Algoritmos y Estructuras de Datos

Guía Práctica 2
Especificación de problemas

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

2.1. Funciones auxiliares

Ejercicio 1. ★ Escriba los siguientes predicados sobre números enteros en lenguaje de especificación:

a) *pred esCuadrado* $(x : \mathbb{Z})$ que sea verdadero si y sólo si x es un numero cuadrado.

b) *pred esPrimo* $(x : \mathbb{Z})$ que sea verdadero si y sólo si x es primo.

c) *pred sonCoprimos* $(x, y : \mathbb{Z})$ que sea verdadero si y sólo si x e y son coprimos.

d) *pred mayorPrimoQueDivide* $(x : \mathbb{Z}, y : \mathbb{Z})$ que sea verdadero si y es el mayor primo que divide a x.

Ejercicio 2. ★ 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.

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

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

d) *sinRepetidos*, que determina si la secuencia no tiene repetidos.

e) *enTresPartes*, que determina si en la secuencia aparecen (de izquierda a derecha) primero 0s, después 1s y por último 2s. Por ejemplo $\langle 0, 0, 1, 1, 1, 1, 2\rangle$ cumple con *enTresPartes*, 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 *enTresPartes*)?

Ejercicio 3. 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.

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

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

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

2.2. Análisis de especificación

Ejercicio 4. 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` $\{res = True \leftrightarrow ((\forall i : \mathbb{Z})(0 \leq i < |l| \rightarrow_L l[i] = 2 * l[i-1]))\}$

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

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

   `requiere` $\{True\}$

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

Ejercicio 5. 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` $\{0 \leq res < |l| \wedge_L ((\forall i : \mathbb{Z})(0 \leq i < |l| \rightarrow_L l[i] \leq l[res])\}$

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

b) `proc indiceDelPrimerMaximo` (in l:$seq\langle\mathbb{R}\rangle$) : $\mathbb{Z}$
    `requiere` $\{|l| > 0\}$
    `asegura` $\{0 \leq res < |l| \wedge_L ((\forall i : \mathbb{Z})(0 \leq i < |l| \rightarrow_L (l[i] < l[res] \vee (l[i] = l[res] \wedge i \geq res))))\}$

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

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

Ejercicio 6. Sea $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ 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 especificaciones 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` $\{(a < 0 \wedge res = 2 * b) \wedge (a \geq 0 \wedge res = b - 1)\}$

b) `proc f` (in a, b: $\mathbb{R}$) : $\mathbb{R}$
    `requiere` $\{True\}$
    `asegura` $\{(a < 0 \wedge res = 2 * b) \vee (a \geq 0 \wedge res = b - 1)\}$

c) `proc f` (in a, b: $\mathbb{R}$) : $\mathbb{R}$
    `requiere` $\{True\}$
    `asegura` $\{(a < 0 \rightarrow res = 2 * b) \vee (a \geq 0 \rightarrow res = b - 1)\}$

d) `proc f` (in a, b: $\mathbb{R}$) : $\mathbb{R}$
    `requiere` $\{True\}$
    `asegura` $\{res = (\text{if } a < 0 \text{ then } 2 * b \text{ else } b - 1 \text{ fi})\}$

Ejercicio 7. 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 postcondición de unoMasGrande?

b) ¿Qué sucede para las entradas $x = 0{,}5$, $x = 1$, $x = -0{,}2$ y $x = -7$?

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

2.3. Relación de fuerza

Ejercicio 8. Sean x y r variables de tipo $\mathbb{R}$. Considerar los siguientes predicados:

P1: $\{x \leq 0\}$
P2: $\{x \leq 10\}$
P3: $\{x \leq -10\}$

Q1: $\{r \geq x^2\}$
Q2: $\{r \geq 0\}$
Q3: $\{r = x^2\}$

a) Indicar la relación de fuerza entre P1, P2 y P3

b) Indicar la relación de fuerza enrte Q1, Q2 y Q3

c) Escribir 2 programas que cumplan con la siguiente especificación:

