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
2 puntos

Modelo de examen: t4m3-∃

Solución

Última actualización: 05 - 04 - 2022

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, 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 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 (página 4) se recogen las abreviaciones que se recomienda utilizar durante el proceso de derivación. En la tabla 2 (página 4) se recopilan las denominaciones de las letras griegas utilizadas en este enunciado. Finalmente, en la tabla 3 (página 6) 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*

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 $while$.

- ¿$\varphi \rightarrow INV$?

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

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

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

  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 hacer 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 $q \leftrightarrow \mu(i)$, 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 $n \geq 2$. Con esa información en mano, podemos asegurar que si asignamos a i el valor 1 o el valor n, entonces i cumplirá tanto $1 \leq i$ como $i \leq n$. 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 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 el vector de izquierda a derecha. Si la expresión cota tiene la forma $i - z$, siendo z un número

Estructura del programa a derivar:

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

Definición de φ, INV, E y ψ:

$$\varphi \equiv \quad n \geq 2 \ \wedge \ posit(A(1..n))$$

$$INV \equiv \quad n \geq 2 \ \wedge \ posit(A(1..n)) \ \wedge \ (1 \leq i \leq n) \wedge \\ q \leftrightarrow \exists k((2 \leq k \leq i) \ \wedge \ ((A(k) \ mod \ k) = 0))$$

$$E = \quad n - i$$

$$\psi \equiv \quad q \leftrightarrow \exists k((2 \leq k \leq n) \ \wedge \ ((A(k) \ mod \ k) = 0))$$

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 ψ : psi γ : gamma μ : mu λ : lambda

Tabla 2: Denominaciones de las letras griegas utilizadas.

Esquema de partida para el cálculo de las inicializaciones:

$\{\varphi\}$ $\quad \iota\varphi \rightarrow INV?$

$\{INV\}$

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

entero, ello significa que hay que recorrer el vector de derecha a izquierda.

En nuestro caso, E tiene la forma $n+z-i$ con $z=0$. Por tanto, hay que recorrer el vector de izquierda a derecha.

Consecuentemente, se ha de inicializar i con 1. Una vez que colocamos 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 (página 6), 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).

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

Para simplificar la expresión, primero se ha descompuesto $(1 \leq 1 \leq n)$ en $(1 \leq 1) \ \wedge \ (1 \leq n)$. 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 tenido en cuenta que $True \wedge \delta \equiv \delta$ para cualquier fórmula δ y se han eliminado las apariciones de $True$.

- ¿$\varphi \rightarrow \varphi_1$?

$$¿\underbrace{\lambda}_{\varphi} \rightarrow \underbrace{\lambda \ \wedge \ (1 \leq n) \ \wedge \ (q \leftrightarrow \mu(1))}_{\varphi_1}? \tag{2}$$

En la primera parte de la implicación (φ) tenemos la información y en la segunda parte (φ_1) tenemos tres preguntas: ¿λ? ¿$1 \leq n$? ¿$q \leftrightarrow \mu(1)$?

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.
  - ¿$1 \leq n$? En φ se nos dice que $n \geq 2$. Si n es mayor o igual que 2, entonces necesariamente ha de ser también mayor o igual que 1. No es posible tener un número que sea mayor o igual que 2 y que no sea mayor o igual que 1. Por tanto, se cumple $1 \leq n$. Técnicamente, podemos argumentar de la siguiente forma: ¿$1 \leq n$? es equivalente a ¿$(1 < n) \ \vee \ (1 = n)$?; por estar en $\mathbb{Z}$, eso es equivalente a ¿$(2 \leq n) \ \vee \ (1 = n)$?; en φ se nos dice que $n \geq 2$; por tanto, la pregunta es ¿$True \ \vee \ (1 = n)$?; y eso es equivalente a ¿$True$?. La respuesta es que sí, es decir, $True$ es cierto.
  - ¿$q \leftrightarrow \mu(1)$? De la información que se tiene en φ no se puede deducir esto. No sabemos si se cumple o no se cumple $q \leftrightarrow \mu(1)$.

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 $q \leftrightarrow \mu(1)$. El significado de $\mu(1)$ es el siguiente:

$$\exists k(2 \leq k \leq 1 \ \wedge \ A(k) \ mod \ k = 0) \tag{3}$$

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.

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.

