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: 2019-2020
Grupo 01
Tema 3: Verificación formal de programas
2 puntos

12-03-2020

Respuestas

Índice

Índice de figuras

Índice de tablas

****

1. Verificación formal de un programa iterativo (*2 puntos*)

1.1. Enunciado

Verificar, utilizando el Cálculo de Hoare, la corrección total del programa que se muestra en la figura 1 (página 4), con respecto a la precondición φ, la postcondición ψ, el invariante INV y la expresión cota E. Según la especificación pre-post formalizada mediante las fórmulas φ y ψ, dados un número entero positivo x que es una potencia de 2 y dos vectores no vacíos de enteros positivos $A(1..n)$ y $B(1..n)$ que solo contienen potencias de 2, el programa debería decidir en la variable booleana w si algún elemento de $A(1..n)$ multiplicado por x es igual al correspondiente elemento de $B(1..n)$.

En el programa de la figura 1, div representa la división entera. Ejemplos: $19\ div\ 2 = 9$; $19\ div\ 3 = 6$; $19\ div\ 4 = 4$; $17\ div\ 3 = 5$; $8\ div\ 12 = 0$.

Las potencias enteras de 2 son $2^0, 2^1, 2^2, 2^3, 2^4, 2^5, \ldots$. Es decir, $1, 2, 4, 8, 16, 32, \ldots$. El conjunto formado por todas las potencias enteras de 2 se puede definir formalmente de la siguiente manera: $\{y \mid y \in \mathbb{N} \ \wedge\ \exists \ell(\ell \geq 0\ \wedge\ y = 2^\ell)\}$, donde $\mathbb{N}$ es el conjunto de los números naturales, es decir, $\mathbb{N} = \{0, 1, 2, 3, 4, \ldots\}$. Cualquier potencia entera de 2 es un número positivo (≥ 1).

Durante el proceso de verificación, al analizar el punto (IV) de la regla del While, es necesario utilizar la propiedad matemática $(Prop)$ de la figura 2 (página 4), que dice que, si se tienen tres valores enteros positivos m, p y q que son potencias de 2, entonces el que q sea igual al cociente de dividir m por p, quiere decir que el resultado de multiplicar q por p es igual a m. Por ejemplo, si $m = 32$, $p = 4$ y $q = 8$, puesto que se cumple $8 = 32\ div\ 4$, también se cumple $8 * 4 = 32$. Nótese que no todos los enteros positivos cumplen lo expresado en la segunda parte de la implicación que aparece en la definición de $(Prop)$. Por ejemplo, $3 = 19\ div\ 5$, pero $3 * 5 \neq 19$.

En la tabla 1 (página 5) se recogen las abreviaciones que se recomienda utilizar durante el proceso de verificación. En la tabla 2 (página 5) se recopilan las denominaciones de las letras griegas utilizadas en este documento. Finalmente, en la tabla 3 (página 5) se muestra la puntuación de los distintos pasos o apartados que han de ser considerados en el proceso de verificación.

Todos los elementos numéricos de las figuras 1 y 2 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:

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

Formalmente, $\mathbb{Z} = \mathbb{N}\ \cup\ \{-y \mid y \in \mathbb{N}\ \wedge\ y \geq 1\}$, donde $\cup$ es la unión de conjuntos.

Teniendo en cuenta las abreviaciones de la tabla 1, la propiedad $(Prop)$ de la figura 2 expresa que cada $\gamma(\ell)$ es igual al correspondiente $\sigma(\ell)$, siempre que se garantice que $\gamma(\ell)$, $\sigma(\ell)$ y x son potencias enteras de 2, lo cual es garantizado por λ.

Ejemplo. (Para el programa de la figura 1) *Sean $x = 4$ y los siguientes vectores $A(1..8)$ y $B(1..8)$:*

A(1..8)	*1*	*2*	*4*	*32*	*8*	*4*	*4*	*16*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
B(1..8)	*16*	*1*	*16*	*8*	*32*	*4*	*64*	*32*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*

Para esos valores de x, $A(1..8)$ y $B(1..8)$, el programa de la figura 1 devolvería el valor booleano $True$ en w, porque al menos para un elemento de $A(1..8)$ se cumple que el resultado que se obtiene al multiplicarlo por x es igual al elemento que ocupa la misma posición de $B(1..8)$. En concreto, se cumple para $A(3)$ y $A(5)$. Es decir, $A(3) * x = B(3)$ y $A(5) * x = B(5)$.

1.2. Solución

1.2.1. *(a)* Partición inicial y esquema

La información correspondiente a la partición incial y a los esquemas de los dos programas que surgen tras la partición se muestra en las figuras 3, 4 y 5 (páginas 4, 6 y 7 respectivamente).

En la figura 3 (página 4), se ha colocado el invariante INV entre la inicialización de la variable i y la instrucción $while$. En la figura 4 (página 6) se muestran los dos programas que se obtienen tras la partición en dos del programa original. La tarea es ahora verificar, por separado, la corrección total de esos dos programas (programa 1 y programa 2). Finalmente, en la figura 5 (página 7), se muestran las fórmulas que han de ser calculadas y las implicaciones que han de ser analizadas para decidir si los dos programas son correctos.

