Metodología de la Programación

Grado en Ingeniería Informática de Gestión y Sistemas de Información
Escuela de Ingeniería de Bilbao (UPV/EHU)
Departamento de Lenguajes y Sistemas Informáticos
Curso: 1^o
Curso académico: 2020-2021
Grupo 01
Tema 5: Especificación ecuacional de los tipos abstractos de datos (TAD)
2,5 puntos

06-05-2021

Enunciado

Índice

Lista de figuras

Lista de tablas

****

1 Especificación ecuacional — Listas (*0,750 puntos*)

1.1 Especificación ecuacional de la función $selec$ (0,550 puntos)

Especificar ecuacionalmente la función $selec$ que, dadas una lista de enteros y una lista de booleanos, devuelve la lista que se obtiene seleccionando, de la primera lista, los elementos que ocupan las posiciones que tienen el valor $True$ en la segunda lista. Si las listas de entrada son de distinta longitud, se ha de devolver un mensaje de error. Si las dos listas son vacías, se ha de devolver la lista vacía.

Además de las operaciones constructoras [ ] y (:), se pueden utilizar las siguientes funciones auxiliares sin necesidad de definirlas:

- es_vacia, que dada una lista, devuelve $True$ si la lista es vacía y $False$ en caso contrario.
- $longitud$, que dada una lista, devuelve el número de elementos de la lista.
- $primero$, que dada una lista, devuelve el primer elemento de la lista (el de la esquina izquierda).
- $resto$, que dada una lista, devuelve la lista que se obtiene al quitar el primer elemento de la lista (el de la esquina izquierda).

Ejemplos

- $selec([8,7,5,9,7],[True,False,False,True,False]) = [8,9]$
  La lista resultado contiene los elementos de la primera posición y de la cuarta posición de la primera lista porque la segunda lista tiene el valor $True$ en esas dos posiciones.
- $selec([8,7,5],[False,False,False]) = [\,]$
  Puesto que en todas las posiciones de la segunda lista se tiene el valor $False$, el resultado es la lista vacía.
- $selec([8,7,5],[True,True,True]) = [8,7,5]$
  En todas las posiciones de la segunda lista se tiene el valor $True$ y, por tanto, el resultado es la primera lista.
- $selec([\,],[\,]) = [\,]$

1.2 Desarrollo de un ejemplo para la función $selec$ (0,200 puntos)

Una vez dadas las ecuaciones, desarrollar paso a paso el siguiente ejemplo. En cada paso hay que indicar qué ecuación se ha utilizado:

$$selec([8,7,5,9,7],[True,False,False,True,False])$$

2 Inducción sobre listas (*1,000 puntos*)

2.1 Especificación ecuacional de la función $longitud$ (0,100 puntos)

Especificar ecuacionalmente la función $longitud$ que, dada una lista de tipo t, devuelve el número de elementos de la lista.

$$longitud :: ([t]) \to Int$$

Ejemplos

- ▷ $longitud([8,5,9,5]) = 4$
- ▷ $longitud([80]) = 1$
- ▷ $longitud([\,]) = 0$

2.2 Especificación ecuacional de la función $eliminar$ (0,100 puntos)

Especificar ecuacionalmente la función $eliminar$ que, dados un elemento de tipo t y una lista de tipo $[t]$, devuelve la lista que se obtiene eliminando, de la lista dada, todas las apariciones del elemento dado.

$$eliminar :: (t, [t]) \to [t]$$

Ejemplos

- ▷ $eliminar(6, [5,6,20,6,6,10,34]) = [5,20,10,34]$
- ▷ $eliminar(8, [5,20,34]) = [5,20,34]$
- ▷ $eliminar(8, [8,8,8]) = [\,]$
- ▷ $eliminar(15, [\,]) = [\,]$

2.3 Especificación ecuacional de la función $++$ (0,100 puntos)

Especificar ecuacionalmente la función $++$ que, dadas dos listas de tipo $[t]$, devuelve la lista que se obtiene concatenando las dos listas dadas.