El dominio $2 \leq k \leq 1$ de esa fórmula existencial es vacío. Consiguientemente, el valor de la fórmula es $False$. Es decir, $\mu(1)$ es $False$. Por tanto, el objetivo es que se cumpla $q \leftrightarrow False$. Para conseguir ese objetivo, hay que asignar $False$ a q.

Una vez que colocamos la asignación $q := False$; 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).

$$\begin{aligned}\varphi_2 &\equiv def(False) \ \wedge \ (\varphi_1)_q^{False} \\ &\equiv True \ \wedge \ \lambda \ \wedge \ (1 \leq n) \ \wedge \ (False \leftrightarrow \mu(1)) \\ &\equiv \lambda \ \wedge \ (1 \leq n) \ \wedge \ (False \leftrightarrow \mu(1))\end{aligned}$$

Para simplificar la fórmula φ_2, se ha tenido en cuenta que $True \wedge \delta \equiv \delta$ para cualquier fórmula δ y se ha eliminado la aparición de $True$.

- ¿$\varphi \rightarrow \varphi_2$?

$$¿\underbrace{\lambda}_{\varphi} \rightarrow \underbrace{\lambda \ \wedge \ (1 \leq n) \ \wedge \ (False \leftrightarrow \mu(1))}_{\varphi_2}? \tag{4}$$

En la primera parte de la implicación (φ) tenemos la información y en la segunda parte (φ_2) tenemos tres preguntas: ¿λ? ¿$1 \leq n$? ¿$False \leftrightarrow \mu(1)$?

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

  - ¿λ? Sí, porque en φ tenemos λ.
  - ¿$1 \leq n$? Puesto que en φ, es decir, en λ, se nos dice que $n \geq 2$, podemos afirmar que también se cumple $1 \leq n$.
  - ¿$False \leftrightarrow \mu(1)$? En la fórmula (3) de la página 5 se constata que $\mu(1)$ representa una fórmula existencial de dominio vacío. Consecuentemente, su valor es $False$. Por tanto, la pregunta es la siguiente: ¿$False \leftrightarrow False$? 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 extraida 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 $True$ en q cuando se encuentre una posición de $A(1..n)$ mayor o igual a 2, para la cual se cumpla que el elemento de esa posición es múltiplo de la posición. Es decir, se sabrá

que hay que devolver $True$ en q si se cumple que en el tramo recorrido desde la posición 2 hasta la actual posición i, se tiene que algún elemento del vector es múltiplo de la posición que ocupa. Por tanto, cuando la fórmula $\exists k(2 \leq k \leq i \ \wedge \ A(k) \ mod \ k = 0)$ —que aparece en el invariante— sea $True$. 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 $\exists k(2 \leq k \leq i \ \wedge \ A(k) \ mod \ k = 0)$ no se puede poner directamente en 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 $True$ en cuanto q tome el valor $True$.

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

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

$$\neg B \equiv (q = True) \vee (i = n)$$

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

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

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 $while$.

$$\begin{array}{l} ¿INV \rightarrow def(B)? \\ ¿INV \rightarrow def((q = False) \wedge (i \neq n))? \\ ¿INV \rightarrow True? \end{array}$$

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. Recordemos que $(\delta \rightarrow True)$ es equivalente a $((\neg\delta) \vee True)$. Por consiguiente, estamos ante una disyunción donde uno de los componentes es $True$. Eso conlleva que toda la fórmula $((\neg\delta) \vee True)$ sea $True$.

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

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

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

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

$$\lambda \ \wedge \ (1 \leq i \leq n) \ \wedge \ (q \leftrightarrow \mu(i)) \ \wedge \ ((q = True) \ \vee \ (i = n))$$

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

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

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

$$¿q \leftrightarrow \mu(n)?$$

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 11).

- Casos 1 y 3: $i = n$

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

  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 11.

- Caso 2: $q = True$ e $i \neq n$

  Si se cumple $i \neq n$, entonces por α_3 deducimos que $i < n$. Consecuentemente, α_4 y ψ no son la misma fórmula.