1.2.2. *(b)* Verificación de la asignación inicial

- Cálculo de la fórmula φ_1 a partir de la fórmula INV utilizando el axioma de la asignación (AA).

$$\begin{aligned}
\varphi_1 &\equiv def(2) \ \wedge \ INV_i^2 \\
&\equiv True \ \wedge \ \lambda \ \wedge \ (2 \leq 2 \leq n+2) \ \wedge \ (w \leftrightarrow \mu(2-2)) \\
&\equiv True \ \wedge \ \lambda \ \wedge \ (2 \leq 2) \ \wedge \ (2 \leq n+2) \ \wedge \ (w \leftrightarrow \mu(0)) \\
&\equiv True \ \wedge \ \lambda \ \wedge \ True \ \wedge \ (2 \leq n+2) \ \wedge \ (w \leftrightarrow \mu(0)) \\
&\equiv \lambda \ \wedge \ (2 \leq n+2) \ \wedge \ (w \leftrightarrow \mu(0)) \\
&\equiv \lambda \ \wedge \ (0 \leq n) \ \wedge \ (w \leftrightarrow \mu(0))
\end{aligned}$$

  Para simplificar la expresión, se ha descompuesto, por un parte, $(2 \leq 2 \leq n+2)$ en $(2 \leq 2) \ \wedge \ (2 \leq n+2)$. Por otra parte, se ha tenido en cuenta que $True \ \wedge \ \delta \equiv \delta$ para cualquier fórmula δ. Además, se ha transformado $2 \leq n+2$ en $0 \leq n$, restando 2 en ambos lados.

- Comprobación de la implicación: ¿$\varphi \rightarrow \varphi_1$?

$$¿\underbrace{\lambda \ \wedge \ \neg w}_{\varphi} \rightarrow \underbrace{\lambda \ \wedge \ (0 \leq n) \ \wedge \ (w \leftrightarrow \mu(0))}_{\varphi_1}?$$

  En la primera parte (parte izquierda) de la implicación tenemos la información: $\lambda \wedge \neg w$. En la segunda parte (parte derecha) tenemos tres preguntas: ¿λ? ¿$(0 \leq n)$? ¿$(w \leftrightarrow \mu(0))$?

  - ¿λ? Sí, porque en la primera parte, es decir, en φ, aparece λ.
  - ¿$(0 \leq n)$? Sí, porque en la primera parte, es decir, en φ, dentro de λ se tiene que $n \geq 1$. Nótese que ¿$(0 \leq n)$? se puede plantear como ¿$(0 = n \ \vee \ 0 < n)$?. Lo cual es lo mismo que ¿$(0 = n \ \vee \ 1 \leq n)$?. Puesto que se sabe que se cumple $n \geq 1$, nos queda que $0 = n$ es $False$ y $1 \leq n$ es $True$. Por consiguiente, la pregunta se transforma en ¿$(False \ \vee \ True)$?. Y la respuesta es $True$. Así que, sí se cumple.

Programa:

$\{\varphi\}$
i := 2;
while $\{INV\}$ $\{E\}$ i $\neq$ n + 2 **and not** w **loop**
    w := (A(i $-$ 1) = B(i $-$ 1) div x);
    i := i + 1;
end loop;
$\{\psi\}$

Definición de φ, INV, E y ψ:

$$\varphi \equiv \quad n \geq 1 \ \wedge \ pot\_dos\_num(x) \ \wedge \ pot\_dos\_vector(A(1..n)) \wedge pot\_dos\_vector(B(1..n)) \ \wedge \ \neg w$$

$$INV \equiv \quad n \geq 1 \ \wedge \ pot\_dos\_num(x) \ \wedge \ pot\_dos\_vector(A(1..n)) \wedge pot\_dos\_vector(B(1..n)) \ \wedge \ (2 \leq i \leq n+2) \wedge (w \leftrightarrow algun\_cociente(x, A(1..n), B(1..n), i-2))$$

$$E = \quad n + 2 - i$$

$$\psi \equiv \quad w \leftrightarrow \exists k(1 \leq k \leq n \ \wedge \ A(k) * x = B(k))$$

Definición de los predicados utilizados:

$$pot\_dos\_num(z) \equiv \exists \ell(\ell \geq 0 \ \wedge \ z = 2^{\ell})$$

$$pot\_dos\_vector(H(1..r)) \equiv \forall k(1 \leq k \leq r \ \rightarrow \ pot\_dos\_num(H(k)))$$

$$algun\_cociente(y, F(1..r), G(1..r), pos) \equiv \exists k(1 \leq k \leq pos \ \wedge \ F(k) = G(k) \ div \ y)$$

Figura 1: Programa a verificar, definiciones de φ, INV, E y ψ y definiciones de los predicados utilizados.

Propiedad $(Prop)$:

$$\forall m, p, q((pot\_dos\_num(m) \ \wedge \ pot\_dos\_num(p) \ \wedge \ pot\_dos\_num(q)) \rightarrow (q = m \ div \ p \leftrightarrow q * p = m))$$

Figura 2: Propiedad que cumplen tres potencias enteras de 2 cualesquiera.

El invariante se cumple también justo antes de empezar el $while$

$\{\varphi\}$
i := 2;
$\{INV\}$
while $\{INV\}$ $\{E\}$ i $\neq$ n + 2 **and not** w **loop**
    w := (A(i $-$ 1) = B(i $-$ 1) div x);
    i := i + 1;
end loop;
$\{\psi\}$

Figura 3: Programa a verificar, con el invariante entre la inicialización de i y la instrucción $while$.

