METODOLOGÍA DE LA PROGRAMACIÓN
SOLUCIONES DEL TEMA 3
VERIFICACIÓN DE PROGRAMAS

ÍNDICE

7. Programa que decide en la variable booleana ord si los elementos del vector A(1..n) están en orden creciente. -- #

$\{\varphi\} \equiv \{n \geq 1 \wedge i = 1\}$
ord : = true;
while {INV} $i \neq n$ **and** ord **loop**
    ord : = $(A(i) \leq A(i + 1))$;
    i : = i + 1;
end loop;
$\{\psi\} \equiv \{ord \leftrightarrow creciente(A(1..n))\}$

$\{INV\} \equiv \{(1 \leq i \leq n) \wedge (ord \leftrightarrow creciente(A(1..i)))\}$

$E = n - i$

$creciente(C(1..p)) \equiv \{\forall k(2 \leq k \leq p \rightarrow C(k - 1) \leq C(k))\}$

Esquema:

- Como aparte del while hay una asignación previa del while, hay que distinguir dos subprogramas:

$\{\varphi\}$
ord : = true;
{INV}

y

{INV}
while {INV} $i \neq n + 1$ **loop**
    ord : = $(A(i) \leq A(i + 1))$;
    i : = i + 1;
end loop;
$\{\psi\}$

- Primero se ha de verificar el primer subprograma

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

$\{\varphi_1\}$

ord : = true;

(AA)

$\{INV\}$

- $\{\varphi_1\} \equiv \{def(true) \wedge (INV)_{ord}^{true}\} \equiv$
  $\equiv \{true \wedge (1 \leq i \leq n) \wedge (true \leftrightarrow creciente(A(1..i)))\} \equiv^{simplificación}$
  $\equiv \{(1 \leq i \leq n) \wedge creciente(A(1..i))\}$

La simplificación está basada en que para cualquier fórmula δ se cumple por una parte que *true* $\wedge\, \delta \equiv \delta$ y por otra parte que *true* $\leftrightarrow \delta \equiv \delta$.

- ¿$\varphi \rightarrow \varphi_1$?

$$\{\underbrace{n \geq 1}_{\alpha} \wedge \underbrace{i = 1}_{\beta}\}$$

$$\downarrow ?$$

$$\underbrace{(1 \leq i \leq n)}_{\text{por } \alpha \text{ y } \beta} \wedge \underbrace{creciente(A(1..i))}_{\text{por } \beta}$$

Cuando i vale 1, en creciente(A(1..i)) tenemos un intervalo vacío y se cumple la fórmula.

- La aplicación de la Regla del While (RWH) al **segundo subprograma** supondrá realizar los siguientes cálculos y comprobaciones

  - Luego aplicamos la Regla del While (RWH) considerando que la precondición del while es {INV} y que la postcondición del while es $\{\psi\}$.

I. ¿$INV \rightarrow INV$?
II. ¿$INV \rightarrow def(B)$?

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

$\{\varphi_3\}$
ord : = (A(i) > A(i + 1));
(AA)
$\{\varphi_2\}$
i : = i + 1;
(AA)
{INV}

IV. ¿$(INV \wedge \neg B) \rightarrow \psi$?
V. ¿$(INV \wedge B) \rightarrow E > 0$?
VI. $\{INV \wedge B \wedge E = v\}$ ¿$(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?

$\{\varphi_5\}$
ord : = (A(i) ≤ A(i + 1));
(AA)
$\{\varphi_4\}$
i : = i + 1;
(AA)
$\{E < v\}$

I. ¿$INV \rightarrow INV$? Sí por ser iguales.

II. ¿$INV \rightarrow def(B)$?

¿$INV \rightarrow true$? Sí, porque la segunda parte de la implicación es true.

III.

