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 4: Derivación formal de programas
1,5 puntos

15-04-2021

Solución

Índice

Lista de figuras

Lista de tablas

****

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

1.1 Enunciado

Derivar, utilizando la regla del while y el axioma de la asignación del Cálculo de Hoare, un programa que recibe como datos de entrada los siguientes elementos:

- un vector no vacío de enteros no negativos $A(1..n)$,
- un entero x, que es mayor o igual que 2,
- un entero no negativo s.

Y devuelve en la variable booleana q, la decisión sobre si para cada elemento de $A(1..n)$, el resto de dividirlo por x es s.

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 negativa, el programa ha de parar sin analizar las posiciones restantes.

En la figura 1 de la página 4 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 de la página 4, se recogen las abreviaciones que se recomienda utilizar durante el proceso de verificación. En la tabla 2 de la página 4, se recopilan las denominaciones de las letras griegas utilizadas en este enunciado. Finalmente, en la tabla 3 de la página 5, 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 de la página 4 y de la tabla 1 de la página 4) 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)	*28*	*33*	*8*	*18*	*3*	*43*	*48*	*33*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*

Para esos valores de $A(1..8)$ y los valores $x = 5$ y $s = 3$, el programa a derivar —cuya estructura se muestra en la figura 1— devolvería el valor booleano $True$ en q, porque para cada elemento de $A(1..8)$, el resto de dividirlo por x es s.

En cambio, si el vector $A(1..8)$ fuese el que se muestra a continuación, la respuesta debería ser $False$ en q porque no ocurre que para cada elemento de $A(1..n)$, el resto de dividirlo por x sea s. En concreto, para los elementos de las posiciones 2, 6 y 7 de $A(1..8)$, el resto de dividirlos por x no es s puesto que $10\ mod\ 5 = 0 \neq 3$, $29\ mod\ 5 = 4 \neq 3$ y $31\ mod\ 5 = 1 \neq 3$.

A(1..8)	*28*	**10**	*8*	*18*	*3*	**29**	**31**	*33*
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*

1.2 Solución

En la tabla 5 se recogen las letras griegas adicionales utilizadas en este apartado correspondiente a la solución.

1.2.1 *(a)* Cálculo de las inicializaciones previas al while

El apartado de inicializaciones previas al $while$ va asociado al punto (I) de la regla del $while$.

- ¿$\varphi \rightarrow INV$?

$$¿\underbrace{\lambda}_{\varphi} \rightarrow \underbrace{\lambda\ \wedge\ (1 \leq i \leq n+1)\ \wedge\ (q \leftrightarrow \mu(i-1))}_{INV}? \tag{1}$$

  En la primera parte de la implicación (es decir, en φ) tenemos la información y en la segunda parte (INV) tenemos cuatro preguntas: ¿λ? ¿$1 \leq i$? ¿$i \leq n+1$? ¿$q \leftrightarrow \mu(i-1)$?

  Si la implicación se cumple, no hace falta ninguna inicialización. En cambio, si la implicación no se cumple, será necesario introducir las inicializaciones que haga falta con el objetivo de hacer que la implicación se cumpla.

  En la figura 2 se muestra la situación de partida para el cálculo de las inicializaciones.

  - ¿λ? Sí, porque en φ tenemos λ.
  - ¿$1 \leq i$? De la información que se tiene en φ no se puede deducir esto. No sabemos si se cumple o no se cumple $1 \leq i$.
  - ¿$i \leq n+1$? De la información que se tiene en φ no se puede deducir esto. No sabemos si se cumple o no se cumple $i \leq n+1$.
  - ¿$q \leftrightarrow \mu(i-1)$? De la información que se tiene en φ no se puede deducir esto. No sabemos si se cumple o no se cumple $q \leftrightarrow \mu(i-1)$.

  Por tanto, la implicación $\varphi \rightarrow INV$ no se cumple. En φ no hay información ni sobre i ni sobre q.

- **Objetivo:** que se cumpla lo expresado en INV.

  Para hacer que se cumpla lo expresado en INV disponemos de la asignación. Tenemos que conseguir que se cumpla lo expresado en INV haciendo uso de la asignación. Puesto que tenemos dos variables, i y q, cuyo valor se desconoce, necesitaremos inicializar esas dos variables. En el caso de q, queremos que se cumpla $q \leftrightarrow \mu(i-1)$, pero al no conocer nada sobre el valor de i, resulta imposible averiguar cuál es el valor que habría que darle a q: $True$ o $False$. Por ello, nos centramos en i. De φ sabemos que $n \geq 1$. Con esa información en mano, podemos asegurar que si asignamos a i el valor 1 o el valor $n+1$, entonces i cumplirá tanto $1 \leq i$ como $i \leq n+1$. Así que en este momento tenemos dos opciones. Esas dos opciones son los extremos del intervalo que hay en INV para la variable i. La primera opción

Estructura del programa a derivar:

$\{\varphi\}$
¿Inicializaciones?
while $\{INV\}$ $\{E\}$ ¿B? **loop**
    ¿Instrucciones?
end loop;
$\{\psi\}$

Definición de φ, INV, E y ψ:

$$\varphi \equiv \quad n \geq 1 \ \wedge \ mayor\_igual(A(1..n), 0) \ \wedge \ x \geq 2 \ \wedge \ s \geq 0$$

$$INV \equiv \quad n \geq 1 \ \wedge \ mayor\_igual(A(1..n), 0) \ \wedge \ x \geq 2 \ \wedge \ s \geq 0 \ \wedge \ (1 \leq i \leq n+1) \wedge (q \leftrightarrow \forall k(1 \leq k \leq i-1 \ \rightarrow \ A(k) \ mod \ x = s))$$

$$E = \quad n + 1 - i$$

$$\psi \equiv \quad q \leftrightarrow \forall k(1 \leq k \leq n \ \rightarrow \ A(k) \ mod \ x = s)$$

Definición del predicado utilizado:

$$mayor\_igual(H(1..r), y) \equiv \forall k(1 \leq k \leq r \ \rightarrow \ H(k) \geq y)$$

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

Abreviaciones recomendadas:

$$\lambda \equiv n \geq 1 \ \wedge \ mayor\_igual(A(1..n), 0) \ \wedge \ x \geq 2 \ \wedge \ s \geq 0$$

$$\gamma(\ell) \equiv A(\ell) \ mod \ x = s$$

$$\mu(\ell) \equiv \forall k(1 \leq k \leq \ell \ \rightarrow \ A(k) \ mod \ x = s)$$

$$\mu(\ell) \equiv \forall k(1 \leq k \leq \ell \ \rightarrow \ \gamma(k))$$

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.