Abreviaciones recomendadas:
$\lambda \equiv n \geq 1 \ \wedge \ pot\_dos\_num(x) \ \wedge \ pot\_dos\_vector(A(1..n)) \ \wedge \ pot\_dos\_vector(B(1..n))$
$B \equiv i \neq n+2 \ and \ not \ w$
$\gamma(\ell) \equiv A(\ell) = B(\ell) \ div \ x$
$\sigma(\ell) \equiv A(\ell) * x = B(\ell)$
$\mu(\ell) \equiv algun\_cociente(x, A(1..n), B(1..n), \ell)$

Tabla 1: Abreviaciones que se recomienda utilizar.

Letras griegas utilizadas:
φ : fi $\quad \psi$: psi $\quad \gamma$: gamma $\quad \sigma$: sigma $\quad \mu$: mu $\quad \lambda$: lambda

Tabla 2: Denominaciones de las letras griegas utilizadas.

Puntuación:
(a) Partición inicial y esquema: 0,100 *(b)* Verificación de la asignación inicial: 0,150 (Cálculo de fórmulas: 0,050. Comprobación de la implicación: 0,100) *(c)* Punto (I) de la regla del while: 0,005 *(d)* Punto (II) de la regla del while: 0,020 *(e)* Punto (III) de la regla del while: 0,700 (Cálculo de fórmulas: 0,200. Comprobación de la implicación: 0,500) *(f)* Punto (IV) de la regla del while: 0,500 (Casos $i = n+2$: 0,200. Caso $i \neq n+2$: 0,300) *(g)* Punto (V) de la regla del while: 0,075 *(h)* Punto (VI) de la regla del while: 0,200 (Cálculo de fórmulas: 0,050. Comprobación de la implicación: 0,150) *(i)* Demostración formal de la corrección: 0,250 ■ 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 tanto del apartado *(e)* como del apartado *(f)* (puntos (III) y (IV) de la regla del while).

Tabla 3: Puntuación por apartados.

P1 (Programa 1):

$\{\varphi\}$
i := 2;
$\{INV\}$

P2 (Programa 2):

$\{INV\}$
while $\{INV\}$ $\{E\}$ i $\neq$ n + 2 **and not** w **loop**
    w := (A(i − 1) = B(i − 1) div x);
    i := i + 1;
end loop;
$\{\psi\}$

Figura 4: Los dos programas a verificar tras la partición.

- ¿$(w \leftrightarrow \mu(0))$? Si recuperamos el significado de $\mu(0)$, tenemos la siguiente pregunta:

  $$¿(w \leftrightarrow algun\_cociente(x, A(1..n), B(1..n), 0))?$$

  Teniendo en cuenta la definición del predicado que aparece ahí, es decir, la definición del predicado $algun\_cociente(x, A(1..n), B(1..n), 0)$, la pregunta es la siguiente:

  $$¿(w \leftrightarrow \exists k(1 \leq k \leq 0 \ \wedge \ A(k) = B(k) \ div \ x))?$$

  El dominio de definición de la formula existencial es vacío y, consecuentemente, la fórmula existencial es $False$. La pregunta es, por tanto, ¿$(w \leftrightarrow False)$? Dicho de otra forma, ¿$(w = False)$? La respuesta es afirmativa porque en la primera parte de la implicación se nos indica que $\neg w$ es $True$, lo cual signifiva que w es $False$.

1.2.3. *(c)* Punto (I) de la regla del while

¿Se cumple $INV \rightarrow INV$? Se nos dice, en la primera parte de la implicación, que INV es cierto y en la segunda parte se nos pregunta si INV es cierto. La respuesta es que sí.

1.2.4. *(d)* Punto (II) de la regla del while

¿$INV \rightarrow def(B)$?
¿$INV \rightarrow def(i \neq n + 2 \ and \ not \ w)$?
¿$INV \rightarrow True$?
La pregunta es, por tanto, ¿se cumple $True$?. La información que tenemos es que la fórmula INV es cierta. La respuesta es afirmativa, porque $True$ se cumple siempre, es decir, $True$ es $True$ independientemente de la información que se tenga en INV.

1.2.5. *(e)* Punto (III) de la regla del while

- Cálculo de la fórmula φ_2 a partir de la fórmula INV utilizando el axioma de la asignación (AA).

  $$\begin{aligned} \varphi_2 \ &\equiv def(i+1) \ \wedge \ INV_i^{i+1} \\ &\equiv True \ \wedge \ \lambda \ \wedge \ (2 \leq i+1 \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i+1-2)) \\ &\equiv \lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (w \leftrightarrow \mu(i-1)) \end{aligned}$$

  Para simplificar la expresión, se ha tenido en cuenta que $True \ \wedge \ \delta \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(2 \leq i+1 \leq n+2)$ en $(1 \leq i \leq n+1)$, restando 1 en los tres componentes.

Esquema para el programa 1:

$\{\varphi\}$ ¿$\varphi \to \varphi_1$?

$\nearrow$ $\{\varphi_1 \equiv def(2) \ \wedge \ INV_i^2\}$

(AA) i := 2;

$\nwarrow$ $\{INV\}$

Esquema para el programa 2:

(I) ¿$INV \to INV$?

(II) ¿$INV \to def(B)$?