- $\{\varphi_2\} \equiv \{def(i + 1) \wedge (INV)_i^{i+1}\} \equiv$
  $\equiv \{(1 \le i + 1 \le n) \wedge (ord \leftrightarrow creciente(A(1..i + 1)))\}$

- $\{\varphi_3\} \equiv \{def(A(i) \le A(i + 1)) \wedge (\varphi_2)_{ord}^{A(i) \le A(i+1)}\} \equiv$
  $\equiv \{(1 \le i \le n) \wedge (1 \le i + 1 \le n) \wedge (1 \le i + 1 \le n) \wedge$
  $(A(i) \le A(i + 1) \leftrightarrow creciente(A(1..i + 1)))\} \equiv^{simplificación}$
  $\equiv \{(1 \le i \le n) \wedge (1 \le i + 1 \le n) \wedge$
  $(A(i) \le A(i + 1) \leftrightarrow creciente(A(1..i + 1)))\} \equiv^{simplificación}$
  $\equiv \{(1 \le i \le n) \wedge (0 \le i \le n - 1) \wedge$
  $(A(i) \le A(i + 1) \leftrightarrow creciente(A(1..i + 1)))\} \equiv^{simplificación}$
  $\equiv \{(1 \le i \le n - 1) \wedge (A(i) \le A(i + 1) \leftrightarrow creciente(A(1..i + 1)))\}$

  En la primera simplificación se ha eliminado uno de los intervalos repetidos. En la segunda simplificación se ha pasado de tener $(1 \le i + 1 \le n)$ a tener $(0 \le i \le n - 1)$ restando 1 a los tres componentes. En la tercera simplificación, al tener dos intervalos distintos para i, hay que quedarse con el mayor de los límites inferiores (el mayor entre 1 y 0) y el menor de entre los límites superiores (el menor entre n y n – 1). Por ello el nuevo intervalo es $(1 \le i \le n - 1)$.

- ¿$(INV \wedge B) \rightarrow \varphi_3$?

$$\{\underbrace{(1 \le i \le n)}_{\alpha} \wedge \underbrace{(ord \leftrightarrow creciente(A(1..i)))}_{\beta} \wedge \underbrace{i \ne n}_{\gamma} \wedge \underbrace{ord}_{\delta}\}$$

$$\downarrow ?$$

$$\underbrace{(1 \le i \le n - 1)}_{\text{por } \alpha \text{ y } \gamma} \wedge \underbrace{(A(i) \le A(i + 1) \leftrightarrow creciente(A(1..i + 1)))}_{\text{por } \beta \text{ y } \delta}$$

Por α y γ deducimos que $1 \le i \le n - 1$. En la doble implicación de φ_3 se pregunta a ver si el hecho de que se cumpla creciente(A(1..i + 1)) depende de que se cumpla $A(i) \le A(i + 1)$. Y la respuesta es que sí porque por δ sabemos que ord vale *true* y como consecuense de ello y por β sabemos que cumple creciente(A(1..i)). Por tanto $A(i) \le A(i + 1)$ es la única comprobación que falta por hacer para ver si se cumple creciente(A(1..i + 1)).

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

$$\{\underbrace{(1 \le i \le n)}_{\alpha} \wedge \underbrace{(ord \leftrightarrow creciente(A(1..i)))}_{\beta} \wedge (\underbrace{i = n}_{\gamma} \vee \underbrace{\neg ord}_{\delta})\}$$

$$\downarrow\mathbf{?}$$

$$ord \leftrightarrow creciente(A(1..n))?$$

Como tenemos una disyunción hay que tener en cuenta las tres posibilidades de que $(i = n \vee \neg ord)$ sea True:

i = n	**¬ord**
True	True
True	False
False	True

- ✓ En los dos primeros casos al ser $i = n$, ocurre que β y ψ son iguales y por tanto se cumple la implicación.
- ✓ En el tercer caso tenemos $i \neq n$ y ord = False. De α deducimos que $i < n$ y de β deducimos que $\neg$creciente(A(1..i)). Es decir, creciente(A(1..i)) es False. Al ser creciente(A(1..i)) False e $i < n$, también creciente(A(1..n)) es False. Por tanto, en ψ nos queda False $\leftrightarrow$ False y ello supone que ψ es True.

Esto todo quiere decir que la implicación se cumple.

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

$$\{\underbrace{(1 \le i \le n)}_{\alpha} \wedge \underbrace{(ord \leftrightarrow creciente(A(1..i)))}_{\beta} \wedge \underbrace{i \ne n}_{\gamma} \wedge \underbrace{ord}_{\delta}\}$$

$$\downarrow ?$$

$$\underbrace{n - i > 0}_{\text{por } \alpha \text{ y } \gamma}$$

VI.