  Recordemos que tenemos que responder a la siguiente pregunta:

  $$¿q \leftrightarrow \mu(n)?$$

  Si consideramos el significado de $\mu(n)$, tenemos la siguiente pregunta:

  $$¿q \leftrightarrow \exists k(2 \leq k \leq n \ \wedge \ A(k) \ mod \ k = 0)?$$

  También podemos reescribir esa pregunta de la siguiente forma:

  $$¿q \leftrightarrow \exists k(2 \leq k \leq n \ \wedge \ \gamma(k))?$$

  La fórmula existencial se puede expresar mediante la disyunción:

  $$¿q \leftrightarrow (\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(n))? \tag{5}$$

  Por α_4 sabemos que se cumple lo siguiente:

  $$q \leftrightarrow \exists k(2 \leq k \leq i \ \wedge \ A(k) \ mod \ k = 0)$$

  También podemos reescribir esa equivalencia de la siguiente forma:

  $$q \leftrightarrow \exists k(2 \leq k \leq i \ \wedge \ \gamma(k))$$

  Si sustituimos la fórmula existencial por la disyunción, tenemos que se cumple lo siguiente:

  $$q \leftrightarrow (\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i)) \tag{6}$$

  De α_3 e $i \neq n$ hemos deducido que $i \leq n - 1$. Por tanto, el número de gammas de (6) es menor que el número de gammas de (5).

  Consecuentemente, podemos reescribir la pregunta (5) de la siguiente manera:

  $$¿q \leftrightarrow (\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i) \ \vee \ \gamma(i+1) \ \vee \cdots \vee \ \gamma(n))? \tag{7}$$

  Puesto que estamos considerando el caso en el que $q = True$ e $i \neq n$, por la equivalencia (6) de la página 9 deducimos que $\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i)$ es $True$. Por tanto, la pregunta (7) de la página 9 se puede reescribir de la siguiente forma:

  $$¿True \leftrightarrow (True \ \vee \ \gamma(i+1) \ \vee \cdots \vee \ \gamma(n))? \tag{8}$$

No tenemos ninguna información sobre la disyunción $\gamma(i+1) \ \vee \cdots \vee \ \gamma(n)$, pero utilizando la equivalencia lógica $(True \vee \delta) \equiv True$, la pregunta (8) de la página 9 se puede reescribir de la siguiente forma:

$$¿True \leftrightarrow True?$$

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 $q = True$ e $i \neq n$, sí nos hace falta conocer el valor de q.

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

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

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

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

$$\lambda \ \wedge \ (1 \leq i \leq n) \ \wedge \ (q \leftrightarrow \mu(i)) \ \wedge \ (q = False) \ \wedge \ (i \neq n)$$

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

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

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

$$¿n - i > 0?$$

Por β_3 y β_6 deducimos que $n > i$. Restando i en ambos lados, obtenemos $n - i > i - i$, es decir, $n - i > 0$. 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 a $(\neg(\delta_1 \wedge \delta_2) \vee \delta_1)$, 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, $((\neg\delta_1) \vee (\neg\delta_2) \vee \delta_1) \equiv True \vee (\neg\delta_2) \equiv True$.

Esquema tras inicializar la variable q:

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

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

(AA) q := False;

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

(AA) i := 1;

$\{INV\}$

Figura 4: Inicialización de q.

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

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

Esquema de partida para el punto (III):

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

$\{INV\}$

Esquema de partida para el punto (VI):

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

$\{E < v\}$

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

- ¿$(INV \ \wedge \ B \ \wedge \ E = v) \rightarrow E < v$? Se nos dice que las fórmulas INV, B y $E = v$ son ciertas y se nos pregunta si $E < v$ 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 $n - i < v$ sabiendo que ahora se cumple $E = v$, es decir, $n - i = v$.

$n - i$ representa la distancia entre el punto al que hay que llegar (n) 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) 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).

$$\begin{aligned} \varphi_3 \ &\equiv \ def(i+1) \ \wedge \ INV_i^{i+1} \\ &\equiv \ True \ \wedge \ \lambda \ \wedge \ (1 \leq i+1 \leq n) \ \wedge \ (q \leftrightarrow \mu(i+1)) \\ &\equiv \lambda \ \wedge \ (0 \leq i \leq n-1) \ \wedge \ (q \leftrightarrow \mu(i+1)) \end{aligned}$$

Para simplificar la expresión φ_3, se ha tenido en cuenta que $True \wedge \delta \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(1 \leq i+1 \leq n)$ en $(0 \leq i \leq n-1)$, restando 1 en los tres componentes.

