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: 2021-2022
Grupo 01
Tema 2: Especificación y documentación formal de programas
2 puntos

21-02-2022

Solución

Índice

Lista de figuras

Lista de tablas

****

1 Primer programa: Simulación de la conjunción y la disyunción con los vectores $A(1..n)$ y $B(1..n)$ que solo contienen ceros y unos (*1 punto*)

1.1 Enunciado

Es habitual representar los valores Booleanos $False$ y $True$ mediante los números enteros 0 y 1 respectivamente. Si utilizamos 0 y 1 en vez de $False$ y $True$, la conjunción ($\wedge$) y la disyunción ($\vee$) se pueden simular utilizando el producto ($*$), la suma ($+$) y el resto de la división entera (mod):

conjunción	$\delta \ \wedge \ \rho = \delta * \rho$
disyunción	$\delta \ \vee \ \rho = (\delta * \rho) + ((\delta + \rho) \ mod \ 2)$
	donde δ (delta) y ρ (ro) son dos valores del conjunto $\{0, 1\}$.

En la figura 1 de la página 4 se muestran los cuatro casos posibles para las expresiones $(\delta * \rho)$ y $((\delta * \rho) + ((\delta + \rho) \ mod \ 2))$.

Se pide documentar el programa que se muestra en la figura 2. Dicho programa realiza lo siguiente:

- Recibe como datos de entrada dos vectores de enteros $A(1..n)$ y $B(1..n)$ no vacíos que contienen solo ceros y unos.
- Devuelve como resultado, los vectores $A(1..n)$ y $B(1..n)$ modificados de tal forma que para cada posición k comprendida entre 1 y n, se tendrá:
  - en $A(k)$, el producto del valor inicial de $A(k)$ y del valor inicial de $B(k)$;
  - en $B(k)$, la suma de los siguientes dos elementos:
    - el producto del valor inicial de $A(k)$ y del valor inicial de $B(k)$;
    - el resto que se genera al dividir por 2 la suma del valor inicial de $A(k)$ y del valor inicial de $B(k)$.

En el programa de la figura 2, mod representa el resto de la división entera. Ejemplos: $19 \ mod \ 2 = 1$; $19 \ mod \ 3 = 1$; $19 \ div \ 4 = 3$; $17 \ div \ 3 = 2$; $8 \ mod \ 12 = 8$; $20 \ mod \ 5 = 0$.

1.2 Solución

La especificación correspondiente al programa que se muestra en la figura 2 de la página 4 se encuentra en la tabla 1 de la página 5.

La documentación correspondiente al programa que se muestra en la figura 2 de la página 4 se encuentra en la tabla 2 de la página 6. En la documentación se utilizará c_k para abreviar la expresión $(a_k * b_k) + ((a_k + b_k) \ mod \ 2)$. Ver tabla 2 de la página 6.

Siendo δ (delta) y ρ (ro) dos valores del conjunto $\{0, 1\}$, los cuatro casos posibles para las expresiones $(\delta * \rho)$ y $((\delta * \rho) + ((\delta + \rho)\ mod\ 2))$ son los siguientes:

δ	ρ	$\delta * \rho$	$(\delta + \rho)$	$((\delta + \rho)\ mod\ 2)$	$(\delta * \rho) + ((\delta + \rho)\ mod\ 2)$
0	0	0	0	0	0
0	1	0	1	1	1
1	0	0	1	1	1
1	1	1	2	0	1

Figura 1: Los cuatro casos posibles para las expresiones $(\delta * \rho)$ y $((\delta * \rho) + ((\delta + \rho)\ mod\ 2))$.

(1) {Precondición}	(1) (0,080 puntos)
i := 0;	
(2) {Aserción intermedia}	(2) (0,005 puntos)
while (3) {Invariante} (4) {Expresión cota E} i $\neq$ n **loop**	(3) (0,200 puntos); (4) (0,015 puntos)
(5) {Aserción intermedia}	(5) (0,010 puntos)
aux := A(i + 1) + B(i + 1);	
(6) {Aserción intermedia}	(6) (0,010 puntos)
A(i + 1) := A(i + 1) $*$ B(i + 1);	
(7) {Aserción intermedia}	(7) (0,150 puntos)
B(i + 1) := A(i + 1);	
(8) {Aserción intermedia}	(8) (0,150 puntos)
i := i + 1;	
(9) {Aserción intermedia}	(9) (0,150 puntos)
B(i) := B(i) + (aux mod 2);	
(10) {Aserción intermedia}	(10) (0,150 puntos)
end loop;	
(11) {Postcondición}	(11) (0,080 puntos)

Figura 2: Primer programa: Simulación de la conjunción y la disyunción con los vectores $A(1..n)$ y $B(1..n)$ que solo contienen ceros y unos.