Esquema de partida para el cálculo de las inicializaciones:

$\{\varphi\}$ ¿$\varphi \rightarrow INV$?

$\{INV\}$

Figura 2: Esquema de partida para el cálculo de las inicializaciones.

supone empezar a recorrer el vector $A(1..n)$ desde la izquierda, mientras que la segunda opción supone empezar a recorrer el vector $A(1..n)$ desde la derecha. La elección que hagamos ha de estar de acuerdo con lo que dice INV y con lo que dice la expresión cota E.

Si la expresión cota tiene la forma $n + z - i$, siendo z un número entero, ello significa que hay que recorrer los vectores de izquierda a derecha. Si la expresión cota tiene la forma $i - z$, siendo z un número entero, ello significa que hay que recorrer los vectores de derecha a izquierda.

En nuestro caso, E tiene la forma $n + z - i$ con $z = 1$. Por tanto, hay que recorrer los vectores de izquierda a derecha.

Consecuentemente, se ha de inicializar i con 1. Una vez colocada la asignación $i := 1$; en el programa que estamos construyendo, hay que calcular la fórmula correspondiente, φ_1, haciendo uso del axioma de la asignación (AA).

En la figura 3 de la página 8, se muestra la situación que se tiene tras inicializar i.

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

$$\begin{aligned}\varphi_1 &\equiv def(1) \ \wedge \ INV_i^1 \\ &\equiv True \ \wedge \ \lambda \ \wedge \ (1 \leq 1 \leq n+1) \ \wedge \ (q \leftrightarrow \mu(1-1)) \\ &\equiv True \ \wedge \ \lambda \ \wedge \ (1 \leq 1) \ \wedge \ (1 \leq n+1) \ \wedge \ (q \leftrightarrow \mu(0)) \\ &\equiv True \ \wedge \ \lambda \ \wedge \ True \ \wedge \ (0 \leq n) \ \wedge \ (q \leftrightarrow \mu(0)) \\ &\equiv \lambda \ \wedge \ (0 \leq n) \ \wedge \ (q \leftrightarrow \mu(0))\end{aligned}$$

  Para simplificar φ_1, primero se ha descompuesto $(1 \leq 1 \leq n+1)$ en $(1 \leq 1) \ \wedge \ (1 \leq n+1)$. Después, por una parte, se ha tenido en cuenta que $z \leq z$ es cierto para cualquier número entero z y se ha puesto $True$ en vez de $(1 \leq 1)$. Por otra parte, se ha restado 1 a cada componente de $1 \leq n+1$ y se ha obtenido $0 \leq n$. Finalmente, se ha tenido en cuenta que $(True \wedge \delta) \equiv \delta$ para cualquier fórmula δ.

- ¿$\varphi \rightarrow \varphi_1$?

$$¿\underbrace{\lambda}_{\varphi} \rightarrow \underbrace{\lambda \ \wedge \ (0 \leq n) \ \wedge \ (q \leftrightarrow \mu(0))}_{\varphi_1}? \tag{2}$$

  En la primera parte de la implicación (φ) tenemos la información y en la segunda parte (φ_1) tenemos tres preguntas: ¿λ? ¿$0 \leq n$? ¿$q \leftrightarrow \mu(0)$?

  Si la implicación se cumple, no hace falta ninguna otra inicialización. En cambio, si la implicación no se cumple, será necesario introducir las inicializaciones que haga falta con el objetivo de hacer que la implicación se cumpla.

  - ¿λ? Sí, porque en φ tenemos λ.
  - ¿$0 \leq n$? En φ se nos dice que $n \geq 1$. Si n es mayor o igual que 1, entonces necesariamente ha de ser también mayor o igual que 0. No es posible tener un número que sea mayor o igual que 1 y que no sea mayor o igual que 0. Por tanto, se cumple $0 \leq n$.
  - ¿$q \leftrightarrow \mu(0)$? De la información que se tiene en φ no se puede deducir esto. No sabemos si se cumple o no se cumple $q \leftrightarrow \mu(0)$.

  En resumen, la implicación $\varphi \rightarrow \varphi_1$ no se cumple. En φ no hay información sobre q.

- **Objetivo:** que se cumpla lo expresado en φ_1.

  Hemos de utilizar la asignación para hacer que sea cierto lo expresado en φ_1. En concreto, queremos que la variable q cumpla $q \leftrightarrow \mu(0)$. El significado de $\mu(0)$ es el siguiente:

$$\forall k(1 \leq k \leq 0 \ \rightarrow \ A(k) \ mod \ x = s) \tag{3}$$

El dominio $1 \leq k \leq 0$ de esa fórmula universal es vacío. Consiguientemente, el valor de la fórmula es $True$. Es decir, $\mu(0)$ es $True$. Por tanto, el objetivo es que se cumpla $q \leftrightarrow True$. Para conseguir ese objetivo, hay que asignar $True$ a q.

Una vez que colocamos la asignación $q := True$; en el programa que estamos construyendo, hay que calcular la fórmula correspondiente, φ_2, haciendo uso del axioma de la asignación (AA).

En la figura 4, se muestra la situación que se tiene tras inicializar q.

- Cálculo de la fórmula φ_2 a partir de la fórmula φ_1 utilizando el axioma de la asignación (AA).

$$\begin{aligned}\varphi_2 &\equiv def(True) \ \wedge \ (\varphi_1)_q^{True} \\ &\equiv True \ \wedge \ \lambda \ \wedge \ (0 \leq n) \ \wedge \ (True \leftrightarrow \mu(0)) \\ &\equiv \lambda \ \wedge \ (0 \leq n) \ \wedge \ (True \leftrightarrow \mu(0))\end{aligned}$$

  Para simplificar la fórmula φ_2, se ha tenido en cuenta que $(True \wedge \delta) \equiv \delta$ para cualquier fórmula δ.