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

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

Para simplificar la expresión φ_4, se ha tenido en cuenta que $True \wedge \delta \equiv \delta$ para cualquier fórmula δ. Por otra parte, se ha transformado $(n - (i+1) < v)$ en $(n - i - 1 < v)$, aplicando el signo negativo a $(i+1)$.

En la figura 6 de la página 18 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: ¿$(INV \ \wedge \ B) \rightarrow \varphi_3$?

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

$$\underbrace{\lambda \ \wedge \ (0 \leq i \leq n-1) \ \wedge \ (q \leftrightarrow \mu(i+1))}_{\varphi_3}?$$

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

$$\lambda \ \wedge \ (1 \leq i \leq n) \ \wedge \ (q \leftrightarrow \mu(i)) \ \wedge \ (q = False) \ \wedge \ (i \neq n)$$

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

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

En la segunda parte (parte derecha o segunda línea) tenemos cuatro preguntas: ¿λ? ¿$0 \leq i$? ¿$i \leq n-1$? ¿$q \leftrightarrow \mu(i+1)$?

- ¿λ? Sí, por α_1.
- ¿$0 \leq i$? Sí, por α_2.
- ¿$i \leq n-1$? Sí, por α_3 y α_6.
- ¿$q \leftrightarrow \mu(i+1)$? Considerando el significado de $\mu(i+1)$, la pregunta queda de la siguiente forma:

  $$¿q \leftrightarrow \exists k(2 \leq k \leq i+1 \ \wedge \ A(k) \ mod \ k = 0)?$$

  Esa pregunta la podemos representar también de la siguiente forma:

  $$¿q \leftrightarrow \exists k(2 \leq k \leq i+1 \ \wedge \ \gamma(k))?$$

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

  $$¿q \leftrightarrow (\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i) \vee \ \gamma(i+1))? \tag{9}$$

  De α_4 sabemos que se cumple lo siguiente:

  $$q \leftrightarrow \exists k(2 \leq k \leq i \ \wedge \ A(k) \ mod \ k = 0)$$

  Esa fórmula la podemos representar también de la siguiente manera:

  $$q \leftrightarrow \exists k(2 \leq k \leq i \ \wedge \ \gamma(k))$$

  Si expresamos la fórmula existencial como una disyunción, tenemos que se cumple lo siguiente:

  $$q \leftrightarrow (\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i))$$

  Por α_5 sabemos que q es $False$. Por tanto, $\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i)$ es $False$.
  En definitiva, la pregunta (9) queda de la siguiente forma:

  $$¿False \leftrightarrow (False \ \vee \ \gamma(i+1))? \tag{10}$$

  Puesto que $False \vee \delta \equiv \delta$ para cualquier fórmula δ, la pregunta (10) queda de la siguiente forma:

  $$¿False \leftrightarrow \gamma(i+1)? \tag{11}$$

Es decir:

$$¿False \leftrightarrow (A(i+1)\ mod\ (i+1) = 0)? \tag{12}$$

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: ¿$(INV\ \wedge\ B\ \wedge\ E = v) \rightarrow \varphi_4$?

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

$$\rightarrow \underbrace{(n - i - 1 < v)}_{\varphi_4}?$$

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

Si descomponemos $(1 \leq i \leq n)$, nos queda:
$\underbrace{\lambda}_{\alpha_1} \wedge \underbrace{(1 \leq i)}_{\alpha_2} \wedge \underbrace{(i \leq n)}_{\alpha_3} \wedge \underbrace{(q \leftrightarrow \mu(i))}_{\alpha_4} \wedge \underbrace{(q = False)}_{\alpha_5} \wedge \underbrace{(i \neq n)}_{\alpha_6} \wedge \underbrace{(n - i = v)}_{\alpha_7}$

En la segunda parte de la implicación tenemos una pregunta: ¿$n - i - 1 < v$?