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

- $\{\varphi_5\} \equiv \{def(A(i) \le A(i + 1)) \wedge (\varphi_4)_{ord}^{A(i) \le A(i+1)}\} \equiv$
  $\equiv \{(1 \le i \le n) \wedge (1 \le i + 1 \le n) \wedge (n - i - 1 < v)\} \equiv$
  $\equiv \{(1 \le i \le n - 1) \wedge (n - i - 1 < v)\}$

- ¿$(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?

$$\{\underbrace{(1 \le i \le n)}_{\alpha} \wedge \underbrace{(ord \leftrightarrow creciente(A(1..i)))}_{\beta} \wedge \underbrace{i \ne n}_{\gamma} \wedge \underbrace{ord}_{\delta} \wedge \underbrace{(n - i = v)}_{\lambda}\}$$

$$\downarrow ?$$

$$\underbrace{(1 \le i \le n - 1)}_{\text{por } \alpha \text{ y } \gamma} \wedge \underbrace{(n - i - 1 < v)}_{\text{por } \lambda}$$

- **Demostración formal:**

1. $\varphi \to \varphi_1$
2. $\{\varphi_1\}$ ord : = true; {INV} (AA)
3. $\{\varphi\}$ ord : = true; {INV} (RCN 1, 2)

I

4. INV → INV

II

5. INV → def(B)

III

6. $(INV \wedge B) \to \varphi_3$
7. $\{\varphi_3\}$ ord : = (A(i) ≤ A(i + 1)); $\{\varphi_2\}$ (AA)
8. $\{INV \wedge B\}$ ord : = (A(i) ≤ A(i + 1)); $\{\varphi_2\}$ (RCN 6, 7)
9. $\{\varphi_2\}$ i : = i + 1; {INV} (AA)
10. $\{INV \wedge B\}$
 ord : = (A(i) ≤ A(i + 1));
 i : = i + 1;
 {INV} (RCP 8, 9)

IV

11. $(INV \wedge \neg B) \to \psi$

V

12. $(INV \wedge B) \to E > 0$

VI

13. $(INV \wedge B \wedge E = v) \to \varphi_5$
14. $\{\varphi_5\}$ ord : = (A(i) ≤ A(i + 1)); $\{\varphi_4\}$ (AA)
15. $\{INV \wedge B \wedge E = v\}$ ord : = (A(i) ≤ A(i + 1)); $\{\varphi_4\}$ (RCN 13, 14)
16. $\{\varphi_4\}$ i : = i + 1; {E < v} (AA)
17. $\{INV \wedge B \wedge E = v\}$
 ord : = (A(i) ≤ A(i + 1));
 i : = i + 1;
 {E < v} (RCP 15, 16)

18. {INV}
 while {INV} i ≠ n **and** ord **loop**
  ord : = (A(i) ≤ A(i + 1));
  i : = i + 1;
 end loop;
 $\{\psi\}$ (RWH 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
 ord : = true;
 while {INV} i ≠ n **and** ord **loop**
  ord : = (A(i) ≤ A(i + 1));
  i : = i + 1;
 end loop;
 $\{\psi\}$ (RCP 3, 18)

8. Programa que decide en la variable booleana res si se cumple que para cada valor par del vector A(1..n) en la correspondiente posición de R(1..n) se tiene un 0 y para cada valor impar del vector A(1..n) en la correspondiente posición de R(1..n) se tiene un 1. -- #

$\{\varphi\} \equiv \{n \geq 1 \land res\}$
$i := 0;$
while $\{INV\}$ $i \neq n$ **and** res **loop**
    res := (A(i + 1) **mod** 2 = R(i + 1));
    i := i + 1;
end loop;
$\{\psi\} \equiv \{res \leftrightarrow concuerda(A(1..n), R(1..n))\}$

$\{INV\} \equiv \{(0 \leq i \leq n) \land (res \leftrightarrow concuerda(A(1..i), R(1..i)))\}$

$E = n - i$

$concuerda(D(1..p), F(1..p)) \equiv \{\forall k(1 \leq k \leq p \rightarrow D(k)\ \underline{mod}\ 2 = F(k))\}$