(III)

$\{INV \ \wedge \ B\}$ ¿$(INV \ \wedge \ B) \to \varphi_3$?

$\nearrow$ $\{\varphi_3 \equiv def(\gamma(i-1)) \ \wedge \ (\varphi_2)_w^{\gamma(i-1)}\}$

(AA) w := $\gamma(i-1)$;

$\nearrow$ $\{\varphi_2 \equiv def(i+1) \ \wedge \ INV_i^{i+1}\}$

(AA) i := i + 1;

$\nwarrow$ $\{INV\}$

(IV) ¿$(INV \ \wedge \ \neg B) \to \psi$?

(V) ¿$(INV \ \wedge \ B) \to E > 0$?

(VI)

$\{INV \ \wedge \ B \ \wedge \ E = v\}$ ¿$(INV \ \wedge \ B \ \wedge \ E = v) \to \varphi_5$?

$\nearrow$ $\{\varphi_5 \equiv def(\gamma(i-1)) \ \wedge \ (\varphi_4)_w^{\gamma(i-1)}\}$

(AA) w := $\gamma(i-1)$;

$\nearrow$ $\{\varphi_4 \equiv def(i+1) \ \wedge \ (E < v)_i^{i+1}\}$

(AA) i := i + 1;

$\nwarrow$ $\{E < v\}$

donde $\gamma(i-1)$ representa (A(i − 1) = B(i − 1) div x).

Figura 5: Esquemas para el programa 1 y para el programa 2 de la figura 4 (página 6).

- Cálculo de la fórmula φ_3 a partir de la fórmula φ_2 utilizando el axioma de la asignación (AA).

$$\begin{aligned}
\varphi_3 \quad &\equiv def(\gamma(i-1)) \ \wedge \ (\varphi_2)_w^{\gamma(i-1)} \\
&\equiv def(A(i-1) = B(i-1)\ div\ x) \ \wedge \ \lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (\gamma(i-1) \leftrightarrow \mu(i-1)) \\
&\equiv (1 \leq i-1 \leq n) \ \wedge \ (1 \leq i-1 \leq n) \ \wedge \ (x \neq 0) \ \wedge \\
&\qquad \lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (\gamma(i-1) \leftrightarrow \mu(i-1)) \\
&\equiv (1 \leq i-1 \leq n) \ \wedge \ (x \neq 0) \ \wedge \ \lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (\gamma(i-1) \leftrightarrow \mu(i-1)) \\
&\equiv (2 \leq i \leq n+1) \ \wedge \ (x \neq 0) \ \wedge \ \lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (\gamma(i-1) \leftrightarrow \mu(i-1)) \\
&\equiv (2 \leq i \leq n+1) \ \wedge \ (x \neq 0) \ \wedge \ \lambda \ \wedge \ (\gamma(i-1) \leftrightarrow \mu(i-1)) \\
&\equiv (2 \leq i) \ \wedge \ (i \leq n+1) \ \wedge \ (x \neq 0) \ \wedge \ \lambda \ \wedge \ (\gamma(i-1) \leftrightarrow \mu(i-1))
\end{aligned}$$

  Para simplificar la expresión, se ha tenido en cuenta que $\delta \ \wedge \ \delta \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(1 \leq i-1 \leq n)$ en $(2 \leq i \leq n+1)$, sumando 1 a los tres componentes de la expresión. Para determinar el intervalo de i a partir de $(2 \leq i \leq n+1)$ y $(1 \leq i \leq n+1)$, se ha cogido el mayor de los límites inferiores (2) y el menor de los límites superiores $(n+1)$. Finalmente, se ha separado $(2 \leq i \leq n+1)$ en dos trozos: $(2 \leq i)$ y $(i \leq n+1)$.

- Comprobación de la implicación: $¿INV \ \wedge \ B \rightarrow \varphi_3?$

$$¿\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ (i \neq n+2) \ \wedge \ \neg w \rightarrow (2 \leq i) \ \wedge \ (i \leq n+1) \ \wedge \ (x \neq 0) \ \wedge \ \lambda \ \wedge \ (\gamma(i-1) \leftrightarrow \mu(i-1))?$$

  En la primera parte (parte izquierda o primera línea) de la implicación tenemos la información:

$$\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ (i \neq n+2) \ \wedge \ \neg w$$

  Si descomponemos $(2 \leq i \leq n+2)$, nos queda:

$$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(2 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n+2)}_{\alpha_3} \wedge \underbrace{(w \leftrightarrow \mu(i-2))}_{\alpha_4} \wedge \underbrace{(i \neq n+2)}_{\alpha_5} \wedge \underbrace{\neg w}_{\alpha_6}$$

  En la segunda parte (parte derecha o segunda línea) tenemos cinco preguntas: $¿2 \leq i?$ $¿i \leq n+1?$ $¿x \neq 0?$ $¿\lambda?$ $¿\gamma(i-1) \leftrightarrow \mu(i-1)?$