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

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 $(INV\ \wedge\ B\ \wedge\ E = v) \rightarrow \varphi_4$ 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 $q \leftrightarrow \mu(i+1)$. De manera extendida, lo que no se cumple es la siguiente equivalencia:

$$¿q \leftrightarrow (\gamma(2)\ \vee\ \gamma(3)\ \vee \cdots \vee\ \gamma(i) \vee\ \gamma(i+1))?$$

De α_4 sabemos que se cumple lo siguiente:

$$q \leftrightarrow (\gamma(2)\ \vee\ \gamma(3)\ \vee \cdots \vee\ \gamma(i))$$

Sabiendo que el valor de q coincide con el valor de $\gamma(2)\ \vee\ \gamma(3)\ \vee \cdots \vee\ \gamma(i)$ y sabiendo que lo que se quiere es que el valor de q coincida con el valor de $\gamma(2)\ \vee\ \gamma(3)\ \vee \cdots \vee\ \gamma(i) \vee\ \gamma(i+1)$, vemos que el único elemento que le falta a q para que se cumpla eso, es que se le añada $\vee\ \gamma(i+1)$ al valor que ya tiene.

Por tanto, la asignación que se necesita es $q := q \vee \gamma(i+1);$.

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 $False$, teniendo en cuenta que $False \vee \delta \equiv \delta$ para cualquier δ, la asignación puede ponerse también como $q := \gamma(i+1);$. De todas formas, en los siguientes apartados utilizaremos la asignación $q := q \vee \gamma(i+1);$.

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).

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

   Para simplificar φ_5, primero se ha transformado $(1 \le i+1 \le n)$ en $(0 \le i \le n-1)$ restando 1 a los tres componentes. Puesto que el intervalo $(0 \le i \le n-1)$ ha aparecido dos veces, se eliminado una de las copias. A partir de $((i+1) \neq 0)$ se obtenido $(i \neq -1)$ restando 1 en ambos lados. La propiedad $(i \neq -1)$ se puede eliminar porque está incluida en la propiedad $(0 \le i \le n-1)$. Es decir, si i cumple $(0 \le i \le n-1)$, entonces también cumple necesariamente $(i \neq -1)$. Finalmente, se ha separado $(0 \le i \le n-1)$ en dos trozos: $(0 \le i)$ y $(i \le n-1)$.

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

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

   Para simplificar φ_6, primero se ha transformado $(1 \le i+1 \le n)$ en $(0 \le i \le n-1)$ restando 1 a los tres componentes. A partir de $((i+1) \neq 0)$ se obtenido $(i \neq -1)$ restando 1 en ambos lados. La propiedad $(i \neq -1)$ se puede eliminar porque está incluida en la propiedad $(0 \le i \le n-1)$. Es decir, si i cumple $(0 \le i \le n-1)$, entonces también cumple necesariamente $(i \neq -1)$. Finalmente, se ha separado $(0 \le i \le n-1)$ en dos trozos: $(0 \le i)$ y $(i \le n-1)$.

En la figura 7 de la página 19 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: ¿$(INV \ \wedge \ B) \rightarrow \varphi_5$?

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

   $$\underbrace{\lambda \ \wedge \ (0 \leq i) \ \wedge \ (i \leq n-1) \ \wedge \ ((q \vee \gamma(i+1)) \leftrightarrow \mu(i+1))}_{\varphi_5}?$$

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

   $$\lambda \ \wedge \ (1 \leq i \leq n) \ \wedge \ (q \leftrightarrow \mu(i)) \ \wedge \ (q = False) \ \wedge \ (i \neq n)$$

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

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

   En la segunda parte (parte derecha o segunda línea) tenemos cuatro preguntas: ¿λ? ¿$0 \leq i$? ¿$i \leq n-1$? ¿$(q \vee \gamma(i+1)) \leftrightarrow \mu(i+1)$?

   - ¿λ? Sí, por α_1.
   - ¿$0 \leq i$? Sí, por α_2.
   - ¿$i \leq n-1$? Sí, por α_3 y α_6.
   - ¿$(q \vee \gamma(i+1)) \leftrightarrow \mu(i+1)$? Considerando el significado de $\mu(i+1)$, la pregunta queda de la siguiente forma:

     $$¿(q \vee \gamma(i+1)) \leftrightarrow \exists k(2 \leq k \leq i+1 \ \wedge \ A(k) \ mod \ k = 0)?$$

     Esa pregunta se puede reescribir de la siguiente manera:

     $$¿(q \vee \gamma(i+1)) \leftrightarrow \exists k(2 \leq k \leq i+1 \ \wedge \ \gamma(k))?$$

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

     $$¿(q \vee \gamma(i+1)) \leftrightarrow (\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i) \vee \ \gamma(i+1))? \tag{13}$$

