Introducción a la Programación

Guía Práctica 2
Especificación de problemas

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

Ejercicio 1. Escribir semiformalmente los siguientes predicados.

a) `esPrimo`: que dado un número verifica si cumple las propiedades de ser un número primo.

b) `esPosicionValida`: que dado un entero i y una secuencia l, verifica si i es un índice válido para l.

c) `esMinimo`: que dado una secuencia de enteros l y un entero $elem$, verifica que $elem$ sea el mínimo.

d) `esMaximo`: que dado una secuencia de enteros l y un entero $elem$, verifica que $elem$ sea el máximo.

Ejercicio 2. ★ Especificar semiformalmente los siguientes problemas. Recuerde que es recomendable descomponer un problema en otros problemas más sencillos (en caso de ya haber especificado los subproblemas no es necesario especificarlos nuevamente).

a) `min`: que dado dos enteros devuelve el menor entre ellos.

b) `max`: que dado dos enteros devuelve el mayor entre ellos.

c) `elMayorPrimo`: que dado dos números primos devuelve el mayor entre ambos.

d) `buscar`: que dado un entero $elem$ y una secuencia de enteros l que incluye a $elem$, devuelva una posición donde esté $elem$.

e) `buscarMinimo`: que dado una secuencia de enteros devuelva la posición donde se encuentra el mínimo.

f) `#apariciones`: que dado un entero n y una secuencia de enteros l devuelva la cantidad de veces que aparece n en l.

g) `ordenadaCrecientemente`: que dada una secuencia de enteros sin repetidos, verifique si la lista está ordenada crecientemente.

h) `elMásRepetido`: que dada una secuencia de enteros devuelva el valor que más apariciones tiene.

i) `borrar`: que dada una secuencia de enteros sin repetidos y un elemento $elem$, devuelva el resultado de borrar $elem$ (preservando el resto de posiciones intactas).

Ejercicio 3. ★ Las siguientes especificaciones formales no son correctas. Indicar por qué, y corregirlas para que describan correctamente el problema.

a) `buscar:` Dada una secuencia y un elemento de ésta, devuelve en *resultado* alguna posición de la secuencia en la cual se encuentre el elemento.

`problema buscar` (l: $seq\langle\mathbb{R}\rangle$, elem: $\mathbb{R}$) : $\mathbb{Z}$ {
    `requiere:` $\{elem \in l\}$
    `asegura:` $\{l[resultado] = elem\}$
}

b) `minimo:` Dada una secuencia de enteros l, devuelve en *result* el menor elemento de l.

`problema minimo` (l: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ {
    `requiere:` $\{True\}$
    `asegura:` $\{(\forall y : \mathbb{Z})((y \in l \land y \neq x) \rightarrow y > result)\}$
}

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

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

  `requiere`: $\{True\}$

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

}

Ejercicio 4. La siguiente especificación informal es una resolución válida del Ejercicio 2 item (d) `buscar`.

`problema buscar` (l: $seq\langle\mathbb{R}\rangle$, elem: $\mathbb{R}$) : $\mathbb{Z}$ {
  `requiere`: $\{elem$ pertenece a $l\}$
  `asegura`: $\{resultado$ es una posición de l donde esta el elemento $elem\}$
}

Compare esta especificación con la versión formal corregida del ejercicio 3 (a). Considerando la definición de especificación, ¿qué diferencia hay entre una especficicación semiformal y una formal? Ejemplifique con el ejercicio `buscar`.

Ejercicio 5. La siguiente no es una especificación válida, ya que para ciertos valores de entrada que cumplen la precondición, no existe una salida que cumpla con la postcondición.

`problema elementosQueSumen` (l: $seq\langle\mathbb{Z}\rangle$, suma:$\mathbb{Z}$) : $seq\langle\mathbb{Z}\rangle$ {
  `requiere`: $\{True\}$
  `asegura`: {
  /* La secuencia result está incluída en la secuencia l*/
  $(\forall x : \mathbb{Z})(x \in result \rightarrow \#apariciones(x, result) \leq \#apariciones(x, l))$
  /* La suma de la lista result coincide con el valor suma */
  $\land\ suma = \sum_{i=0}^{|result|-1} result[i]$
  }
}

a) Mostrar valores para l y $suma$ que hagan verdadera la precondición, pero tales que no exista $result$ que cumpla la postcondición.

b) Supongamos que agregamos a la especificación la siguiente cláusula y las especificaciones de los subproblemas que usa la cláusula:

`requiere`: $\{min\_suma(l) \leq suma \leq max\_suma(l)\}$

problema `min_suma` (l :$seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ {

  `requiere`: $\{True\}$

  `asegura`: $\{resultado = \sum_{i=0}^{|l|-1}$ if $l[i] < 0$ then $l[i]$ else 0 fi$\}$

}

problema `max_suma` (l :$seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ {

  `requiere`: $\{True\}$

  `asegura`: $\{resultado = \sum_{i=0}^{|l|-1}$ if $l[i] > 0$ then $l[i]$ else 0 fi$\}$

}

¿Ahora es una especificación válida? Si no lo es, justificarlo con un ejemplo como en el punto anterior.

c) Dar una precondición en lengaje semiformal que haga correcta la especificación.