Esquema:

- Como aparte del while hay una asignación previa del while, hay que distinguir dos subprogramas:

$\{\varphi\}$
$i := 0;$
$\{INV\}$

y

$\{INV\}$
while $\{INV\}$ $i \neq n$ **and** res **loop**
    res := (A(i + 1) **mod** 2 = R(i + 1));
    i := i + 1;
end loop;
$\{\psi\}$

➢ Primero se ha de verificar el primer subprograma

$\{\varphi\}$ ¿$(\varphi) \rightarrow \varphi_1$?
$\{\varphi_1\}$
$i := 0;$
(AA)
$\{INV\}$

- $\{\varphi_1\} \equiv \{def(0) \land (INV)_i^0\} \equiv$
  $\equiv \{true \land (0 \leq 0 \leq n) \land (res \leftrightarrow concuerda(A(1..0), R(1..0)))\} \equiv^{simplificación}$

$\equiv \{(0 \le n) \wedge (res \leftrightarrow concuerda(A(1..0), R(1..0)))\} \equiv^{\text{simplificación}}$
$\equiv \{(0 \le n) \wedge (res \leftrightarrow true\} \equiv^{\text{simplificación}}$
$\equiv \{(0 \le n) \wedge res\}$

La segunda simplificación se realiza porque en concuerda(A(1..0), R(1..0)) el intervalo es vacío y por tanto es true.
En el proceso de simplificación se ha tenido en cuenta que para cualquier fórmula δ se cumple por una parte que *true* $\wedge\ \delta \equiv \delta$ y por otra parte que *true* $\leftrightarrow \delta \equiv \delta$.

- ¿$\varphi \rightarrow \varphi_1$?

$$\{\underbrace{n \ge 1}_{\alpha} \wedge \underbrace{res}_{\beta}\}$$
$$\downarrow ?$$
$$\underbrace{(0 \le n)}_{\text{por } \alpha} \wedge \underbrace{res}_{\text{por } \beta}$$

➢ La aplicación de la Regla del While (RWH) al **<u>segundo subprograma</u>** supondrá realizar los siguientes cálculos y comprobaciones considerando que la precondición del while es {INV} y que la postcondición del while es $\{\psi\}$.

I. ¿INV $\rightarrow$ INV?
II. ¿INV $\rightarrow$ def(B)?

III. {INV $\wedge$ B} ¿(INV $\wedge$ B) $\rightarrow \varphi_3$?
$\{\varphi_3\}$
res : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1));
(AA)
$\{\varphi_2\}$
i : = i + 1;
(AA)
{INV}

IV. ¿(INV $\wedge \neg$B) $\rightarrow \psi$?
V. ¿(INV $\wedge$ B) $\rightarrow$ E > 0?
VI. {INV $\wedge$ B $\wedge$ E = v} ¿(INV $\wedge$ B $\wedge$ E = v) $\rightarrow \varphi_5$?
$\{\varphi_5\}$
res : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1));
(AA)
$\{\varphi_4\}$
i : = i + 1;
(AA)
{E < v}

I. ¿INV $\rightarrow$ INV? Sí por ser iguales.

II. ¿$INV \rightarrow def(B)$?

¿$INV \rightarrow true$? Sí, porque la segunda parte de la implicación es true.

III.

- $\{\varphi_2\} \equiv \{def(i + 1) \wedge (INV)_i^{i+1}\} \equiv$
  $\equiv \{(0 \le i + 1 \le n) \wedge (res \leftrightarrow concuerda(A(1..i + 1), R(1..i + 1)))\}$

- $\{\varphi_3\} \equiv \{def(A(i + 1)\ \underline{\mathbf{mod}}\ 2 = R(i + 1)) \wedge (\varphi_2)_{res}^{A(i+1)\ \underline{\mathbf{mod}}\ 2 = R(i+1)}\} \equiv$
  $\equiv \{(1 \le i + 1 \le n) \wedge (1 \le i + 1 \le n) \wedge (0 \le i + 1 \le n) \wedge$
  $(A(i + 1)\ \underline{\mathbf{mod}}\ 2 = R(i + 1) \leftrightarrow concuerda(A(1..i + 1), R(1..i + 1)))\} \equiv$
  $\equiv \{(0 \le i \le n - 1) \wedge$
  $(A(i + 1)\ \underline{\mathbf{mod}}\ 2 = R(i + 1) \leftrightarrow concuerda(A(1..i + 1), R(1..i + 1)))\}$