- ¿$\varphi \rightarrow \varphi_2$?

$$¿\underbrace{\lambda}_{\varphi} \rightarrow \underbrace{\lambda \ \wedge \ (0 \leq n) \ \wedge \ (True \leftrightarrow \mu(0))}_{\varphi_2}? \tag{4}$$

  En la primera parte de la implicación (φ) tenemos la información y en la segunda parte (φ_2) tenemos tres preguntas: ¿λ? ¿$0 \leq n$? ¿$True \leftrightarrow \mu(0)$?

  Si la implicación se cumple, habremos terminado con las inicializaciones. Pero si la implicación no se cumple, harán falta más inicializaciones que hagan que se cumpla la implicación.

  - ¿λ? Sí, porque en φ tenemos λ.
  - ¿$0 \leq n$? Puesto que en φ, es decir, en λ, se nos dice que $n \geq 1$, podemos afirmar que también se cumple $0 \leq n$.
  - ¿$True \leftrightarrow \mu(0)$? En la fórmula (3) de la página 6 se constata que $\mu(0)$ representa una fórmula universal de dominio vacío. Consecuentemente, su valor es $True$. Por tanto, la pregunta es la siguiente: ¿$True \leftrightarrow True$? La respuesta es afirmativa porque $(\delta \leftrightarrow \delta) \equiv True$ para cualquier fórmula δ.

  Hemos deducido que la implicación $\varphi \rightarrow \varphi_2$ se cumple y hemos terminado con las inicializaciones.

1.2.2 *(b)* Cálculo de la condición del while (B)

La condición del $while$ se ha de construir a partir de la información extraída del invariante. Una vez formulada la condición del $while$, se procederá a verificar que, efectivamente, es correcta. Para ello, se comprobarán las implicaciones correspondientes a los puntos (II), (IV) y (V) de la regla del $while$.

1.2.2.1 *(b.1)* **Formulación de $\neg B$ y B** Para calcular la condición B del $while$, se formulará primero $\neg B$. La expresión $\neg B$ indica la condición que ha de darse para que el $while$ termine o se detenga. Por tanto, se obtiene respondiendo a la pregunta ¿Cuándo se parará el $while$? o a la pregunta equivalente ¿Cuándo se saldrá del $while$?

La respuesta genérica es: Cuando se sepa la respuesta definitiva, es decir, cuando se sepa si hay que devolver el valor $True$ en q o hay que devolver el valor $False$ en q.

Se sabrá que hay que devolver el valor $False$ en q cuando se encuentre una posición de $A(1..n)$ para la cual se cumpla que el resto de dividir ese elemento de $A(1..n)$ por x no sea s. Es decir, se sabrá que hay que

Esquema tras inicializar la variable i:

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

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

(AA) i := 1;

$\{INV\}$

Figura 3: Inicialización de i.

Esquema tras inicializar la variable q:

$\{\varphi\}$ ¿$\varphi \rightarrow \varphi_2$?

$\{\varphi_2 \equiv def(True) \ \wedge \ (\varphi_1)_q^{True}\}$

(AA) q := True;

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

(AA) i := 1;

$\{INV\}$

Figura 4: Inicialización de q.

devolver $False$ en q si se cumple que en el tramo recorrido desde la posición 1 hasta lal posición $i-1$, se tenga que para alguna posición de ese tramo, el resto de dividir ese elemento de $A(1..n)$ por x no sea s. Por tanto, cuando la fórmula $\forall k(1 \leq k \leq i-1 \ \rightarrow \ A(k)\ mod\ x = s)$ —que aparece en el invariante— sea $False$. Además, el invariante nos indica que en cualquier vuelta del $while$ el valor de esa fórmula coincide con el valor de la variable q. La fórmula $\forall k(1 \leq k \leq i-1 \ \rightarrow \ A(k)\ mod\ x = s)$ no se puede poner directamente en la condición B porque esa fórmula no está escrita en el lenguaje de programación que se está utilizando. Pero q coincide en valor con esa fórmula y q sí se puede utilizar. Por consiguiente, podemos decir que se sabrá que la respuesta definitiva del programa ha de ser $False$ en cuanto q tome el valor $False$.

Por otro lado, si q se mantiene con el valor $True$ pero se termina de recorrer el vector, en ese caso se sabrá que la respuesta definitiva del programa ha de ser $True$.

En síntesis, se tiene la respuesta definitiva cuando q pasa a tener el valor $False$ o cuando se ha terminado de recorrer los vectores, es decir, cuando i llega a tomar el último valor permitido según el invariante:

$$\begin{aligned}\neg B \ &\equiv \underbrace{(\neg\forall k(1 \leq k \leq i-1 \ \rightarrow \ A(k)\ mod\ x = s))}\vee \underbrace{(i = n+1)}\\ &\equiv \underbrace{(\neg q)}\vee\underbrace{(i = n+1)}\\ &\equiv \underbrace{(q = False)}\vee\underbrace{(i = n+1)}\end{aligned}$$

Una vez que se tiene $\neg B$, su negación nos dará B:

$$\begin{aligned}B \ &\equiv \neg(\neg B)\\ &\equiv \neg((q = False) \vee (i = n))\\ &\equiv (\neg(q = False)) \wedge (\neg(i = n+1))\\ &\equiv (q = True) \wedge (i \neq n+1)\end{aligned}$$

1.2.2.2 *(b.2)* **Comprobación del punto (II) de la regla del while** Para que la condición B sea adecuada se ha de cumplir el punto (II) de la regla del $while$.

$$\begin{aligned}&¿INV \rightarrow def(B)?\\ &¿INV \rightarrow def((q = True) \wedge (i \neq n+1))?\\ &¿INV \rightarrow True?\end{aligned}$$

Se nos dice que la fórmula INV es cierta y se nos pregunta si se cumple $True$. La respuesta es afirmativa, porque $True$ se cumple siempre, es decir, $True$ es $True$ independientemente de la información que se tenga en INV. Dicho de otra manera, $(\delta \rightarrow True) \equiv True$ para cualquier fórmula lógica δ. También es posible indicar lo siguiente: $(\delta \rightarrow True) \equiv ((\neg\delta) \vee True) \equiv True$ puesto que $(\pi \vee True) \equiv True$ para cualquier fórmula π.