  - $¿2 \leq i?$ Sí, por α_2.
  - $¿i \leq n+1?$ Sí, por α_3 y α_5.
  - $¿x \neq 0?$ En α_1, es decir, en λ, tenemos $pot\_dos\_num(x)$. Por tanto, x es una potencia de 2 y, consecuentemente, es ≥ 1. Así que es distinto de 0.
  - $¿\lambda?$ Sí, por α_1.
  - $¿\gamma(i-1) \leftrightarrow \mu(i-1)?$ Considerando el significado de $\mu(i-1)$, la pregunta queda de la siguiente forma:

$$¿(\gamma(i-1) \leftrightarrow algun\_cociente(x, A(1..n), B(1..n), i-1))?$$

    Considerando la definición de $algun\_cociente(x, A(1..n), B(1..n), i-1)$, la pregunta es la siguiente:

$$¿(\gamma(i-1) \leftrightarrow \exists k(1 \leq k \leq i-1 \ \wedge \ A(k) = B(k)\ div\ x))?$$

    Las fórmulas existenciales representan una disyunción. En este caso, la pregunta queda de la siguiente manera:

$$¿(\gamma(i-1) \leftrightarrow \gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2) \vee \ \gamma(i-1))?$$

    Esta pregunta no puede ser respondida directamente, porque depende del valor de $\gamma(1) \ \vee \ \gamma(2) \ \vee \ \ldots \vee \ \gamma(i-2)$. En α_4 se nos dice que $w \leftrightarrow \mu(i-2)$, es decir,

$$w \leftrightarrow \exists k(1 \leq k \leq i-2 \ \wedge \ A(k) = B(k)\ div\ x)$$

Utilizando la disyunción, tenemos que $w \leftrightarrow \gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2)$. Por su parte, en α_6 se nos dice que w es $False$. De ello deducimos que $\gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2)$ es $False$. Por tanto, la pregunta queda de la siguiente forma: ¿$(\gamma(i-1) \leftrightarrow False \ \vee \ \gamma(i-1))$? Por equivalencia lógica, sabemos que $False \ \vee \ \delta \equiv \delta$ para cualquier fórmula δ. Entonces, la pregunta es ahora la que sigue: ¿$(\gamma(i-1) \leftrightarrow \gamma(i-1))$? La respuesta es afirmativa.

Hemos comprobado que la implicación $(INV \ \wedge \ B) \rightarrow \varphi_3$ se cumple.

1.2.6. *(f)* Punto (IV) de la regla del while

¿$(INV \ \wedge \ \neg B) \rightarrow \psi$?

¿$\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ ((i = n+2) \ \vee \ w) \rightarrow$
$(w \leftrightarrow \exists k(1 \leq k \leq n \ \wedge \ A(k) * x = B(k)))$?

En la primera parte (parte izquierda o primera línea) de la implicación tenemos la información:

$$\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ ((i = n+2) \ \vee \ w)$$

Si descomponemos $(2 \leq i \leq n+2)$, nos queda:

$$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(2 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n+2)}_{\alpha_3} \wedge \underbrace{(w \leftrightarrow \mu(i-2))}_{\alpha_4} \wedge (\underbrace{(i = n+2)}_{\alpha_5} \vee \underbrace{w}_{\alpha_6})$$

En la segunda parte (parte derecha o segunda línea) tenemos una pregunta:

$$¿w \leftrightarrow \exists k(1 \leq k \leq n \ \wedge \ A(k) * x = B(k))?$$

Si sustituimos la fórmula existencial por la disyunción, tenemos la siguiente pregunta:

$$¿w \leftrightarrow \sigma(1) \ \vee \ \sigma(2) \ \vee \ldots \vee \ \sigma(n)? \tag{1}$$

Por α_4 sabemos que se cumple lo siguiente:

$$w \leftrightarrow \gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2) \tag{2}$$

De α_3 deducimos que $i - 2 \leq n$. Por tanto, el número de gammas (γ) es menor o igual que el número de sigmas (σ). Por otra parte, nos encontramos con que en α_4 tenemos información sobre las gammas pero en ψ necesitamos información sobre las sigmas. En principio, las gammas y las sigmas son expresiones distintas y su valor no tiene porqué coincidir. Pero en α_1, es decir, en λ, se nos dice que tanto x como los valores de los vectores $A(1..n)$ y $B(1..n)$ son potencias enteras de 2. Bajo esas circunstancias, se puede aplicar la propiedad $(Prop)$ de la figura 2 (página 4) y tenemos que cada $\gamma(\ell)$ es igual al correspondiente $\sigma(\ell)$, siendo $1 \leq \ell \leq i-2$:

$$\gamma(1) = \sigma(1), \gamma(2) = \sigma(2), \ldots, \gamma(i-2) = \sigma(i-2) \tag{3}$$

Y, por consiguiente:

$$\gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2) = \sigma(1) \ \vee \ \sigma(2) \ \vee \ldots \vee \ \sigma(i-2) \tag{4}$$

Aunque sabemos que se cumple $w \leftrightarrow \gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2)$, de momento no sabemos el valor de w ni el de $\gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2)$. Para obtener más información, se ha de recurrir a la disyunción $\alpha_5 \vee \alpha_6$. Al ser $\alpha_5 \vee \alpha_6$ una disyunción, puede ser cierta porque se cumplen tanto α_5 como α_6 o porque se cumple solo α_5 o porque se cumple solo α_6. En total hay tres opciones. Esas opciones se muestran en la tabla 4 (página 10).

	α_5	α_6
	$i = n + 2$	$w = True$
1	True	True
2	True	False
3	False	True

Tabla 4: Las tres opciones para que la expresión $((i = n + 2) \ \vee \ w)$ sea cierta.