     De α_4 sabemos que se cumple lo siguiente:

     $$q \leftrightarrow \exists k(2 \leq k \leq i \ \wedge \ A(k) \ mod \ k = 0)$$

     Esa equivalencia se puede reescribir de la siguiente manera:

     $$q \leftrightarrow \exists k(2 \leq k \leq i \ \wedge \ \gamma(k))$$

     Si expresamos la fórmula existencial como una disyunción, tenemos que se cumple lo siguiente:

     $$q \leftrightarrow (\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i))$$

     Por α_5 sabemos que q es $False$. Por tanto, $\gamma(2) \ \vee \ \gamma(3) \ \vee \cdots \vee \ \gamma(i)$ es $False$.
     En definitiva, la pregunta (13) queda de la siguiente forma:

     $$¿(False \vee \gamma(i+1)) \leftrightarrow (False \ \vee \ \gamma(i+1))? \tag{14}$$

Puesto que $False \vee \delta \equiv \delta$ para cualquier fórmula δ, la pregunta (14) queda de la siguiente forma:

$$¿\gamma(i+1) \leftrightarrow \gamma(i+1)? \tag{15}$$

La respuesta es afirmativa porque $\delta \leftrightarrow \delta \equiv True$ para cualquier fórmula δ.

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

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

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

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

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

Si descomponemos $(1 \leq i \leq n)$, nos queda:
$\underbrace{\lambda}_{\alpha_1} \ \wedge \ \underbrace{(1 \leq i)}_{\alpha_2} \ \wedge \ \underbrace{(i \leq n)}_{\alpha_3} \ \wedge \ \underbrace{(q \leftrightarrow \mu(i))}_{\alpha_4} \ \wedge \ \underbrace{(q = False)}_{\alpha_5} \ \wedge \ \underbrace{(i \neq n)}_{\alpha_6} \ \wedge \ \underbrace{(n - i = v)}_{\alpha_7}$

En la segunda parte de la implicación tenemos tres preguntas: ¿$0 \leq i$? ¿$i \leq n-1$? ¿$n-i-1 < v$?

- ¿$0 \leq i$? Sí, por α_2.
- ¿$i \leq n - 1$? Sí, por α_3 y α_6.
- ¿$n - i - 1 < v$? Por α_7 sabemos que $(n - i = v)$. Si restamos 1 a ambos lados de la igualdad, nos queda $(n - i - 1 = v - 1)$. Puesto que para cualquier número entero z se cumple $z - 1 < z$, también se cumple $v - 1 < v$. Por tanto, deducimos que $(n - i - 1 < v)$.

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 programa.

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

En la figura 8 de la página 19, se muestra el programa que se ha construido. Ahí aparecen también las fórmulas que se han calculado.

Esquema para el punto (III) tras actualizar i:

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

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

(AA) i := i + 1;

$\{INV\}$

Esquema para el punto (VI) tras actualizar i:

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

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

(AA) i := i + 1;

$\{E < v\}$

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

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

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

Esquema para el punto (III) tras actualizar q:

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

$\{\varphi_5 \equiv def(q \vee \gamma(i+1)) \ \wedge \ (\varphi_3)_q^{q \vee \gamma(i+1)}\}$

(AA) $q := q \vee \gamma(i+1);$

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

(AA) i := i + 1;

$\{INV\}$

Esquema para el punto (VI) tras actualizar q:

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

$\{\varphi_6 \equiv def(q \vee \gamma(i+1)) \ \wedge \ (\varphi_4)_q^{q \vee \gamma(i+1)}\}$

(AA) $q := q \vee \gamma(i+1);$

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

(AA) i := i + 1;

$\{E < v\}$

$\gamma(i+1)$ representa $(A(i+1) \ mod \ (i+1) = 0)$.

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

Programa derivado:

$\{\varphi\}$
$\{\varphi_2\}$
q := False;
$\{\varphi_1\}$
i := 1;
while $\{INV\}$ $\{E\}$ (**not** q **and** i $\neq$ n) **loop**
$\{\varphi_5\}$ $\{\varphi_6\}$
q := (q **or** (A(i + 1) mod (i + 1) = 0));
$\{\varphi_3\}$ $\{\varphi_4\}$
i := i + 1;
end loop;
$\{\psi\}$

Figura 8: Programa derivado.