1.2.2.3 *(b.3)* **Comprobación del punto (IV) de la regla del while**

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

$$¿\underbrace{\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1))}_{INV} \ \wedge \ \underbrace{((q = False) \ \vee \ (i = n+1))}_{B} \rightarrow \underbrace{(q \leftrightarrow \mu(n))}_{\psi}?$$

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

$$\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1)) \ \wedge \ ((q = False) \ \vee \ (i = n+1))$$

Si descomponemos $(1 \leq i \leq n+1)$, nos queda:

$$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(1 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n+1)}_{\alpha_3} \wedge \underbrace{(q \leftrightarrow \mu(i-1))}_{\alpha_4} \wedge (\underbrace{(q = False)}_{\alpha_5} \vee \underbrace{(i = n+1)}_{\alpha_6})$$

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

$$¿\underbrace{q \leftrightarrow \mu(n)}_{\psi}?$$

Por α_3, sabemos que $i \leq n+1$, es decir, $i-1 \leq n$. Por tanto, puede que α_4 y ψ no sean la misma fórmula.

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 14).

- Casos 1 y 3: $i = n + 1$

  Si se cumple $i = n + 1$, entonces $i - 1 = n$ y, consecuentemente, α_4 y ψ son la misma fórmula. Como α_4 se cumple, ψ también se cumple.

  Nótese que en esta deducción no nos hace falta conocer el valor de q. Nos da igual que sea $True$ o $False$. De ahí que se hayan podido analizar juntos los casos 1 y 3 de la tabla 4 de la página 14.

- Caso 2: $q = False$ e $i \neq n + 1$

  Puesto que se cumplen α_3 e $i \neq n+1$, deducimos que $i \leq n$ y, por consiguiente, se cumple $i-1 \leq n-1$. Por tanto, α_4 y ψ no son la misma fórmula.

  Recordemos que la pregunta es la siguiente:

  $$¿q \leftrightarrow \mu(n)? \tag{5}$$

  Si recuperamos el significado de $\mu(n)$, la pregunta es la siguiente:

  $$¿q \leftrightarrow \forall k(1 \leq k \leq n \ \rightarrow \ A(k) \ mod \ x = s)? \tag{6}$$

  Si utilizamos abreviaturas, podemos reescribir la pregunta de la siguiente manera:

  $$¿q \leftrightarrow \forall k(1 \leq k \leq n \ \rightarrow \ \gamma(k))?$$

  Si sustituimos la fórmula universal por la conjunción, tenemos la siguiente pregunta:

  $$¿q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(n))? \tag{7}$$

  Por α_4 sabemos que se cumple lo siguiente:

  $$q \leftrightarrow \forall k(1 \leq k \leq i-1 \ \rightarrow \ A(k) \ mod \ x = s)$$

  Si utilizamos abreviaturas, podemos reescribir esa equivalencia de la siguiente manera:

  $$q \leftrightarrow \forall k(1 \leq k \leq i-1 \ \rightarrow \ \gamma(k))$$

  Si sustituimos la fórmula universal por la conjunción, tenemos que se cumple lo siguiente:

  $$q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1)) \tag{8}$$

  Puesto que por cumplirse α_3 e $i \neq n + 1$ hemos deducido que $i \leq n$ e $i - 1 \leq n - 1$, el número de gammas de (8) es menor que el número de gammas de (7).

  Consecuentemente, la pregunta (7) de la página 10 se puede reescribir de la siguiente forma:

  $$¿q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1) \ \wedge \ \gamma(i) \ \wedge \cdots \wedge \ \gamma(n))? \tag{9}$$

Podemos observar que hay más gammas en (9) que en la equivalencia (8) (página 10).

Puesto que estamos considerando el caso en el que $q = False$ e $i \neq n + 1$, por la equivalencia (8) de la página 10 deducimos que $\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1)$ es $False$. Por tanto, la pregunta (9) de la página 10 se puede reescribir de la siguiente forma:

$$¿False \leftrightarrow (False \ \wedge \ \gamma(i) \ \wedge \cdots \wedge \ \gamma(n))? \tag{10}$$

No tenemos ninguna información sobre la conjunción $\gamma(i) \wedge \cdots \wedge \gamma(n)$, pero utilizando la equivalencia lógica $(False \wedge \delta) \equiv False$, la pregunta (10) de la página 11 se puede reescribir de la siguiente forma:

$$¿False \leftrightarrow False?$$

La respuesta a esa última pregunta es afirmativa porque $(\delta \leftrightarrow \delta) \equiv True$ para cualquier fórmula δ.

Nótese que en la deducción correspondiente al caso $q = False$ e $i \neq n + 1$, sí nos hace falta conocer el valor de q.

1.2.2.4 *(b.4)* Comprobación del punto (V) de la regla del while

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

$$¿\underbrace{\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1))}_{INV} \wedge \underbrace{(q = True) \ \wedge \ (i \neq n+1)}_{B} \rightarrow \underbrace{(n+1-i > 0)}_{E>0}?$$

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

$$\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1)) \ \wedge \ (q = True) \ \wedge \ (i \neq n+1)$$

Si descomponemos $(1 \leq i \leq n + 1)$, nos queda:

$$\underbrace{\lambda}_{\beta_1} \wedge \underbrace{(1 \leq i)}_{\beta_2} \wedge \underbrace{(i \leq n+1)}_{\beta_3} \wedge \underbrace{(q \leftrightarrow \mu(i-1))}_{\beta_4} \wedge \underbrace{(q = True)}_{\beta_5} \wedge \underbrace{(i \neq n+1)}_{\beta_6}$$

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

$$¿n + 1 - i > 0?$$

Por β_3 y β_6 deducimos que $n + 1 > i$. Restando i en ambos lados, obtenemos $n + 1 - i > i - i$, es decir, $n + 1 - i > 0$. Y eso es lo que queríamos obtener. Por tanto, la implicación se cumple.

1.2.3 *(c)* Cálculo de las instrucciones que van dentro del while

La idea es la misma que en el cálculo de las inicializaciones: a base de analizar las implicaciones correspondientes, se sabrá si hace falta añadir más asignaciones y, además, las fórmulas utilizadas nos indicarán qué asignaciones añadir. Es necesario llevar el punto (III) y el punto (VI) en paralelo, porque en ambos casos las instrucciones (asignaciones) han de ser las mismas, aunque las fórmulas a considerar sean distintas.