- Casos 1 y 2: $i = n + 2$

  Si se cumple $i = n+2$, entonces se cumple $i-2 = n$. Consecuentemente, la ecuación (4) de la página 9 se puede reescribir de la siguiente forma:

  $$\gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(n) = \sigma(1) \ \vee \ \sigma(2) \ \vee \ldots \vee \ \sigma(n) \tag{5}$$

  La equivalencia (2) de la página 9 se puede reescribir de la siguiente forma:

  $$w \leftrightarrow \gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(n) \tag{6}$$

  Recordemos que tenemos que responder a la pregunta (1) planteada en la página 9:

  $$¿w \leftrightarrow \sigma(1) \ \vee \ \sigma(2) \ \vee \ldots \vee \ \sigma(n)?$$

  La respuesta es afirmativa por la igualdad (5) (página 10) y la equivalencia (6) (página 10).

  Nótese que en esta deducción no nos hace falta conocer el valor de w. Nos da igual que sea $True$ o $False$. De ahí que se hayan podido analizar juntos los casos 1 y 2 de la tabla 4 de la página 10.

- Caso 3: $i \neq n + 2$ y $w = True$

  Si se cumple $i \neq n + 2$, entonces por α_3 deducimos que $i < n + 2$, es decir, $i - 2 < n$. Consecuentemente, la pregunta (1) de la página 9 se puede reescribir de la siguiente forma:

  $$¿w \leftrightarrow \sigma(1) \ \vee \ \sigma(2) \ \vee \ldots \vee \ \sigma(i-2) \ \vee \ \sigma(i-1) \ \vee \ldots \vee \ \sigma(n)? \tag{7}$$

  Podemos observar que hay más sigmas que gammas. Sabemos que se cumple la igualdad (4) (página 9) pero no tenemos ninguna información sobre la disyunción $\sigma(i-1) \ \vee \ldots \vee \ \sigma(n)$.

  Puesto que estamos considerando el caso en el que $i \neq n + 2$ y $w = True$, por la equivalencia (2) de la página 9 deducimos que $\gamma(1) \ \vee \ \gamma(2) \ \vee \ldots \vee \ \gamma(i-2)$ es $True$. Por la igualdad (4) de la página 9, deducimos que $\sigma(1) \ \vee \ \sigma(2) \ \vee \ldots \vee \ \sigma(i-2)$ es $True$. Por tanto, la pregunta (7) de la página 10 se puede reescribir de la siguiente forma:

  $$¿True \leftrightarrow True \ \vee \ \sigma(i-1) \ \vee \ldots \vee \ \sigma(n)? \tag{8}$$

  Utilizando la equivalencia lógica $True \vee \delta \equiv True$, la pregunta (8) de la página 10 se puede reescribir de la siguiente forma:

  $$¿True \leftrightarrow True?$$

  La respuesta a esa última pregunta es afirmativa.

  Nótese que en la deducción correspondiente al caso $i \neq n + 2$ y $w = True$, sí nos hace falta conocer el valor de w.

1.2.7. *(g)* Punto (V) de la regla del while

¿$(INV \ \wedge \ B) \rightarrow (E > 0)$?

¿$\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ (i \neq n+2) \ \wedge \ \neg w \rightarrow (n+2-i > 0)$?

En la primera parte (parte izquierda) de la implicación tenemos la información:

$$\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ (i \neq n+2) \ \wedge \ \neg w$$

Si descomponemos $(2 \leq i \leq n+2)$, nos queda:

$$\underbrace{\lambda}_{\beta_1} \wedge \underbrace{(2 \leq i)}_{\beta_2} \wedge \underbrace{(i \leq n+2)}_{\beta_3} \wedge \underbrace{(w \leftrightarrow \mu(i-2))}_{\beta_4} \wedge \underbrace{(i \neq n+2)}_{\beta_5} \wedge \underbrace{\neg w}_{\beta_6}$$

En la segunda parte (parte derecha) tenemos una pregunta:

$$¿n+2-i > 0?$$

Por β_3 y β_5 deducimos que $n+2 > i$. Restanto i en ambos lados, obtenemos $n+2-i > i-i$, es decir, $n+2-i > 0$. Y eso es lo que queríamos obtener.

1.2.8. *(h)* Punto (VI) de la regla del while

- Cálculo de la fórmula φ_4 a partir de la fórmula $E < v$ utilizando el axioma de la asignación (AA).

$$\begin{aligned}
\varphi_4 \ &\equiv def(i+1) \ \wedge \ (E < v)_i^{i+1} \\
&\equiv True \ \wedge \ (n+2-(i+1) < v) \\
&\equiv True \ \wedge \ (n+2-i-1) < v) \\
&\equiv (n+1-i < v)
\end{aligned}$$

  Para simplificar la expresión, se ha tenido en cuenta que $True \ \wedge \ \delta \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(n+2-(i+1) < v)$ en $(n+1-i < v)$, aplicando el signo negativo a $(i+1)$ y realizando la resta 2-1.