- ¿$(INV \wedge B) \rightarrow \varphi_3$?

$$\{\underbrace{(0 \le i \le n)}_{\alpha} \wedge \underbrace{(res \leftrightarrow concuerda(A(1..i), R(1..i)))}_{\beta} \wedge \underbrace{i \ne n}_{\gamma} \wedge \underbrace{res}_{\delta}\}$$

$$\downarrow ?$$

$$\underbrace{(0 \le i \le n - 1)}_{\text{por } \alpha \text{ y } \gamma} \wedge \underbrace{(A(i + 1)\ \underline{\mathbf{mod}}\ 2 = R(i + 1) \leftrightarrow concuerda(A(1..i + 1), R(1..i + 1)))}_{\text{por } \beta \text{ y } \delta}$$

Por β y δ deducimos que es cierto **concuerda(A(1..i), R(1..i)))**. En la doble implicación de φ_3 se pregunta a ver si el hecho de que se cumpla **concuerda(A(1..i + 1), R(1..i + 1))** depende de que se cumpla A(i + 1) <u>**mod**</u> 2 = R(i + 1). Y la respuesta es que sí porque sabemos que se cumple **concuerda(A(1..i), R(1..i)))** y por tanto A(i + 1) <u>**mod**</u> 2 = R(i + 1) es la única comprobación que falta por hacer para ver si se cumple **concuerda(A(1..i + 1), R(1..i + 1))**.

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

$$\{\underbrace{(0 \le i \le n)}_{\alpha} \wedge \underbrace{(res \leftrightarrow concuerda(A(1..i), R(1..i)))}_{\beta} \wedge (\underbrace{i = n}_{\gamma} \vee \underbrace{\neg res}_{\delta})\}$$

$$\downarrow ?$$

$$res \leftrightarrow concuerda(A(1..n), R(1..n))$$

Como tenemos una disyunción hay que tener en cuenta las tres posibilidades de que ($i = n \lor \neg res$) sea True:

i = n	**¬res**
True	True
True	False
False	True

- En los dos primeros casos al ser $i = n$, ocurre que β y ψ son iguales y por tanto se cumple la implicación.
- En el tercer caso tenemos $i \neq n$ y res = False. De $i \neq n$ y a deducimos que $i < n$ y de res = False y b deducimos que ¬concuerda(A(1..i), R(1..i)). Es decir, concuerda(A(1..i), R(1..i)) es False. Al ser concuerda(A(1..i), R(1..i)) False e $i < n$, también concuerda(A(1..n), R(1..n)) es False. Por tanto en ψ nos queda False ↔ False y de ahí tenemos que ψ es True.

Esto todo quiere decir que la implicación se cumple.

V. ¿$(INV \land B) \rightarrow E > 0$?

$$\{\underbrace{(0 \leq i \leq n)}_{\alpha} \land \underbrace{(res \leftrightarrow concuerda(A(1..i), R(1..i)))}_{\beta} \land \underbrace{i \neq n}_{\gamma} \land \underbrace{res}_{\delta}\}$$

$$\downarrow ?$$

$$n - i > 0$$

por α y γ

VI.

- $\{\varphi_4\} \equiv \{def(i + 1) \land (E < v)_i^{i+1}\} \equiv$
  $\equiv \{true \land n - (i + 1) < v\} \equiv \{n - i - 1 < v\}$

- $\{\varphi_5\} \equiv \{def(A(i + 1) \ \mathbf{\underline{mod}}\ 2 = R(i + 1)) \land (\varphi_4)_{res}^{A(i+1)\ \mathbf{\underline{mod}}\ 2 = R(i+1)}\} \equiv$
  $\equiv \{(1 \leq i + 1 \leq n) \land (1 \leq i + 1 \leq n) \land (n - i - 1 < v)\} \equiv$
  $\equiv \{(0 \leq i \leq n - 1) \land (n - i - 1 < v)\}$