1.2.3.1 *(c.1)* y *(c.2)* Desarrollo relacionado con el punto (III) y con el punto (VI) de la regla del while

1. $¿(INV \ \wedge \ B) \rightarrow INV?$ $¿(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow E < v?$

   En la figura 5, se muestra el punto de partida para el cálculo de las instrucciones que han de ir dentro del $while$. Si se cumplen la implicación $(INV \ \wedge \ B) \rightarrow INV$ y la implicación $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow E < v$, entonces no hará falta añadir ninguna asignación. Si alguna de ellas no se cumple, habrá que añadir al menos una asignación con el objetivo de hacer que la implicación se cumpla.

- ¿$(INV \ \wedge \ B) \rightarrow INV$?
  Se nos dice que las fórmulas INV y B son ciertas y se nos pregunta si la fórmula INV es cierta. La respuesta es afirmativa, trivialmente. Otra manera de justificar que esta implicación se cumple, consiste en tener en cuenta que $((\delta_1 \wedge \delta_2) \rightarrow \delta_1) \equiv True$ para dos fórmulas cualesquiera δ_1 y δ_2. Nótese que $((\delta_1 \wedge \delta_2) \rightarrow \delta_1)$ es equivalente $(\neg(\delta_1 \wedge \delta_2) \vee \delta_1)$, que a su vez es equivalente a $((\neg\delta_1) \vee (\neg\delta_2) \vee \delta_1)$. Siempre que en una disyunción de fórmulas aparezcan una fórmula y su negación, la disyunción será cierta porque $((\neg\pi) \vee \pi) \equiv True$ para cualquier fórmula lógica π. Consecuentemente, $((\neg\delta_1) \vee (\neg\delta_2) \vee \delta_1) \equiv True \vee (\neg\delta_2) \equiv True$.
- ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow E < v$? Se nos dice que las fórmulas INV, B y $E = v$ son ciertas y se nos pregunta si la fórmula $E < v$ es cierta. La respuesta es negativa, trivialmente. Si E es igual a v, entonces E no puede ser, al mismo tiempo, menor que v.

El que se cumpla la implicación $(INV \ \wedge \ B) \rightarrow INV$ conlleva que en lo que respecta al punto (III) no haya necesidad de añadir ninguna asignación. Pero el que no se cumpla la implicación $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow E < v$ conlleva que en lo que respecta al punto (VI) sí haya que añadir alguna asignación, siempre con el objetivo de que se cumpla $E < v$.

2. **Objetivo:** que se cumpla $E < v$:
   El objetivo es que se cumpla $n+1-i < v$ sabiendo que ahora se cumple $E = v$, es decir, $n+1-i = v$.

   $n+1-i$ representa la distancia entre el punto al que hay que llegar $(n+1)$ y el punto en el que estamos (i). Puesto que previamente hemos visto que el vector se va a recorrer de izquierda a derecha, es decir, hemos visto que el valor de i ha de ir creciendo, si incrementamos el valor de i en uno, conseguiremos que la distancia entre el punto al que hay que llegar $(n+1)$ y el punto en el que estamos (i) decrezca.

   Por tanto, deducimos que hay que añadir la asignación $i := i + 1;$.

   Esa asignación hay que añadirla tanto en el punto (III) como en el (VI), porque las instrucciones han de ser las mismas en ambos casos. Una vez añadida la asignación, se ha de calcular la fórmula correspondiente, utilizando para ello el axioma de la asignación (AA).

3. Cálculo de φ_3 y φ_4:

   - Cálculo de la fórmula φ_3 a partir de la fórmula INV utilizando el axioma de la asignación (AA).
     $$\begin{aligned} \varphi_3 \ &\equiv def(i+1) \ \wedge \ INV_i^{i+1} \\ &\equiv True \ \wedge \ \lambda \ \wedge \ (1 \leq i+1 \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i+1-1)) \\ &\equiv \lambda \ \wedge \ (0 \leq i \leq n) \ \wedge \ (q \leftrightarrow \mu(i)) \end{aligned}$$
     Para simplificar φ_3, se ha tenido en cuenta que $(True \wedge \delta) \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(1 \leq i+1 \leq n+1)$ en $(0 \leq i \leq n)$, restando 1 en los tres componentes. También se ha realizado la resta en $i+1-1$ y en vez de $\mu(i+1-1)$ ha quedado $\mu(i)$.
   - Cálculo de la fórmula φ_4 a partir de la fórmula $E < v$ utilizando el axioma de la asignación (AA).
     $$\begin{aligned} \varphi_4 \ &\equiv def(i+1) \ \wedge \ (E < v)_i^{i+1} \\ &\equiv True \ \wedge \ (n+1-(i+1) < v) \\ &\equiv True \ \wedge \ (n+1-i-1 < v) \\ &\equiv (n-i < v) \end{aligned}$$
     Para simplificar φ_4, se ha tenido en cuenta que $(True \wedge \delta) \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(n+1-(i+1) < v)$ en $(n+1-i-1 < v)$, aplicando el signo negativo a $(i+1)$ y, además, se ha realizado la resta, transformando $(n+1-i-1 < v)$ en $(n-i < v)$.

En la figura 6 de la página 14 se muestran los esquemas para los puntos (III) y (VI) tras la actualización de i. Una vez que se han calculado φ_3 y φ_4, se han de comprobar las implicaciones $(INV \ \wedge \ B) \rightarrow \varphi_3$ e $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_4$.

4. ¿$(INV \ \wedge \ B) \rightarrow \varphi_3$? ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_4$?

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

$$¿\underbrace{\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1))}_{INV} \ \wedge \ \underbrace{(q = True) \ \wedge \ (i \neq n+1)}_{B} \rightarrow$$

$$\underbrace{\lambda \ \wedge \ (0 \leq i \leq n) \ \wedge \ (q \leftrightarrow \mu(i))}_{\varphi_3}?$$

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

$$\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1)) \ \wedge \ (q = True) \ \wedge \ (i \neq n+1)$$

Si descomponemos $(1 \leq i \leq n+1)$, nos queda:

$$\underbrace{\lambda}_{\alpha_1} \ \wedge \ \underbrace{(1 \leq i)}_{\alpha_2} \ \wedge \ \underbrace{(i \leq n+1)}_{\alpha_3} \ \wedge \ \underbrace{(q \leftrightarrow \mu(i-1))}_{\alpha_4} \ \wedge \ \underbrace{(q = True)}_{\alpha_5} \ \wedge \ \underbrace{(i \neq n+1)}_{\alpha_6}$$

En la segunda parte (parte derecha o segunda línea) tenemos cuatro preguntas: ¿λ? ¿$0 \leq i$? ¿$i \leq n$? ¿$q \leftrightarrow \mu(i)$?