Ejercicio 6. ★ Para los siguientes problemas, dar todas las soluciones posibles a las entradas dadas:

a) `problema` `raizCuadrada` (x: $\mathbb{R}$) : $\mathbb{R}$ {

`requiere`: $\{x \geq 0\}$

`asegura`: $\{resultado^2 = x\}$

}

I) $x = 0$

II) $x = 1$

III) $x = 27$

b) `problema` `indiceDelMaximo` (l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{Z}$ {

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

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

}

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) `problema` `indiceDelPrimerMaximo` (l: $seq\langle\mathbb{R}\rangle$) : $\mathbb{Z}$ {

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

`asegura`: {
$0 \leq result < |l|$
$\wedge_L ((\forall i : \mathbb{Z})((0 \leq i < |l| \wedge i \neq resultado) \rightarrow_L (l[i] < l[resultado] \vee (l[i] = l[resultado] \wedge i \geq resultado))))$
}

}

I) $l = \langle 1, 2, 3, 4\rangle$

II) $l = \langle 15.5, -18, 4.215, 15.5, -1\rangle$

III) $l = \langle 0, 0, 0, 0, 0, 0\rangle$

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

Ejercicio 7. ★ Sea $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ definida como:

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

¿Cuáles de las siguientes especificaciones son correctas para el problema de calcular $f(a, b)$?
Para las que no lo son, indicar por qué.

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

`requiere`: $\{True\}$

`asegura`: $\{(a < 0 \wedge resultado = 2 * b) \wedge (a \geq 0 \wedge resultado = b - 1)\}$

}

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

`requiere`: $\{True\}$

`asegura`: $\{(a < 0 \wedge resultado = 2 * b) \vee (a > 0 \wedge resultado = b - 1)\}$

}

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

    `requiere:` $\{True\}$

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

}

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

    `requiere:` $\{True\}$

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

}

e) `problema f` (a, b: $\mathbb{R}$) : $\mathbb{R}$ {

    `requiere:` $\{True\}$

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

}

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

    `requiere:` $\{True\}$

    `asegura:` $\{resultado = (\textsf{if } a < 0 \textsf{ then } 2 * b \textsf{ else } b - 1 \textsf{ fi})\}$

}

Ejercicio 8. ★ Considerar la siguiente especificación, junto con un algoritmo que dado x devuelve x^2.

`problema unoMasGrande` (x: $\mathbb{R}$) : $\mathbb{R}$ {

    `requiere:` $\{True\}$

    `asegura:` $\{resultado > x\}$

}

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.

Ejercicio 9. ★ Sean x y r variables de tipo $\mathbb{R}$. Considerar los siguientes predicados:

P1: $\{x \leq 0\}$	Q1: $\{r \geq x^2\}$
P2: $\{x \leq 10\}$	Q2: $\{r \geq 0\}$
P3: $\{x \leq -10\}$	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 E1:

`problema hagoAlgo` (x: $\mathbb{R}$) : $\mathbb{R}$ {

    `requiere:` $\{x \leq 0\}$

    `asegura:` $\{res \geq x^2\}$

}

d) Sea A un algoritmo que cumple con la especificación E1 del ítem anterior. Decidir si necesariamente cumple las siguientes especificaciones:

a) Pre: $\{x \leq -10\}$, Post: $\{res \geq x^2\}$

b) Pre: $\{x \leq 10\}$, Post: $\{res \geq x^2\}$

c) Pre: $\{x \leq 0\}$, Post: $\{res \geq 0\}$

d) Pre: $\{x \leq 0\}$, Post: $\{res = x^2\}$

e) Pre: $\{x \leq -10\}$, Post: $\{res \geq 0\}$

f) Pre: $\{x \leq 10\}$, Post: $\{res \geq 0\}$

g) Pre: $\{x \leq -10\}$, Post: $\{res = x^2\}$

h) Pre: $\{x \leq 10\}$, Post: $\{res = x^2\}$

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

Ejercicio 10. ★ Considerar las siguientes dos especificaciones, junto con un algoritmo a que satisface la especificación de `p2`.

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

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

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`?

Ejercicio 11. Considerar las siguientes especificaciones:

`problema n-esimo1` (l: $seq\langle\mathbb{Z}\rangle$, n: $\mathbb{Z}$) : $\mathbb{Z}$ {
    `requiere:` {La longitud de la secuencia sea mayor a 0}
    `requiere:` {Los elementos están ordenados crecientemente}
    `requiere:` {n es mayor o igual a cero y menor que la longitud de l}
    `asegura:` {*resultado* es el valor n-esimo de la lista l}
}

`problema n-esimo2` (l: $seq\langle\mathbb{Z}\rangle$, n: $\mathbb{Z}$) : $\mathbb{Z}$ {
    `requiere:` {La longitud de la secuencia sea mayor a 0}
    `requiere:` {Los elementos son distintos entre sí}
    `requiere:` {n es mayor o igual a cero y menor que longitud de l}
    `asegura:` {*resultado* pertenece a l}
    `asegura:` {La cantidad de elementos de l que son menores a resultado es igual a n}
}

¿Es cierto que todo algoritmo que cumple con n-esimo1 cumple también con n-esimo2? ¿Y al revés?
Sugerencia: Razonar de manera análoga a la del ejercicio anterior.

Ejercicio 12. Especificar los siguientes problemas semiformalmente:

a) ★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.

b) Dada una secuencia de números reales, obtener la diferencia máxima entre dos de sus elementos.

Ejercicio 13. Especificar semiformalmente los siguientes problemas sobre secuencias:

a) Dadas dos secuencias s y t, decidir si s es una subcadena de t.

b) ★ 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.

c) problema mezclarOrdenado$(s, t : seq\langle\mathbb{Z}\rangle) : seq\langle\mathbb{Z}\rangle$, que recibe dos secuencias ordenadas y devuelve el resultado de intercalar sus elementos de manera ordenada.

d) ★Dado una secuencia l y un entero n, devolver la secuencia resultante de multiplicar solamente los valores pares por n.

e) Dada una secuencia de números enteros, devolver la secuencia que resulta de borrar los valores múltiplos de 3.