`proc hagoAlgo` (in x: $\mathbb{R}$) : $\mathbb{R}$
    `requiere` $\{x \leq 0\}$
    `asegura` $\{res \geq x^2\}$

d) Sea A un algoritmo que cumple con la especificación del ítem anterior. Decidir si necesariamente cumple las siguientes especificaciones:

a) `requiere` $\{x \leq -10\}$, `asegura` $\{r \geq x^2\}$
b) `requiere` $\{x \leq 10\}$, `asegura` $\{r \geq x^2\}$
c) `requiere` $\{x \leq 0\}$, `asegura` $\{r \geq 0\}$
d) `requiere` $\{x \leq 0\}$, `asegura` $\{r = x^2\}$
e) `requiere` $\{x \leq -10\}$, `asegura` $\{r \geq 0\}$
f) `requiere` $\{x \leq 10\}$, `asegura` $\{r = x^2\}$

e) ¿Qué conclusión pueden sacar? ¿Qué debe cumplirse con respecto a las precondiciones y postcondiciones para que sea seguro **reemplazar la especificación**?

Ejercicio 9. Considerar las siguientes dos especificaciones, junto con un algoritmo a que satisface la especificación de `p2`.

`proc p1` (in x: $\mathbb{R}$, in n: $\mathbb{Z}$) : $\mathbb{Z}$
    `requiere` $\{x \neq 0\}$
    `asegura` $\{x^n - 1 < res \leq x^n\}$

`proc p2` (in x: $\mathbb{R}$, in n: $\mathbb{Z}$) : $\mathbb{Z}$
    `requiere` $\{n \leq 0 \rightarrow x \neq 0\}$
    `asegura` $\{res = \lfloor x^n \rfloor\}$

a) Dados valores de x y n que hacen verdadera la precondición de `p1`, demostrar que hacen también verdadera la precondición de `p2`.

b) Ahora, dados estos valores de x y n, supongamos que se ejecuta a: llegamos a un valor de res que hace verdadera la postcondición de `p2`. ¿Será también verdadera la postcondición de `p1` con este valor de res?

c) ¿Podemos concluir que a satisface la especificación de `p1`?

2.4. Especificación de problemas

Ejercicio 10. Especificar los siguientes problemas:

a) Dado un entero, decidir si es par

b) Dado un entero n y otro m, decidir si n es un múltiplo de m

c) Dado un entero, listar todos sus divisores positivos (sin duplicados)

d) Dado un entero positivo, obtener su descomposición en factores primos. Devolver una secuencia de tuplas (p, e), donde p es un factor primo y e es su exponente, ordenada en forma creciente con respecto a p

Ejercicio 11. Especificar los siguientes problemas sobre secuencias:

a) Dadas dos secuencias s y t, decidir si s está *incluida* en t, es decir, si todos los elementos de s aparecen en t en igual o mayor cantidad

b) Dadas dos secuencias s y t, devolver su *intersección*, es decir, una secuencia con todos los elementos que aparecen en ambas. Si un mismo elemento tiene repetidos, la secuencia retornada debe contener la cantidad mínima de apariciones del elemento en s y en t

c) Dada una secuencia de números enteros, devolver aquel que divida a más elementos de la secuencia. El elemento tiene que pertenecer a la secuencia original. Si existe más de un elemento que cumple esta propiedad, devolver alguno de ellos

d) Dada una secuencia de secuencias de enteros l, devolver una secuencia de l que contenga el máximo valor. Por ejemplo, si $l = \langle\langle 2,3,5\rangle, \langle 8,1\rangle, \langle 2,8,4,3\rangle\rangle$, devolver $\langle 8,1\rangle$ o $\langle 2,8,4,3\rangle$

e) Dada una secuencia l con todos sus elementos distintos, devolver la secuencia de *partes*, es decir, la secuencia de todas las secuencias incluidas en l, cada una con sus elementos en el mismo orden en que aparecen en l

2.5. Especificación de problemas usando inout

Ejercicio 12. Dados dos enteros a y b, se necesita calcular su suma y retornarla en un entero c. ¿Cúales de las siguientes especificaciones son correctas para este problema? Para las que no lo son, indicar por qué.

a) `proc sumar` (inout a, b, c: $\mathbb{Z}$)