   - ¿λ? Sí, por α_1.
   - ¿$0 \leq i$? Sí, por α_2.
   - ¿$i \leq n$? Sí, por α_3 y α_6.
   - ¿$q \leftrightarrow \mu(i)$? Considerando el significado de $\mu(i)$, la pregunta queda de la siguiente forma:

$$¿q \leftrightarrow \forall k(1 \leq k \leq i \ \rightarrow \ A(k) \ mod \ x = s)?$$

     Si utilizamos abreviaturas, tenemos lo siguiente:

$$¿q \leftrightarrow \forall k(1 \leq k \leq i \ \rightarrow \ \gamma(k))?$$

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

$$¿q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1) \wedge \ \gamma(i))? \tag{11}$$

     De α_4 sabemos que se cumple lo siguiente:

$$q \leftrightarrow \forall k(1 \leq k \leq i-1 \ \rightarrow \ A(k) \ mod \ x = s)$$

     Si utilizamos abreviaturas, tenemos lo siguiente:

$$q \leftrightarrow \forall k(1 \leq k \leq i-1 \ \rightarrow \ \gamma(k))$$

     Si expresamos la fórmula universal como una conjunción, tenemos que se cumple lo siguiente:

$$q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1))$$

     Por α_5 sabemos que q es $True$. Por tanto, $\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1)$ es $True$.

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

Tabla 4: Las tres opciones para que la expresión $((q = False) \ \vee \ (i = n + 1))$ sea cierta.

Esquema de partida para el punto (III):

$\{INV \ \wedge \ B\}$ $\quad$ ¿$(INV \ \wedge \ B) \rightarrow INV$?

$\{INV\}$

Esquema de partida para el punto (VI):

$\{INV \ \wedge \ B \ \wedge \ E = v\}$ $\quad$ ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow E < v$?

$\{E < v\}$

Figura 5: Esquemas de partida para los puntos (III) y (VI).

Esquema para el punto (III) tras actualizar i:

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

$\{\varphi_3 \equiv def(i + 1) \ \wedge \ INV_i^{i+1}\}$

(AA) i := i + 1;

$\{INV\}$

Esquema para el punto (VI) tras actualizar i:

$\{INV \ \wedge \ B \ \wedge \ E = v\}$ $\quad$ ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_4$?

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

(AA) i := i + 1;

$\{E < v\}$

Figura 6: Esquemas para los puntos (III) y (VI) tras actualizar i.

En definitiva, la pregunta (11) queda de la siguiente forma:

$$¿True \leftrightarrow (True \ \wedge \ \gamma(i))? \tag{12}$$

Puesto que $(True \wedge \delta) \equiv \delta$ para cualquier fórmula δ, la pregunta (12) queda de la siguiente forma:

$$¿True \leftrightarrow \gamma(i)? \tag{13}$$

Es decir:

$$¿True \leftrightarrow (A(i) \ mod \ x = s)? \tag{14}$$

En $INV \wedge B$ no tenemos información que nos sirva para responder a esa pregunta. Por consiguiente, la implicación $(INV \ \wedge \ B) \rightarrow \varphi_3$ no se cumple.

Hemos comprobado que la implicación $(INV \ \wedge \ B) \rightarrow \varphi_3$ no se cumple.

- Comprobación de la implicación: ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_4$?

$$¿\underbrace{\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1))}_{INV} \wedge \underbrace{(q = True) \ \wedge \ (i \neq n+1)}_{B} \wedge \underbrace{(n+1-i = v)}_{E=v}$$

$$\rightarrow \underbrace{(n-i < v)}_{\varphi_4}?$$

En la primera parte de la implicación tenemos la información:
$\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (w \leftrightarrow \mu(i-1)) \ \wedge \ (q = True) \ \wedge \ (i \neq n+1) \ \wedge \ (n+1-i = v)$

Si descomponemos $(1 \leq i \leq n+1)$, nos queda:
$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(1 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n+1)}_{\alpha_3} \wedge \underbrace{(q \leftrightarrow \mu(i-1))}_{\alpha_4} \wedge \underbrace{(q = True)}_{\alpha_5} \wedge \underbrace{(i \neq n+1)}_{\alpha_6} \wedge$

$\underbrace{(n+1-i = v)}_{\alpha_7}$

En la segunda parte de la implicación tenemos una pregunta: ¿$n - i < v$?

- ¿$n-i < v$? Por α_7 sabemos que $(n+1-i = v)$. Si restamos 1 a ambos lados de la igualdad, nos queda $(n+1-i-1 = v-1)$, es decir, $(n-i = v-1)$. Puesto que para cualquier número entero z se cumple $z-1 < z$, también se cumple $v-1 < v$. Por tanto, deducimos que $(n-i < v)$.

Hemos comprobado que la implicación $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_4$ se cumple.

El que no se cumpla la implicación $(INV \ \wedge \ B) \rightarrow \varphi_3$ conlleva que en lo que respecta al punto (III) sí haya que añadir alguna asignación, siempre con el objetivo de que se cumpla φ_3. Pero el que se cumpla la implicación $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow E < v$ conlleva que en lo que respecta al punto (VI) no haya necesidad de añadir ninguna asignación.

5. **Objetivo:** que se cumpla φ_3:
Hemos visto que $INV \ \wedge \ B$ no implica φ_3. En concreto, dentro de φ_3 el componente que no se cumple es $q \leftrightarrow \mu(i)$. De manera extendida, lo que no se cumple es la siguiente equivalencia:

$$¿q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1) \ \wedge \ \gamma(i))?$$

De α_4 sabemos que se cumple lo siguiente:

$$q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1))$$

Sabiendo que el valor de q coincide con el valor de $\gamma(1) \wedge \gamma(2) \wedge \cdots \wedge \gamma(i-1)$ y sabiendo que lo que se quiere es que el valor de q coincida con el valor de $\gamma(1) \wedge \gamma(2) \wedge \cdots \wedge \gamma(i-1) \wedge \gamma(i)$, vemos que el único elemento que le falta a q para que se cumpla eso, es que se le añada $\wedge \gamma(i)$ al valor que ya tiene.

