: $\mathbb{R}$) : $\mathbb{R}$

   `requiere` $\{True\}$

   `asegura` $\{(a < 0 \land res = 2 * b) \lor (a \geq 0 \land res = b - 1)\}$

   Es correcta.

c) `proc f` (`in` a, b: $\mathbb{R}$) : $\mathbb{R}$

   `requiere` $\{True\}$

   `asegura` $\{(a < 0 \to res = 2 * b) \lor (a \geq 0 \to res = b - 1)\}$

   Falla pues siempre es verdadera.

d) `proc f` (`in` a, b: $\mathbb{R}$) : $\mathbb{R}$

   `requiere` $\{True\}$

   `asegura` $\{res = \texttt{IfThenElse}(a < 0,\ 2 * b,\ b - 1)\}$

   Es correcta.

1.9. Ejercicio 9

Considerar la siguiente especificación, junto con un algoritmo que dado x devuelve x^2.

`proc unoMasGrande` (`in` x: $\mathbb{R}$) : $\mathbb{R}$
   `requiere` $\{True\}$
   `asegura` $\{res > x\}$

a) ¿Qué devuelve el algoritmo si recibe $x = 3$? ¿El resultado hace verdadera la posrtcondición de unoMasGrande?

   El algoritmo devuelve 9 y hace verdadera la postcondición pues 9 ¿3.

b) ¿Qué sucede para las entradas $x = 0.5, x = 1, x = -0.2$ y $x = -7$?

   Para los primeros 2 falla y para los útlimos 2 cumple.

c) Teniendo en cuenta lo respondido en los puntos anteriores, escribir una **precondición** para unoMasGrande, de manera tal que el algoritmo cumpla con la especifiación.

   requiere $\{|x| > 1\}$