	Especificación: Precondición (1) y postcondición (11)
(1)	$n \geq 1$ $\wedge \forall k(1 \leq k \leq n \rightarrow A(k) = a_k)$ $\wedge \forall k(1 \leq k \leq n \rightarrow B(k) = b_k)$ $\wedge \forall k(1 \leq k \leq n \rightarrow (a_k = 0 \ \vee \ a_k = 1))$ $\wedge \forall k(1 \leq k \leq n \rightarrow (b_k = 0 \ \vee \ b_k = 1))$
(11)	$\boldsymbol{\gamma}$ $\wedge \forall k(1 \leq k \leq n \rightarrow A(k) = a_k * b_k)$ $\wedge \forall k(1 \leq k \leq n \rightarrow B(k) = (a_k * b_k) + ((a_k + b_k) \ mod \ 2))$

Se ha utilizado la siguiente abreviatura (gamma):

$$\boldsymbol{\gamma} \equiv n \geq 1 \wedge \forall k(1 \leq k \leq n \rightarrow (a_k = 0 \ \vee \ a_k = 1)) \wedge \forall k(1 \leq k \leq n \rightarrow (b_k = 0 \ \vee \ b_k = 1))$$

En la documentación se utilizará c_k para abreviar la expresión $(a_k * b_k) + ((a_k + b_k) \ mod \ 2)$. Ver tabla 2 de la página 6.

Tabla 1: Especificación correspondiente al programa que se muestra en la figura 2 de la página 4 (Simulación de la conjunción y la disyunción con los vectores $A(1..n)$ y $B(1..n)$ que solo contienen ceros y unos).

2 Segundo programa: Suma de los números que pertenecen al intervalo $[r..w]$ y tienen q divisores (*1 punto*)

2.1 Enunciado

(a) (*0,010 puntos*) Definir el predicado $divisor(x, y)$ que exprese que el número positivo x es un divisor del número no negativo y. Por tanto, el predicado ha de expresar que x es mayor o igual que 1, que y es mayor o igual que 0 y que el resto de dividir y por x es 0.

Ejemplos:

$$divisor(3, 15) = True \qquad divisor(15, 3) = False$$
$$divisor(4, 9) = False \qquad divisor(9, 4) = False$$

(b) (*0,100 puntos*) Definir el predicado $cantidad\_div(z, h)$ que exprese que el número positivo z tiene h divisores. Por tanto, el predicado ha de expresar que z es mayor o igual que 1 y que la cantidad de divisores de z que pertenecen al intervalo $[1..z]$ es h. Al definir este predicado, hay que utilizar el predicado del apartado anterior.

Ejemplos:

$cantidad\_div(7, 2) = True$, porque 7 tiene exactamente 2 divisores (el 1 y el 7).
$cantidad\_div(7, 4) = False$, porque 7 no tiene 4 divisores (tiene 2).
$cantidad\_div(10, 4) = True$, porque 10 tiene exactamente 4 divisores (el 1, el 2, el 5 y el 10).
$cantidad\_div(10, 1) = False$, porque 1 no es la cantidad exacta de divisores de 10 (la cantidad exacta es 4).