- ¿$(INV \land B \land E = v) \rightarrow \varphi_5$?

$$\{\underbrace{(0 \leq i \leq n)}_{\alpha} \land \underbrace{(res \leftrightarrow concuerda(A(1..i), R(1..i)))}_{\beta} \land \underbrace{i \neq n}_{\gamma} \land \underbrace{res}_{\delta} \land \underbrace{(n - i = v)}_{\lambda}\}$$

$$\downarrow ?$$

$$\underbrace{(0 \leq i \leq n - 1)}_{\text{por } \alpha \text{ y } \gamma} \land \underbrace{(n - i - 1 < v)}_{\text{por } \lambda}$$

- **Demostración formal:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ i := 0; {INV} (AA)
3. $\{\varphi\}$ i := 0; {INV} (RCN 1, 2)

I

4. INV → INV

II

5. INV → def(B)

III

6. $(INV \wedge B) \rightarrow \varphi_3$
7. $\{\varphi_3\}$ res := (A(i + 1) **mod** 2 = R(i + 1)); $\{\varphi_2\}$ (AA)
8. $\{INV \wedge B\}$ res := (A(i + 1) **mod** 2 = R(i + 1)); $\{\varphi_2\}$ (RCN 6, 7)
9. $\{\varphi_2\}$ i := i + 1; {INV} (AA)
10. $\{INV \wedge B\}$
    res := (A(i + 1) **mod** 2 = R(i + 1));
    i := i + 1;
    {INV} (RCP 8, 9)

IV

11. $(INV \wedge \neg B) \rightarrow \psi$

V

12. $(INV \wedge B) \rightarrow E > 0$

VI

13. $(INV \wedge B \wedge E = v) \rightarrow \varphi_5$
14. $\{\varphi_5\}$ res := (A(i + 1) **mod** 2 = R(i + 1)); $\{\varphi_4\}$ (AA)
15. $\{INV \wedge B \wedge E = v\}$ res := (A(i + 1) **mod** 2 = R(i + 1)); $\{\varphi_4\}$
    (RCN 13, 14)
16. $\{\varphi_4\}$ i := i + 1; $\{E < v\}$ (AA)
17. $\{INV \wedge B \wedge E = v\}$
    res := (A(i + 1) **mod** 2 = R(i + 1));
    i := i + 1;
    $\{E < v\}$ (RCP 15, 16)

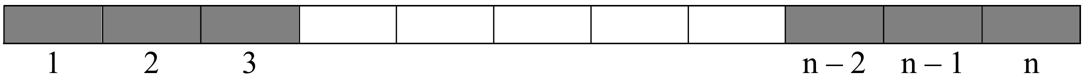
18. {INV}
    while {INV} i ≠ n **and** res **loop**
        res := (A(i + 1) **mod** 2 = R(i + 1));
        i := i + 1;
    end loop;
    $\{\psi\}$ (RWH 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
    i := 0;
    while {INV} i ≠ n **and** res **loop**
        res := (A(i + 1) **mod** 2 = R(i + 1));
        i := i + 1;
    end loop;
    $\{\psi\}$ (RCP 3, 18)

9. (Junio 2009) Programa que decide en la variable booleana sim si A(1..n) es simétrico. -- #

```
{φ} ≡ {n ≥ 1 ∧ i = 0}
sim : = true;
while {INV} i ≠ (n div 2) and sim loop
    i : = i + 1;
    sim : = (A(i) = A(n – i + 1));
end loop;
{ψ} ≡ {sim ↔ simetrico(A(1..n), n div 2)}
------------------------------------------------------------------------------

{INV} ≡ {(0 ≤ i ≤ n div 2) ∧ (sim ↔ simetrico(A(1..n), i))}

E = n div 2 – i

simetrico(A(1..n), pos) ≡ ∀k(1 ≤ k ≤ pos → A(k) = A(n – k + 1))

div es la división entera (Ejemplos: 30 div 2 = 15; 9 div 2 = 4)
```

Por ejemplo, simetrico(A(1..n), 3) quiere decir que la parte sombreada de la izquierda es simétrica con respecto a la parte sombreada de la derecha. El predicado no dice nada sobre la parte no sombreada.