$$++ :: ([t], [t]) \to [t]$$

Ejemplos

- ▷ $[1,8] ++ [4,0,8,7] = [1,8,4,0,8,7]$
- ▷ $[\,] ++ [4,0,8,7] = [4,0,8,7]$
- ▷ $[1,8] ++ [\,] = [1,8]$
- ▷ $[\,] ++ [\,] = [\,]$

2.4 Prueba por inducción (0,100 + 0,600 puntos)

Probar por inducción que para cualquier elemento x (de tipo t) y cualquier par de listas s y r (de tipo $[t]$), se cumple la siguiente propiedad.

$$longitud(eliminar(x, s++r)) = longitud(eliminar(x, s)) + longitud(eliminar(x, r))$$

La inducción debe realizarse sobre la lista s considerando el caso básico $s = [\,]$ y el caso inductivo $s = z : w$. La hipótesis de la inducción consistirá en suponer que para el elemento x y las listas w y r se cumple la propiedad que se está probando. Se necesitan las siguientes propiedades:

- ○ $(Prop\ 1)$ La suma es asociativa: $i + (j + k) = (i + j) + k$.
- ○ $(Prop\ 2)$ 0 es elemento neutro para la suma: $i + 0 = 0 + i = i$.

3 Especificación ecuacional — Pilas (*0,200 puntos*)

3.1 Especificación ecuacional de la función sm (0,200 puntos)

Especificar ecuacionalmente la función sm que, dadas dos pilas de enteros, realiza lo siguiente:

- □ Si las pilas tienen distinta altura, devuelve un mensaje de error.
- □ Si las pilas tienen la misma altura, devuelve la pila que se genera colocando a cada altura un valor que depende del signo de los elementos que están a la misma altura. El criterio a seguir es el siguiente:
  - ◇ Si las dos pilas son vacías, se ha de devolver la pila vacía.
  - ◇ Si no, si el elemento de la segunda pila es 0, colocar el valor -1.
  - ◇ Si no, si alguno de los dos elementos es negativo, colocar el valor -2.
  - ◇ Si no, es decir, si el elemento de la primera pila no es negativo (≥ 0) y el elemento de la segunda pila es positivo (≥ 1), colocar el resultado de la división entera entre el elemento de la primera pila y el elemento de la segunda pila. Por tanto, el elemento de la primera pila dividido por el elemento de la segunda pila.

Además de las operaciones constructoras $Pvacia$ y $Apilar$, se pueden utilizar las siguientes funciones auxiliares sin necesidad de definirlas:

- ⬡ es_pvacia, que dada una pila, devuelve $True$ si la pila es vacía y $False$ en caso contrario.
- ⬡ $altura$, que dada una pila, devuelve el número de elementos de la pila.
- ⬡ $cima$, que dada una pila, devuelve el elemento que está más arriba.
- ⬡ $desapilar$, que dada una pila, devuelve la pila que se obtiene al quitar el elemento de la cima (el que está más arriba).

Ejemplos

- ▷ En este primer ejemplo, se tienen dos pilas de altura 9 representadas gráficamente. En la pila resultado se indica en qué casos se ha realizado la división entera.

	12		3		4	$12\ div\ 3$
	2		2		1	$2\ div\ 2$
	-7		10		-2	
	5		8		0	$5\ div\ 8$
	20		2		10	$20\ div\ 2$
	-3		0		-1	
	32		4		8	$32\ div\ 4$
	-6		-4		-2	
$sm($	15	,	0	$) =$	-1	

- ▷ En este segundo ejemplo, se tienen dos pilas de altura 2 representadas mediante expresiones formadas con las dos operaciones constructoras: $Pvacia$ y $Apilar$. En la altura 2 se realiza la división $12\ div\ 3$, es decir, 4. En la altura 1, estamos en uno de los casos en los que no hay que realizar la división: el elemento de la segunda pila no es 0, pero el de la primera pila es negativo y, consecuentemente, se ha puesto -2.

