

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: 1º

Tema 4: Derivación formal de programas
 1,5 puntos

Modelo de examen: t4m1- \exists

Enunciado

Última actualización: 05 - 04 - 2022

Índice

1	Derivación formal de un programa iterativo (1,5 puntos)	1
----------	--	----------

Lista de figuras

1	Estructura del programa a derivar, definiciones de φ , INV , E y ψ y definición del predicado utilizado.	3
---	---	---

Lista de tablas

1	Abreviaciones que se recomienda utilizar.	3
2	Denominaciones de las letras griegas utilizadas.	3
3	Puntuación por apartados.	4

1 Derivación formal de un programa iterativo (1,5 puntos)

Derivar, utilizando la regla del while y el axioma de la asignación del Cálculo de Hoare, un programa que, dados un número entero x y un vector no vacío de enteros $A(1..n)$ que no contiene valores menores que 20, decide en la variable booleana q si x es múltiplo de algún elemento de $A(1..n)$. El programa ha de ser derivado teniendo en cuenta la precondition y la postcondition (φ y ψ), el invariante INV y la expresión cota E . El programa obtenido ha de ser eficiente en el sentido de que si en algún momento se detecta que la respuesta va a ser afirmativa, el programa ha de parar sin analizar las posiciones restantes.

En la figura 1 de la página 3, se muestra la estructura que ha de tener el programa derivado. En esa misma figura se indica cuáles son las fórmulas φ , ψ , INV y E en las que se ha de basar la derivación. Además, se da la definición del predicado que se utiliza tanto en φ como en INV .

En el programa de la figura 1, *mod* representa el resto de la división entera. Ejemplos: $20 \bmod 3 = 2$, $18 \bmod 3 = 0$, $19 \bmod 3 = 1$. En esos tres ejemplos, la división entera, representada aquí como *div*, devolvería 6: $20 \div 3 = 6$, $18 \div 3 = 6$, $19 \div 3 = 6$. Otros ejemplos para la división entera: $19 \div 2 = 9$; $19 \div 3 = 6$; $19 \div 4 = 4$; $17 \div 3 = 5$; $8 \div 12 = 0$.

En la tabla 1 de la página 3, se recogen las abreviaciones que se recomienda utilizar durante el proceso de verificación. En la tabla 2 de la página 3, se recopilan las denominaciones de las letras griegas utilizadas en este enunciado. Finalmente, en la tabla 3 de la página 4, se muestra la puntuación de los distintos pasos o apartados que han de ser considerados en el proceso de derivación.

Todos los elementos numéricos de la figura 1 y de la tabla 1 representan números enteros. Por tanto, los valores representados por esos elementos pertenecen a \mathbb{Z} , donde el conjunto \mathbb{Z} es el siguiente:

$$\{\dots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \dots\}$$

Formalmente, $\mathbb{Z} = \mathbb{N} \cup \{-y \mid y \in \mathbb{N} \wedge y \geq 1\}$, donde $\mathbb{N} = \{0, 1, 2, 3, \dots\}$ es el conjunto de los números naturales y \cup es la unión de conjuntos.

Ejemplo 1.1. (Para el programa a derivar y cuya estructura se muestra en la figura 1) Sean $x = 500$ y el siguiente vector $A(1..8)$:

$A(1..8)$	102	50	25	94	25	27	53	72
	1	2	3	4	5	6	7	8

Para esos valores de x y $A(1..8)$, el programa a derivar —cuya estructura se muestra en la figura 1— devolvería el valor booleano *True* en q , porque x es múltiplo de al menos un elemento de $A(1..8)$. En concreto, x es múltiplo de los elementos de las posiciones 2, 3 y 5 del vector $A(1..8)$.

En cambio, si el vector $A(1..8)$ fuera el que se muestra a continuación, la respuesta debería ser *False* porque x no es múltiplo de ningún elemento de $A(1..n)$:

$A(1..8)$	102	28	21	136	74	27	62	71
	1	2	3	4	5	6	7	8

Estructura del programa a derivar:
$\{\varphi\}$ ζ Inicializaciones? while $\{INV\} \{E\} \zeta B? \text{ loop}$ ζ Instrucciones? end loop; $\{\psi\}$
Definición de φ , INV , E y ψ :
$\varphi \equiv n \geq 1 \wedge veinte(A(1..n))$ $INV \equiv n \geq 1 \wedge veinte(A(1..n)) \wedge (1 \leq i \leq n+1) \wedge (q \leftrightarrow \exists k(1 \leq k \leq i-1 \wedge x \bmod A(k) = 0))$ $E = n+1-i$ $\psi \equiv q \leftrightarrow \exists k(1 \leq k \leq n \wedge x \bmod A(k) = 0)$
Definición del predicado utilizado:
$veinte(H(1..r)) \equiv \forall k(1 \leq k \leq r \rightarrow H(k) \geq 20)$

Figura 1: Estructura del programa a derivar, definiciones de φ , INV , E y ψ y definición del predicado utilizado.

Abreviaciones recomendadas:
$\lambda \equiv n \geq 1 \wedge veinte(A(1..n))$ $\gamma(\ell) \equiv x \bmod A(\ell) = 0$ $\mu(\ell) \equiv \exists k(1 \leq k \leq \ell \wedge x \bmod A(k) = 0)$

Tabla 1: Abreviaciones que se recomienda utilizar.

Letras griegas utilizadas:
φ : fi ψ : psi γ : gamma μ : mu λ : lambda

Tabla 2: Denominaciones de las letras griegas utilizadas.

Puntuación:	
(a)	Cálculo de las inicializaciones previas al while: 0,250
(b)	Cálculo de la condición del while (B): 0,380
	(b.1) Formulación de la condición del while (B): 0,150
	(b.2) Comprobación del punto (II) de la regla del while: 0,005
	(b.3) Comprobación del punto (IV) de la regla del while: 0,200
	(b.4) Comprobación del punto (V) de la regla del while: 0,025
(c)	Cálculo de las instrucciones que van dentro del while: 0,850
	(c.1) Desarrollo relacionado con el punto (III) de la regla del while: 0,550
	(c.2) Desarrollo relacionado con el punto (VI) de la regla del while: 0,300
(d)	Escribir el programa completo al final: 0,020
■	Cuando no se explique por qué se cumple una implicación, se contará cero. Es decir, indicar que una implicación sí se cumple sin razonar por qué se cumple cuenta 0.
■	Para aprobar el ejercicio es obligatorio obtener al menos la mitad de la puntuación en los apartados (a), (b) y (c).

Tabla 3: Puntuación por apartados.