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

Tema 4: Derivación formal de programas
1,5 puntos

Modelo de examen: t4m3-∃

Enunciado

Última actualización: 05 - 04 - 2022

Índice

Lista de figuras

Lista de tablas

****

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, dado un vector de enteros positivos $A(1..n)$ que consta de al menos dos componentes, decide en la variable booleana q si $A(1..n)$ contiene, entre la posición 2 y la posición n, algún número que sea múltiplo de la posición que ocupa. El programa ha de ser derivado teniendo en cuenta la precondición y la postcondición (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 (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: $20\ mod\ 3 = 2$, $18\ mod\ 3 = 0$, $19\ mod\ 3 = 1$. 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 (página 3) se recogen las abreviaciones que se recomienda utilizar durante el proceso de derivación. En la tabla 2 (página 3) se recopilan las denominaciones de las letras griegas utilizadas en este enunciado. Finalmente, en la tabla 3 (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) *Sea el siguiente vector $A(1..8)$:*

A(1..8)	*10*	*9*	*10*	**8**	*17*	**30**	*4*	*2*
	1	*2*	*3*	**4**	*5*	**6**	*7*	*8*

Para ese vector $A(1..8)$, el programa a derivar —cuya estructura se muestra en la figura 1— devolvería el valor booleano $True$ en q, ya que el elemento de la posición 4 (el número 8) es múltiplo de la posición que ocupa (posición 4) y el elemento de la posición 6 (el número 30) es múltiplo de la posición que ocupa (posición 6). Como al menos hay un elemento de $A(1..8)$ que está entre las posiciones 2 y 8 y es múltiplo de la posición que ocupa, la respuesta ha de ser $True$.

En cambio, si el vector $A(1..8)$ fuera el que se muestra a continuación, la respuesta debería ser $False$ puesto que ningún elemento de $A(1..n)$ que está entre las posiciones 2 y 8, es múltiplo de la posición que ocupa:

A(1..8)	*2*	*5*	*2*	*7*	*3*	*39*	*18*	*3*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*

Estructura del programa a derivar:
$\{\varphi\}$ ¿Inicializaciones? **while** $\{INV\}$ $\{E\}$ ¿B? **loop**   ¿Instrucciones? **end loop;** $\{\psi\}$

Definición de φ, INV, E y ψ:
$\varphi \equiv \quad n \geq 2 \;\wedge\; posit(A(1..n))$
$INV \equiv \quad n \geq 2 \;\wedge\; posit(A(1..n)) \;\wedge\; (1 \leq i \leq n) \wedge$ $\quad q \leftrightarrow \exists k((2 \leq k \leq i) \;\wedge\; ((A(k) \; mod \; k) = 0))$
$E = \quad n - i$
$\psi \equiv \quad q \leftrightarrow \exists k((2 \leq k \leq n) \;\wedge\; ((A(k) \; mod \; k) = 0))$

Definición del predicado utilizado:
$posit(H(1..r)) \equiv \forall k((1 \leq k \leq r) \;\rightarrow\; (H(k) \geq 1))$

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

Abreviaciones recomendadas:
$\lambda \equiv n \geq 2 \;\wedge\; posit(A(1..n))$
$\gamma(\ell) \equiv (A(\ell) \; mod \; \ell) = 0$
$\mu(\ell) \equiv \exists k((2 \leq k \leq \ell) \;\wedge\; ((A(k) \; mod \; k) = 0))$

Tabla 1: Abreviaciones que se recomienda utilizar.

Letras griegas utilizadas:
φ : fi $\quad \psi$: psi $\quad \gamma$: gamma $\quad \mu$: mu $\quad \lambda$: 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 $\neg B$ y B: 0,125
    (b.2) Comprobación del punto (II) de la regla del while: 0,030
    (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.