	Documentación: Invariante (3), expresión cota (4) y aserciones intermedias (2, 5, 6, 7, 8, 9 y 10)
(2)	$(1) \land i = 0$
(3)	$\boldsymbol{\gamma} \land (0 \leq i \leq n)$ $\land \forall k(1 \leq k \leq i \rightarrow A(k) = a_k * b_k) \land \forall k(i+1 \leq k \leq n \rightarrow A(k) = a_k)$ $\land \forall k(1 \leq k \leq i \rightarrow B(k) = c_k) \land \forall k(i+1 \leq k \leq n \rightarrow B(k) = b_k)$
(4)	$E = n - i$
(5)	$\boldsymbol{\gamma} \land (0 \leq i \leq n-1)$ $\land \forall k(1 \leq k \leq i \rightarrow A(k) = a_k * b_k) \land \forall k(i+1 \leq k \leq n \rightarrow A(k) = a_k)$ $\land \forall k(1 \leq k \leq i \rightarrow B(k) = c_k) \land \forall k(i+1 \leq k \leq n \rightarrow B(k) = b_k)$
(6)	$\boldsymbol{\gamma} \land (0 \leq i \leq n-1) \land aux = A(i+1) + B(i+1)$ $\land \forall k(1 \leq k \leq i \rightarrow A(k) = a_k * b_k) \land \forall k(i+1 \leq k \leq n \rightarrow A(k) = a_k)$ $\land \forall k(1 \leq k \leq i \rightarrow B(k) = c_k) \land \forall k(i+1 \leq k \leq n \rightarrow B(k) = b_k)$
(7)	$\boldsymbol{\gamma} \land (0 \leq i \leq n-1) \land aux = a_{i+1} + B(i+1)$ $\land \forall k(1 \leq k \leq i+1 \rightarrow A(k) = a_k * b_k) \land \forall k(i+2 \leq k \leq n \rightarrow A(k) = a_k)$ $\land \forall k(1 \leq k \leq i \rightarrow B(k) = c_k) \land \forall k(i+1 \leq k \leq n \rightarrow B(k) = b_k)$
(8)	$\boldsymbol{\gamma} \land (0 \leq i \leq n-1) \land aux = a_{i+1} + b_{i+1}$ $\land \forall k(1 \leq k \leq i+1 \rightarrow A(k) = a_k * b_k) \land \forall k(i+2 \leq k \leq n \rightarrow A(k) = a_k)$ $\land \forall k(1 \leq k \leq i \rightarrow B(k) = c_k) \land \forall k(i+2 \leq k \leq n \rightarrow B(k) = b_k)$ $\land B(i+1) = a_{i+1} * b_{i+1}$
(9)	$\boldsymbol{\gamma} \land (1 \leq i \leq n) \land aux = a_i + b_i$ $\land \forall k(1 \leq k \leq i \rightarrow A(k) = a_k * b_k) \land \forall k(i+1 \leq k \leq n \rightarrow A(k) = a_k)$ $\land \forall k(1 \leq k \leq i-1 \rightarrow B(k) = c_k) \land \forall k(i+1 \leq k \leq n \rightarrow B(k) = b_k)$ $\land B(i) = a_i * b_i$
(10)	$\boldsymbol{\gamma} \land (1 \leq i \leq n) \land aux = a_i + b_i$ $\land \forall k(1 \leq k \leq i \rightarrow A(k) = a_k * b_k) \land \forall k(i+1 \leq k \leq n \rightarrow A(k) = a_k)$ $\land \forall k(1 \leq k \leq i \rightarrow B(k) = c_k) \land \forall k(i+1 \leq k \leq n \rightarrow B(k) = b_k)$

Se han marcado en amarillo los cambios que hay en cada fórmula con respecto a la anterior fórmula. El punto previo al (5) es el (3).
Se ha utilizado la abreviatura $\boldsymbol{\gamma}$ (gamma) definida en la tabla 1 de la página 5.
Se ha utilizado c_k para abreviar la expresión $(a_k * b_k) + ((a_k + b_k)\ mod\ 2)$.

Tabla 2: Documentación correspondiente al programa que se muestra en la figura 2 de la página 4 (Simulación de la conjunción y la disyunción con los vectores $A(1..n)$ y $B(1..n)$ que solo contienen ceros y unos).

(c) (*0,890 puntos*) Documentar el programa que se muestra en la figura 3. Para ello, hay que utilizar el predicado definido en el apartado anterior siempre que sea posible. El programa de la figura 3 realiza lo siguiente:

- Recibe como datos de entrada tres números enteros q, r y w, donde q es mayor o igual que 1, r es mayor o igual que q y w es mayor o igual que r.
- Devuelve, en la variable $suma$, la suma de los elementos del intervalo $[r..w]$ que tienen q divisores.

La función $tiene\_div$ que aparece en el programa de la figura 3, es una implementación del predicado $cantidad\_div$ del apartado *(b)*. La función $tiene\_div$ estará escrito en el lenguaje de programación que se está utilizando, mientras que el predicado $cantidad\_div$ estará escrito en el lenguaje de la lógica de primer orden (o lógica de predicados).