- Cálculo de la fórmula φ_5 a partir de la fórmula φ_4 utilizando el axioma de la asignación (AA).

$$\begin{aligned}
\varphi_5 \ &\equiv def(\gamma(i-1)) \ \wedge \ (\varphi_4)_w^{\gamma(i-1)} \\
&\equiv def(A(i-1) = B(i-1) \ div \ x) \ \wedge \ (n+1-i < v)_w^{\gamma(i-1)} \\
&\equiv (1 \leq i-1 \leq n) \ \wedge \ (1 \leq i-1 \leq n) \ \wedge \ (x \neq 0) \ \wedge \ (n+1-i < v) \\
&\equiv (1 \leq i-1 \leq n) \ \wedge \ (x \neq 0) \ \wedge \ (n+1-i < v) \\
&\equiv (2 \leq i \leq n+1) \ \wedge \ (x \neq 0) \ \wedge \ (n+1-i < v) \\
&\equiv (2 \leq i) \ \wedge \ (i \leq n+1) \ \wedge \ (x \neq 0) \ \wedge \ (n+1-i < v)
\end{aligned}$$

  Para simplificar la expresión, se ha tenido en cuenta que $\delta \ \wedge \ \delta \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(1 \leq i-1 \leq n)$ en $(2 \leq i \leq n+1)$, sumando 1 a los tres componentes de la expresión. Finalmente, se ha separado $(2 \leq i \leq n+1)$ en dos trozos: $(2 \leq i)$ y $(i \leq n+1)$.

- Comprobación de la implicación: ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_5$?

  ¿$\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ (i \neq n+2) \ \wedge \ \neg w \ \wedge \ (n+2-i = v) \rightarrow$
  $(2 \leq i) \ \wedge \ (i \leq n+1) \ \wedge \ (x \neq 0) \ \wedge \ (n+1-i < v)$?

  En la primera parte (parte izquierda o primera línea) de la implicación tenemos la información:

$$\lambda \ \wedge \ (2 \leq i \leq n+2) \ \wedge \ (w \leftrightarrow \mu(i-2)) \ \wedge \ (i \neq n+2) \ \wedge \ \neg w \ \wedge \ (n+2-i = v)$$

  Si descomponemos $(2 \leq i \leq n+2)$, nos queda:

$$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(2 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n+2)}_{\alpha_3} \wedge \underbrace{(w \leftrightarrow \mu(i-2))}_{\alpha_4} \wedge \underbrace{(i \neq n+2)}_{\alpha_5} \wedge \underbrace{\neg w}_{\alpha_6} \wedge \underbrace{(n+2-i = v)}_{\alpha_7}$$

En la segunda parte (parte derecha o segunda línea) tenemos cuatro preguntas: ¿$2 \leq i$? ¿$i \leq n+1$? ¿$x \neq 0$? ¿$n+1-i < v$?

- ¿$2 \leq i$? Sí, por α_2.
- ¿$i \leq n+1$? Sí, por α_3 y α_5.
- ¿$x \neq 0$? En α_1, es decir, en λ, tenemos $pot\_dos\_num(x)$. Por tanto, x es una potencia de 2 y, consecuentemente, es ≥ 1. Así que es distinto de 0.
- ¿$n+1-i < v$? Por α_7 sabemos que $(n+2-i = v)$. Si restamos 1 a ambos lados de la igualdad, nos queda $(n+2-i-1 = v-1)$, es decir, $(n+1-i = v-1)$. Puesto que para cualquier número entero z se cumple $z-1 < z$, también se cumple $v-1 < v$. Por tanto, deducimos que $(n+2-i-1 < v)$.

Hemos comprobado que la implicación $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_5$ se cumple.

1.2.9. *(i)* Demostración formal de la corrección

En la tabla 5 (página 13) se muestra la demostración formal de la corrección total del programa de la figura 1 (página 4) . En esa demostración, P1 y P2 se refieren a los dos programas obtenidos en la figura 4 (página 6) tras la partición en dos del programa original (figura 1, página 4).

Los primeros tres pasos (pasos 1-3) de la demostración formal corresponden al programa 1 (P1) de la figura 4 (página 6). En la tabla 6 (página 13), se indica el bloque que corresponde a cada uno de esos pasos. Los pasos del 4 al 18 corresponden al programa 2 (P2) de la figura 4 (página 6). En concreto, cada punto de la regla del while está representado mediante uno o varios pasos de esa demostración formal. Los puntos (I), (II), (IV) y (V) de la regla del while requieren un único paso (los pasos 4, 5, 11 y 12, respectivamente). En cambio, tanto el punto (III) como el (VI) requieren varios pasos en la demostración formal. El punto (III) abarca los pasos 6-10, mientras que, por su parte, El punto (VI) abarca los pasos 13-17. Una manera sistemática de obtener esos pasos, es considerar bloques en los esquemas de los puntos (III) y (VI). En las tablas 7 (página 14) y 8 (página 14) se muestran los bloques a considerar. Puesto que se tienen las pruebas de corrección de los seis puntos de la regla de while, en el paso 18 se afirma que el programa 2 de la figura 4 (página 6) es correcto. Finalmente, considerando los pasos 3 y 18, se puede afirmar, en el paso 19, que todo el programa de la figura 1 (página 4) es correcto.