Esquema:

- Como aparte del while hay una asignación previa del while, primero ponemos {INV} como precondición del while.

- Hay que distinguir dos subprogramas:

```
{φ}
sim : = true;
{INV}
```

y

```
{INV}
while {INV} i ≠ (n div 2) and sim loop
   i : = i + 1;
   sim : = (A(i) = A(n – i + 1));
end loop;
{ψ}
```

- Primero se ha de verificar el primer subprograma

$$\begin{array}{ll} \{\varphi\} & ¿(\varphi) \to \varphi_1? \\ & \{\varphi_1\} \\ & \text{sim} := \text{true}; \\ \text{(AA)} & \\ & \{\text{INV}\} \end{array}$$

- $\{\varphi_1\} \equiv \{\text{def(true)} \wedge (\text{INV})_{\text{sim}}^{\text{true}}\} \equiv$
  $\equiv \{\text{true} \wedge (0 \le i \le n \text{ div } 2) \wedge (\text{true} \leftrightarrow \text{simetrico}(A(1..n), i))\} \equiv^{\text{simplificación}}$
  $\equiv \{(0 \le i \le n \text{ div } 2) \wedge \text{simetrico}(A(1..n), i)\}$

La simplificación está basada en que para cualquier fórmula δ se cumple por una parte que *true* $\wedge\ \delta \equiv \delta$ y por otra parte que *true* $\leftrightarrow \delta \equiv \delta$.

- $¿\varphi \to \varphi_1?$

$$\{\underbrace{n \ge 1}_{\alpha} \wedge \underbrace{i = 0}_{\beta}\}$$

$$\downarrow ?$$

$$\underbrace{(0 \le}_{\text{por } \beta} \underbrace{i \le n \text{ div } 2)}_{\text{por } \alpha \text{ y } \beta} \wedge \underbrace{\text{simetrico}(A(1..n), i)}_{\text{por } \beta}$$

Hay que tener en cuenta que al ser $n \ge 1$, sabemos que se cumplirá $n \text{ div } 2 \ge 0$. Por otra parte, al ser i igual a 0, la formula universal correspondiente a simetrico(A(1..n), i) es $\forall k(1 \le k \le i \to A(k) = A(n - k + 1))$, es decir, $\forall k(1 \le k \le 0 \to A(k) = A(n - k + 1))$ y como ahí el dominio es vacío, la fórmula es cierta.

- La aplicación de la Regla del While (RWH) al **<u>segundo subprograma</u>** supondrá realizar los siguientes cálculos y comprobaciones:

  Aplicamos la Regla del While (RWH) considerando que la precondición del while es $\{INV\}$ y que la postcondición del while es $\{\psi\}$.

I.	$¿INV \rightarrow INV?$	
II.	$¿INV \rightarrow def(B)?$	
III.	$\{INV \wedge B\}$	$¿(INV \wedge B) \rightarrow \varphi_3?$
	$\{\varphi_3\}$	
	$i := i + 1;$	
(AA)		
	$\{\varphi_2\}$	
	$sim := (A(i) = A(n - i + 1));$	
(AA)		
	$\{INV\}$	
IV.	$¿(INV \wedge \neg B) \rightarrow \psi?$	
V.	$¿(INV \wedge B) \rightarrow E > 0?$	
VI.	$\{INV \wedge B \wedge E = v\}$	$¿(INV \wedge B \wedge E = v) \rightarrow \varphi_5?$
	$\{\varphi_5\}$	
	$i := i + 1;$	
(AA)		
	$\{\varphi_4\}$	
	$sim := (A(i) = A(n - i + 1));$	
(AA)		
	$\{E < v\}$	

I. $¿INV \rightarrow INV?$ Sí por ser iguales.

II. $¿INV \rightarrow def(B)?$
$¿INV \rightarrow def(i \neq (n \text{ div } 2)$ **and** $sim)?$
$¿INV \rightarrow true?$ Sí, porque la segunda parte de la implicación es true.

III.