$$sm(Apilar(12, Apilar(-10, Pvacia)), Apilar(3, Apilar(11, Pvacia))) =$$

$$Apilar(4, Apilar(-2, Pvacia))$$

4 Especificación ecuacional — Colas (*0,150 puntos*)

4.1 Especificación ecuacional de la función edt (0,150 puntos)

Especificar ecuacionalmente la función edt que, dados un número entero y una cola de enteros, devuelve la cola que se obtiene eliminando los elementos de dicha cola que son distintos al número dado. Si la cola inicial es vacía, se ha de devolver la cola vacía.

Ejemplos

- ▷ $edt(8, << 5, 7, 8, 8, 1, 8 >>) =<< 8, 8, 8 >>$
- ▷ $edt(8, << 5, 7, 1 >>) =<< >>$
- ▷ $edt(8, << 7, 8, 4 >>) =<< 8 >>$
- ▷ $edt(8, Poner(Poner(Poner(Cvacia, 7), 8), 4)) = Poner(Cvacia, 8)$

5 Especificación ecuacional — Árboles binarios (*0,400 puntos*)

5.1 Especificación ecuacional de la función riq (0,400 puntos)

Especificar ecuacionalmente la función riq que, dados un valor de tipo t y un árbol binario de tipo t, devuelve el árbol binario que se obtiene sustituyendo por el valor dado, los valores de los nodos de la rama que está más a la izquierda en el árbol de entrada. Si el árbol de entrada es vacío, se ha de devolver el árbol vacío.

Los nodos de la rama que está más a la izquierda son la raíz y todos los nodos que se encuentran en el camino entre la raíz y la hoja que está más a la izquierda. La hoja que está más a la izquierda es aquella a la cual se accede desde la raíz eligiendo, siempre que se pueda, el camino que va por el subárbol izquierdo correspondiente. Solo en caso de que no haya subárbol izquierdo pero sí subárbol derecho, se sigue por el subárbol derecho.

Además de las operaciones constructoras $Avacio$ y $Crear$, se puede utilizar la función es_avacio que decide si un árbolbinario es vacío o no, devolviendo $True$ o $False$ según el caso.

Ejemplos

- ▷ En la figura 1 de la página 6, se muestra la representación gráfica —o diagrama— de un árbol binario al que se le ha llamado a. En esa figura, los nodos que pertenecen a la rama que está más a la izquierda del árbol binario a aparecen marcados con el color amarillo. En la figura 2 de la página 6, se muestra el diagrama del árbol binario $riq(18, a)$, es decir, el árbol binario que se ha obtenido a partir den árbol binario a sustituyendo por 18 el valor de los nodos que conforman la rama que está más a la izquierda.
- ▷ En la figura 3 de la página 7, se muestra el diagrama de un árbol binario al que se le ha llamado b. En esa figura, los nodos que pertenecen a la rama que está más a la izquierda del árbol binario b aparecen marcados con el color amarillo. En la figura 4 de la página 7, se muestra el diagrama del árbol binario $riq(40, b)$, es decir, el árbol binario que se ha obtenido a partir den árbol binario b sustituyendo por 40 el valor de los nodos que conforman la rama que está más a la izquierda.
- ▷ En la tabla 1 de la página 7, se muestra la aplicación de la función riq al número 18 y a un árbol binario representado de manera formal, es decir, utilizando las operaciones constructoras $Avacio$ y $Crear$. Después de la igualdad (=) se presenta el árbol binario resultante.