2.2 Solución

La definición del predicado $divisor(x, y)$ del apartado (a) se encuentra en la tabla 3 de la página 8.

La definición del predicado $cantidad\_div(z, h)$ del apartado (b) se encuentra en la tabla 4 de la página 8.

La especificación y la documentación correspondientes al programa mostrado en la figura 3 de la página 8 se encuentran en la tabla 5 de la página 9.

(1) {Precondición}	(1) (0,020 puntos)
i := r;	
(2) {Aserción intermedia}	(2) (0,005 puntos)
suma := 0;	
(3) {Aserción intermedia}	(3) (0,005 puntos)
while (4) {Invariante} (5) {Expresión cota E} i $\leq$ w **loop**	(4) (0,250 puntos); (5) (0,020 puntos)
(6) {Aserción intermedia}	(6) (0,010 puntos)
i := i + 1;	
(7) {Aserción intermedia}	(7) (0,210 puntos)
if tiene_div(i − 1, q) **then**	
(8) {Aserción intermedia}	(8) (0,010 puntos)
suma := suma + (i − 1);	
(9) {Aserción intermedia}	(9) (0,210 puntos)
end if;	
(10) {Aserción intermedia}	(10) (0,050 puntos)
end loop;	
(11) {Postcondición}	(11) (0,100 puntos)

Figura 3: Segundo programa: Suma de los números que pertenecen al intervalo $[r..w]$ y tienen q divisores.

$$divisor(x, y) \equiv \quad (x \geq 1) \ \wedge \ (y \geq 0) \ \wedge \ (y \ mod \ x = 0)$$

Tabla 3: Definición del predicado $divisor(x, y)$.

$$cantidad\_div(z, h) \equiv \quad (z \geq 1) \ \wedge \ h = \mathcal{N}k(1 \leq k \leq z \ \wedge \ divisor(k, z))$$

Tabla 4: Definición del predicado $cantidad\_div(z, h)$.

	Especificación: Precondición (1) y postcondición (11)
(1)	$(q \geq 1) \wedge (r \geq q) \wedge (w \geq r) \wedge (q = a) \wedge (r = b) \wedge (w = c)$
(11)	$(1) \wedge suma = \sum_{\substack{(r \leq k \leq w \wedge \\ cantidad\_div(k,q))}} k$

	Documentación: Invariante (4), expresión cota (5) y aserciones intermedias (2, 3, 6, 7, 8, 9 y 10)
(2)	$(1) \wedge i = r$
(3)	$(2) \wedge suma = 0$
(4)	$(1) \wedge (r \leq i \leq w+1) \wedge (suma = \sum_{\substack{(r \leq k \leq i-1 \wedge \\ cantidad\_div(k,q))}} k)$
(5)	$E = w + 1 - i$
(6)	$(1) \wedge (r \leq i \leq \boxed{w}) \wedge (suma = \sum_{\substack{(r \leq k \leq i-1 \wedge \\ cantidad\_div(k,q))}} k)$
(7)	$(1) \wedge (\boxed{r+1} \leq i \leq \boxed{w+1}) \wedge (suma = \sum_{\substack{(r \leq k \leq \boxed{i-2} \wedge \\ cantidad\_div(k,q))}} k$
(8)	$(1) \wedge (r+1 \leq i \leq w+1) \wedge (suma = \sum_{\substack{(r \leq k \leq i-2 \wedge \\ cantidad\_div(k,q))}} k)$ $\wedge \boxed{cantidad\_div(i-1,q)}$
(9)	$(1) \wedge (r+1 \leq i \leq w+1) \wedge (suma = \sum_{\substack{(r \leq k \leq \boxed{i-1} \wedge \\ cantidad\_div(k,q))}} k)$ $\wedge\, cantidad\_div(i-1,q)$
(10)	$(1) \wedge (r+1 \leq i \leq w+1) \wedge (suma = \sum_{\substack{(r \leq k \leq \boxed{i-1} \wedge \\ cantidad\_div(k,q))}} k)$

Se han marcado en amarillo los cambios que hay en cada fórmula con respecto a la anterior fórmula. El punto previo al (6) es el (4).

Tabla 5: Especificación y documentación correspondientes al programa que se muestra en la figura 3 de la página 8 (Suma de los números que pertenecen al intervalo $[r..w]$ y tienen q divisores).