Por tanto, la asignación que se necesita es $q := q \wedge \gamma(i);$.

Esa asignación hay que añadirla tanto en el punto (III) —encima de φ_3— como en el (VI) —encima de φ_4—, porque las instrucciones han de ser las mismas en ambos casos. Una vez añadida la asignación, se ha de calcular la fórmula correspondiente, utilizando para ello el axioma de la asignación (AA).

Puesto que por α_5 sabemos que q es $True$, teniendo en cuenta que $(True \wedge \delta) \equiv \delta$ para cualquier δ, la asignación puede ponerse también como $q := \gamma(i);$. De todas formas, en los siguientes apartados utilizaremos la asignación $q := q \wedge \gamma(i);$.

6. Cálculo de φ_5 y φ_6:

   - Cálculo de la fórmula φ_5 a partir de la fórmula φ_3 utilizando el axioma de la asignación (AA).

$$\begin{aligned}
\varphi_5 \ &\equiv \ def(q \wedge \gamma(i)) \ \wedge \ (\varphi_3)_q^{q \wedge \gamma(i)} \\
&\equiv \ def(q \wedge (A(i) \ mod \ x = s)) \ \wedge \ \lambda \ \wedge \ (0 \leq i \leq n) \ \wedge \ ((q \wedge \gamma(i)) \leftrightarrow \mu(i)) \\
&\equiv \ \underbrace{(1 \leq i \leq n)}_{A(i)} \wedge \underbrace{(x \neq 0)}_{mod \ x} \ \wedge \ \lambda \ \wedge \ (0 \leq i \leq n) \ \wedge \ ((q \wedge \gamma(i)) \leftrightarrow \mu(i)) \\
&\equiv \ (1 \leq i \leq n) \ \wedge \ \lambda \ \wedge \ (0 \leq i \leq n) \ \wedge \ ((q \wedge \gamma(i)) \leftrightarrow \mu(i)) \\
&\equiv (1 \leq i \leq n) \ \wedge \ \lambda \ \wedge \ ((q \wedge \gamma(i)) \leftrightarrow \mu(i)) \\
&\equiv (1 \leq i) \ \wedge \ (i \leq n) \ \wedge \ \lambda \ \wedge \ ((q \wedge \gamma(i)) \leftrightarrow \mu(i))
\end{aligned}$$

   En la parte correspondiente a $def(q \wedge \gamma(i))$, se ha puesto el intervalo $(1 \leq i \leq n)$ por $A(i)$ y se ha puesto $(x \neq 0)$ porque aparece el resto de la división por x. Se ha eliminado $(x \neq 0)$ porque en λ se dice $x \geq 2$. Al decir $x \geq 2$, también se dice $(x \neq 0)$ y, por tanto, se puede eliminar $(x \neq 0)$. Para determinar el intervalo de i a partir de $(1 \leq i \leq n)$ y $(0 \leq i \leq n)$, se han cogido el mayor de los límites inferiores (1) y el menor de los límites superiores (n). Finalmente, se ha separado $(1 \leq i \leq n)$ en dos trozos: $(1 \leq i)$ e $(i \leq n)$.

   - Cálculo de la fórmula φ_6 a partir de la fórmula φ_4 utilizando el axioma de la asignación (AA).

$$\begin{aligned}
\varphi_6 \ &\equiv \ def(q \wedge \gamma(i)) \ \wedge \ (\varphi_4)_q^{q \wedge \gamma(i)} \\
&\equiv \ def(q \wedge (A(i) \ mod \ x = s)) \ \wedge \ (n - i < v)_q^{q \wedge \gamma(i)} \\
&\equiv \ \underbrace{(1 \leq i \leq n)}_{A(i)} \wedge \underbrace{(x \neq 0)}_{mod \ x} \ \wedge \ (n - i < v) \\
&\equiv (1 \leq i) \ \wedge \ (i \leq n) \ \wedge \ (x \neq 0) \ \wedge \ (n - i < v)
\end{aligned}$$

   De la misma forma que al calcular φ_5, en la parte correspondiente a $def(q \wedge \gamma(i))$, se ha puesto el intervalo $(1 \leq i \leq n)$ por $A(i)$ y se ha puesto $(x \neq 0)$ porque aparece el resto de la división por x. Pero a diferencia de lo que ha ocurrido al calcular φ_5, en φ_6 no es posible eliminar $(x \neq 0)$ porque en φ_6 no aparece λ. Finalmente, para obtener la versión final de φ_6, se ha separado $(1 \leq i \leq n)$ en dos trozos: $(1 \leq i)$ e $(i \leq n)$.

En la figura 7 de la página 18, se muestran los esquemas para los puntos (III) y (VI) tras la actualización de q. Una vez que se han calculado φ_5 y φ_6, se han de comprobar las implicaciones $(INV \ \wedge \ B) \rightarrow \varphi_5$ e $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_6$.

7. ¿$(INV \ \wedge \ B) \rightarrow \varphi_5$? ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_6$?

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

   $$¿\underbrace{\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1))}_{INV} \ \wedge \ \underbrace{(q = True) \ \wedge \ (i \neq n+1)}_{B} \rightarrow$$

   $$\underbrace{(1 \leq i) \ \wedge \ (i \leq n) \ \wedge \ \lambda \ \wedge \ ((q \wedge \gamma(i)) \leftrightarrow \mu(i))}_{\varphi_5}?$$

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

   $$\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1)) \ \wedge \ (q = True) \ \wedge \ (i \neq n+1)$$

   Si descomponemos $(1 \leq i \leq n+1)$, nos queda:

   $$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(1 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n+1)}_{\alpha_3} \wedge \underbrace{(q \leftrightarrow \mu(i-1))}_{\alpha_4} \wedge \underbrace{(q = True)}_{\alpha_5} \wedge \underbrace{(i \neq n+1)}_{\alpha_6}$$

   En la segunda parte (parte derecha o segunda línea) tenemos cuatro preguntas: ¿$1 \leq i$? ¿$i \leq n$? ¿λ? ¿$(q \wedge \gamma(i)) \leftrightarrow \mu(i)$?