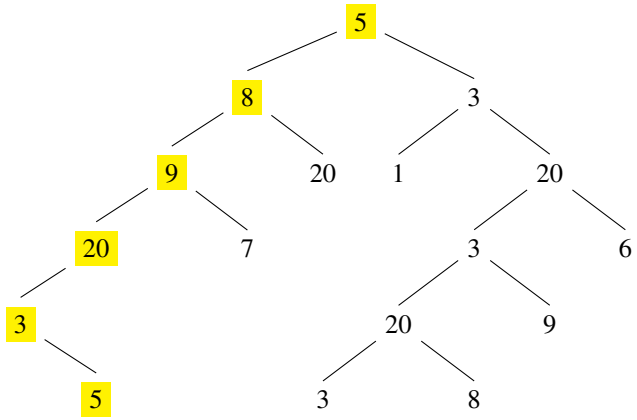

Figura 1: Árbol binario a. Los nodos que forman la rama más a la izquierda están señalados en amarillo.

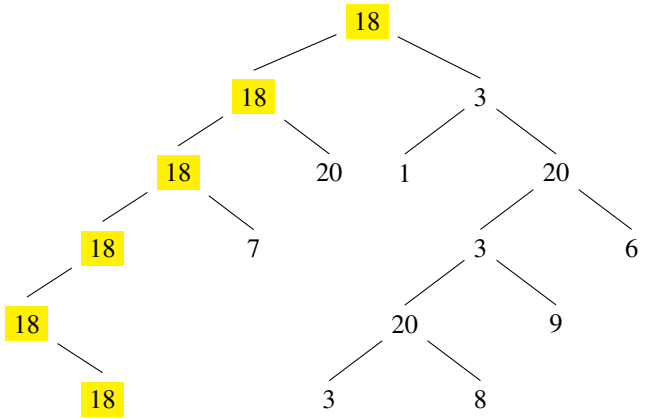

Figura 2: Árbol binario $riq(18, a)$. Los nodos que han sido modificados con respecto al árbol binario a de la figura 1 están señalados en amarillo.

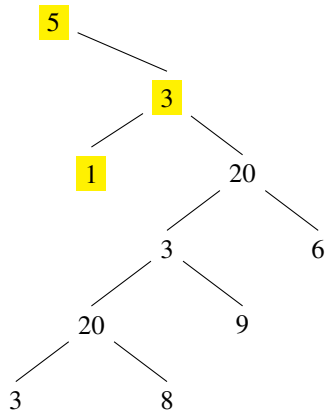

Figura 3: Árbol binario b. Los nodos que forman la rama más a la izquierda están señalados en amarillo.

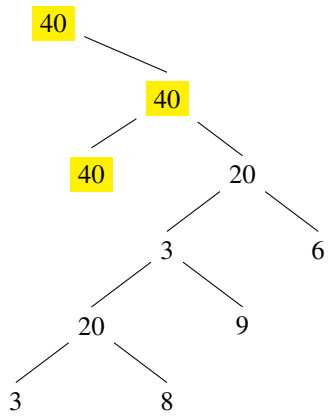

Figura 4: Árbol binario $riq(40, b)$. Los nodos que han sido modificados con respecto al árbol binario b de la figura 3 están señalados en amarillo.

$$
\begin{array}{l}
riq(18, Crear(\,1\,, Crear(\,2\,, Avacio, Avacio), \\
\qquad\qquad\qquad Crear(3, Crear(4, Avacio, Avacio), \\
\qquad\qquad\qquad\qquad Avacio \\
\qquad\qquad\qquad\qquad) \\
\qquad\qquad\qquad) \\
\quad) \\
= \\
\qquad Crear(\,18\,, Crear(\,18\,, Avacio, Avacio), \\
\qquad\qquad\qquad Crear(3, Crear(4, Avacio, Avacio), \\
\qquad\qquad\qquad\qquad Avacio \\
\qquad\qquad\qquad\qquad) \\
\qquad\qquad\qquad)
\end{array}
$$

Tabla 1: Aplicación de la función riq al número 18 y a un árbol binario. Después de la igualdad, se muestra el árbol resultante.