Demostración formal de la corrección total			
P1		1.	$\varphi \rightarrow \varphi_1$
		2.	$\{\varphi_1\}$ i := 2; $\{INV\}$ (AA)
		3.	$\{\varphi\}$ i := 2; $\{INV\}$ (RCN 1, 2)
P2	(I)	4.	$\varphi \rightarrow \varphi_1$
	(II)	5.	$INV \rightarrow def(B)$
	(III)	6.	$INV \ \wedge \ B \rightarrow \varphi_2$
		7.	$\{\varphi_3\}$ w := $\gamma(i-1)$; $\{\varphi_2\}$ (AA)
		8.	$\{INV \ \wedge \ B\}$ w := $\gamma(i-1)$; $\{\varphi_2\}$ (RCN 6, 7)
		9.	$\{\varphi_2\}$ i := i + 1; $\{INV\}$ (AA)
		10.	$\{INV \ \wedge \ B\}$ w := $\gamma(i-1)$; i := i + 1; $\{INV\}$ (RCP 8, 9)
	(IV)	11.	$INV \ \wedge \ \neg B \rightarrow \psi$
	(V)	12.	$INV \ \wedge \ B \rightarrow E > 0$
	(VI)	13.	$INV \ \wedge \ B \ \wedge \ E = v \rightarrow \varphi_5$
		14.	$\{\varphi_5\}$ w := $\gamma(i-1)$; $\{\varphi_4\}$ (AA)
		15.	$\{INV \ \wedge \ B \ \wedge \ E = v\}$ w := $\gamma(i-1)$; $\{\varphi_4\}$ (RCN 13, 14)
		16.	$\{\varphi_4\}$ i := i + 1; $\{E < v\}$ (AA)
		17.	$\{INV \ \wedge \ B \ \wedge \ E = v\}$ w := $\gamma(i-1)$; i := i + 1; $\{E < v\}$ (RCP 15, 16)
		18.	$\{INV\}$ while $\{INV\}$ $\{E\}$ B loop w := $\gamma(i-1)$; i := i + 1; end loop; $\{\psi\}$ (RWH 4, 5, 10, 11, 12, 17)

		19.	$\{\varphi\}$
			i := 2;
			while $\{INV\}$ $\{E\}$ B loop
			w := $\gamma(i-1)$;
			i := i + 1;
			end loop;
			$\{\psi\}$ (RCP 3, 18)

Tabla 5: Demostración formal de la corrección total del programa de la figura 1.

$P1$

3.

1. $\{\varphi\}$
   $\{\varphi_1 \equiv def(2) \ \wedge \ INV_i^2\}$

2. $\{\varphi_1 \equiv def(2) \ \wedge \ INV_i^2\}$
   i := 2;
   $\{INV\}$

Tabla 6: Bloques a considerar en la demostración formal con respecto al programa 1 (ver figura 5 y tabla 5).

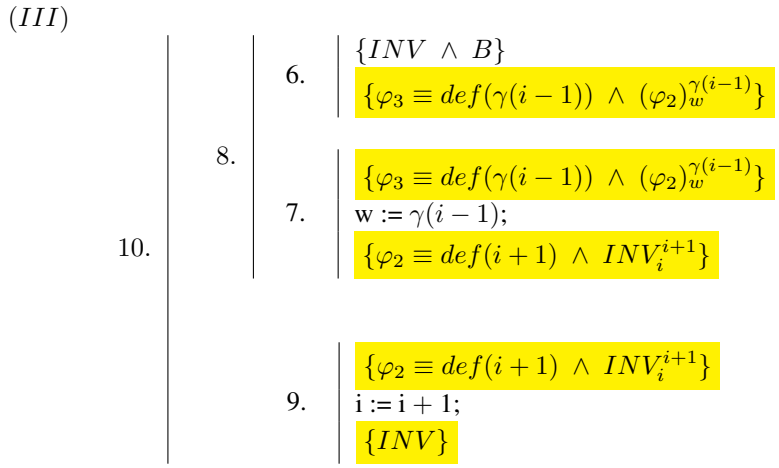

Tabla 7: Bloques a considerar en la demostración formal con respecto al punto (III) (ver figura 5 y tabla 5).

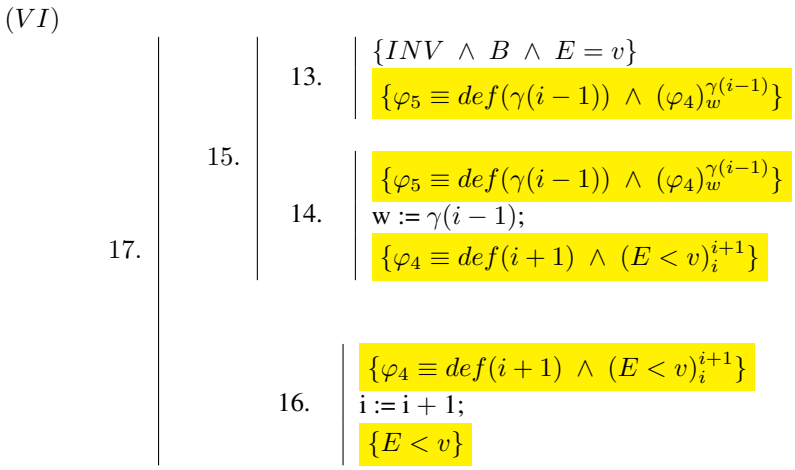

Tabla 8: Bloques a considerar en la demostración formal con respecto al punto (VI) (ver figura 5 y tabla 5).

Letras griegas adicionales utilizadas en el documento de las respuestas:
α : alfa β : beta

Tabla 9: Denominaciones de las letras griegas adicionales utilizadas en el documento de las respuestas.