   - ¿$1 \leq i$? Sí, por α_2.
   - ¿$i \leq n$? Sí, por α_3 y α_6.
   - ¿λ? Sí, por α_1.
   - ¿$(q \wedge \gamma(i)) \leftrightarrow \mu(i)$? Considerando el significado de $\mu(i)$, la pregunta queda de la siguiente forma:

     $$¿(q \wedge \gamma(i)) \leftrightarrow \forall k(1 \leq k \leq i \ \rightarrow \ A(k) \ mod \ x = s)?$$

     Si utilizamos abreviaturas, la pregunta se puede reescribir de la siguiente forma:

     $$¿(q \wedge \gamma(i)) \leftrightarrow \forall k(1 \leq k \leq i \ \rightarrow \ \gamma(k))?$$

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

     $$¿(q \wedge \gamma(i)) \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1) \wedge \ \gamma(i))? \tag{15}$$

     De α_4 sabemos que se cumple lo siguiente:

     $$q \leftrightarrow \forall k(1 \leq k \leq i-1 \ \rightarrow \ A(k) \ mod \ x = s)$$

     Si utilizamos abreviaturas, esa equivalencia se puede reescribir de la siguiente forma:

     $$q \leftrightarrow \forall k(1 \leq k \leq i-1 \ \rightarrow \ \gamma(k))$$

     Si expresamos la fórmula universal como una conjunción, tenemos que se cumple lo siguiente:

     $$q \leftrightarrow (\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1))$$

     Por α_5 sabemos que q es $True$. Por tanto, $\gamma(1) \ \wedge \ \gamma(2) \ \wedge \cdots \wedge \ \gamma(i-1)$ es $True$.

Esquema para el punto (III) tras actualizar q:

$\{INV \ \wedge \ B\}$ ¿$(INV \ \wedge \ B) \rightarrow \varphi_5$?

$\{\varphi_5 \equiv def(q \wedge \gamma(i)) \ \wedge \ (\varphi_3)_q^{q \wedge \gamma(i)}\}$

(AA) q := $q \wedge \gamma(i)$;

$\{\varphi_3 \equiv def(i+1) \ \wedge \ INV_i^{i+1}\}$

(AA) i := i + 1;

$\{INV\}$

Esquema para el punto (VI) tras actualizar q:

$\{INV \ \wedge \ B \ \wedge \ E = v\}$ ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_6$?

$\{\varphi_6 \equiv def(q \wedge \gamma(i)) \ \wedge \ (\varphi_4)_q^{q \wedge \gamma(i)}\}$

(AA) q := $q \wedge \gamma(i)$;

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

(AA) i := i + 1;

$\{E < v\}$

$\gamma(i)$ representa $(A(i) \ mod \ x = s)$.

Figura 7: Esquemas para los puntos (III) y (VI) tras actualizar q.

En definitiva, la pregunta (15) queda de la siguiente forma:

$$¿(True \wedge \gamma(i)) \leftrightarrow (True \ \wedge \ \gamma(i))? \tag{16}$$

Puesto que $(True \wedge \delta) \equiv \delta$ para cualquier fórmula δ, la pregunta (16) queda de la siguiente forma:

$$¿\gamma(i) \leftrightarrow \gamma(i)? \tag{17}$$

La respuesta es afirmativa, ya que $(\delta \leftrightarrow \delta) \equiv True$ para cualquier fórmula δ.

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

- Comprobación de la implicación: $¿(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_6$?

$$¿\underbrace{\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1))}_{INV} \wedge \underbrace{(q = True) \ \wedge \ (i \neq n+1)}_{B} \wedge \underbrace{(n+1-i = v)}_{E=v}$$

$$\rightarrow \underbrace{(1 \leq i) \ \wedge \ (i \leq n) \ \wedge \ (x \neq 0) \ \wedge \ (n-i < v)}_{\varphi_6}?$$

En la primera parte de la implicación tenemos la información:
$\lambda \ \wedge \ (1 \leq i \leq n+1) \ \wedge \ (q \leftrightarrow \mu(i-1)) \ \wedge \ (q = True) \ \wedge \ (i \neq n+1) \ \wedge \ (n+1-i = v)$

Si descomponemos $(1 \leq i \leq n+1)$, nos queda:
$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(1 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n+1)}_{\alpha_3} \wedge \underbrace{(q \leftrightarrow \mu(i-1))}_{\alpha_4} \wedge \underbrace{(q = True)}_{\alpha_5} \wedge \underbrace{(i \neq n+1)}_{\alpha_6} \wedge$

$\underbrace{(n+1-i = v)}_{\alpha_7}$

En la segunda parte de la implicación tenemos cuatro preguntas: $¿1 \leq i?$ $¿i \leq n?$ $¿(x \neq 0)?$ $¿n - i < v?$

- $¿1 \leq i?$ Sí, por α_2.
- $¿i \leq n?$ Sí, por α_3 y α_6.
- $¿x \neq 0?$ En α_1, es decir, en λ, tenemos que se cumple $x \geq 2$. Por tanto, x es distinto de 0.
- $¿n-i < v?$ Por α_7 sabemos que $(n+1-i = v)$. Si restamos 1 a ambos lados de la igualdad, nos queda $(n+1-i-1 = v-1)$, es decir, $(n-i = v-1)$. Puesto que para cualquier número entero z se cumple $z-1 < z$, también se cumple $v-1 < v$. Por tanto, deducimos que $(n-i < v)$.

Hemos comprobado que la implicación $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_6$ se cumple.

Puesto que se han cumplido las implicaciones $(INV \ \wedge \ B) \ \rightarrow \varphi_5$ e $(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow \varphi_6$, no hace falta añadir más asignaciones. Consecuentemente, hemos terminado la derivación del prorgrama.

1.2.4 *(d)* Escribir el programa completo al final

En la figura 8 de la página 20, se muestra el programa que se ha construido. Ahí aparecen también las fórmulas que se han calculado.

Programa derivado:

$\{\varphi\}$
$\{\varphi_2\}$
q := True;
$\{\varphi_1\}$
i := 1;
while $\{INV\}$ $\{E\}$ (q = True **and** i $\neq$ n + 1) **loop**
$\{\varphi_5\}$ $\{\varphi_6\}$
q := (q **and** (A(i) mod x = s));
$\{\varphi_3\}$ $\{\varphi_4\}$
i := i + 1;
end loop;
$\{\psi\}$

Figura 8: Programa derivado.

Letras griegas adicionales utilizadas en la solución:
α : alfa β : beta δ : delta π : pi

Tabla 5: Denominaciones de las letras griegas adicionales utilizadas en la solución.