    `requiere` $\{True\}$

    `asegura` $\{a + b = c\}$

b) `proc sumar` (in a, b: $\mathbb{Z}$, inout c: $\mathbb{Z}$)

    `requiere` $\{True\}$

    `asegura` $\{c = a + b\}$

c) `proc sumar` (inout a, b: $\mathbb{Z}$, inout c: $\mathbb{Z}$)

    `requiere` $\{a = A_0 \land b = B_0\}$

    `asegura` $\{a = A_0 \land b = B_0 \land c = a + b\}$

Ejercicio 13. Dada una secuencia l, se desea sacar su primer elemento y devolverlo. Decidir cúales de estas especificaciones son correctas. Para las que no lo son, indicar por qué y justificar con ejemplos.

a) `proc tomarPrimero` (inout l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{R}$

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

    `asegura` $\{res = \mathsf{head}(l)\}$

b) `proc tomarPrimero` (inout l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{R}$

    `requiere` $\{|l| > 0 \land l = L_0\}$

    `asegura` $\{res = \mathsf{head}(L_0)\}$

c) `proc tomarPrimero` (inout l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{R}$

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

    `asegura` $\{res = \mathsf{head}(L_0) \land |l| = |L_0| - 1\}$

d) `proc tomarPrimero` (inout l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{R}$

`requiere` $\{|l| > 0 \land l = L_0\}$

`asegura` $\{res = \mathsf{head}(L_0) \land l = \mathsf{tail}(L_0)\}$

Ejercicio 14. Dada una secuencia de enteros, se requiere multiplicar por 2 aquéllos valores que se encuentran en posiciones pares. Indicar por qué son incorrectas las siguientes especificaciones y proponer una alternativa correcta.

a) `proc duplicarPares` (inout l: $seq\langle\mathbb{Z}\rangle$)

`requiere` $\{l = L_0\}$

`asegura` {
$|l| = |L_0| \land$
$(\forall i : \mathbb{Z})(0 \leq i < |l| \land i \mod 2 = 0) \rightarrow_L l[i] = 2 * L_0[i]$
}

b) `proc duplicarPares` (inout l: $seq\langle\mathbb{Z}\rangle$)

`requiere` $\{l = L_0\}$

`asegura` {
$(\forall i : \mathbb{Z})((0 \leq i < |l| \land i \mod 2 \neq 0) \rightarrow_L l[i] = L_0[i]) \land$
$(\forall i : \mathbb{Z})((0 \leq i < |l| \land i \mod 2 = 0) \rightarrow_L l[i] = 2 * L_0[i])$
}

c) `proc duplicarPares` (inout l: $seq\langle\mathbb{Z}\rangle$) : $seq\langle\mathbb{Z}\rangle$

`requiere` $\{True\}$

`asegura` {
$|l| = |res| \land$
$(\forall i : \mathbb{Z})((0 \leq i < |l| \land i \mod 2 \neq 0) \rightarrow_L res[i] = l[i]) \land$
$(\forall i : \mathbb{Z})((0 \leq i < |l| \land i \mod 2 = 0) \rightarrow_L res[i] = 2 * l[i])$
}

Ejercicio 15. Especificar los siguientes problemas de modificación de secuencias:

a) proc primosHermanos(inout $l : seq\langle\mathbb{Z}\rangle$), que dada una secuencia de enteros mayores a dos, reemplaza dichos valores por el número primo menor más cercano. Por ejemplo, si $l = \langle 6, 5, 9, 14\rangle$, luego de aplicar primosHermanos(l), $l = \langle 5, 3, 7, 13\rangle$

b) proc reemplazar(inout $l : seq\langle\mathsf{Char}\rangle$, in $a, b : \mathsf{Char}$), que reemplaza todas las apariciones de a en l por b

c) proc limpiarDuplicados(inout $l : seq\langle\mathsf{Char}\rangle$, out $dup : seq\langle\mathsf{Char}\rangle$), que elimina los elementos duplicados de l dejando sólo su primera aparición (en el orden original). Devuelve además, dup una secuencia con todas las apariciones eliminadas (en cualquier orden)