- $\{\varphi_2\} \equiv \{def(A(i) = A(n - i + 1)) \wedge (INV)_{sim}^{(A(i) = A(n - i + 1))}\} \equiv$
  $\equiv \{(1 \leq i \leq n) \wedge (1 \leq n - i + 1 \leq n) \wedge (0 \leq i \leq n \text{ div } 2) \wedge$
  $((A(i) = A(n - i + 1)) \leftrightarrow simetrico(A(1..n), i))\} \equiv^{(simplificación)}$
  $\equiv \{(1 \leq i \leq n) \wedge (1 - n - 1 \leq n - i + 1 - n - 1 \leq n - n - 1) \wedge$
  $(0 \leq i \leq n \text{ div } 2) \wedge$
  $((A(i) = A(n - i + 1)) \leftrightarrow simetrico(A(1..n), i + 1))\} \equiv^{(simplificación)}$
  $\equiv \{(1 \leq i \leq n) \wedge (-n \leq -i \leq -1) \wedge (0 \leq i \leq n \text{ div } 2) \wedge$
  $((A(i) = A(n - i + 1)) \leftrightarrow simetrico(A(1..n), i + 1))\} \equiv^{(simplificación)}$
  $\equiv \{(1 \leq i \leq n) \wedge (1 \leq i \leq n) \wedge (0 \leq i \leq n \text{ div } 2) \wedge$
  $((A(i) = A(n - i + 1)) \leftrightarrow simetrico(A(1..n), i + 1))\} \equiv^{(simplificación)}$
  $\equiv \{(1 \leq i \leq (n \text{ div } 2)) \wedge$

$((A(i) = A(n - i + 1)) \leftrightarrow simetrico(A(1..n), i + 1))\}$

- $\{\varphi_3\} \equiv \{def(i + 1) \wedge (\varphi_2)_i^{i+1}\} \equiv$

  $\equiv \{true \wedge (1 \le i + 1 \le (n \text{ div } 2)) \wedge$

  $((A(i + 1) = A(n - (i + 1) + 1)) \leftrightarrow simetrico(A(1..n), i + 1))\} \equiv^{(\text{simplificación})}$

  $\equiv \{(0 \le i \le (n \text{ div } 2) - 1) \wedge ((A(i + 1) = A(n - i)) \leftrightarrow simetrico(A(1..n), i + 1))\}$

- ¿$(INV \wedge B) \rightarrow \varphi_3$?

$$\underbrace{(0 \le i \le n \text{ div } 2)}_{\alpha} \wedge \underbrace{(sim \leftrightarrow simetrico(A(1..n), i))}_{\beta} \wedge \underbrace{i \ne (n \text{ div } 2)}_{\gamma} \textbf{ and } \underbrace{sim}_{\delta}$$

$$\downarrow\mathbf{?}$$

$$\underbrace{(0 \le i \le (n \text{ div } 2) - 1)}_{\text{por } \alpha \text{ y } \gamma} \wedge \underbrace{(\underbrace{(A(i + 1) = A(n - i))}_{\lambda_1} \leftrightarrow \underbrace{simetrico(A(1..n), i + 1)}_{\lambda_2})}_{\text{por } \beta \text{ y } \delta}$$

Por α y γ deducimos que $0 \le i \le (n \text{ div } 2) - 1$.
En la doble implicación de φ_3 lo mejor es ver la doble implicación como las siguientes dos implicaciones:

- ¿$\lambda_1 \rightarrow \lambda_2$?
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?

Pasamos a desarrollar las dos preguntas:

- ¿$\lambda_1 \rightarrow \lambda_2$?
  por δ sabemos que sim vale *true* y como consecuencia de ello y por β sabemos que se cumple simetrico(A(1..n), i). Por tanto si λ_1 es true, es decir, si se cumple $(A(i + 1) = A(n - i))$, entonces se cumple simetrico(A(1..n), i + 1).
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?
  Si λ_1 es false, es decir, si se cumple $(A(i + 1) \ne A(n - i))$, entonces no se cumple simetrico(A(1..n), i + 1) porque justo el elemento de la posición i + 1 no es igual al elemento que ocupa la posición simétrica dentro de A(1..n).

En consecuencia $(INV \wedge B) \rightarrow \varphi_3$ se cumple.

IV. ¿(INV ∧ ¬B) → ψ?

$$\{\underbrace{(0 \le i \le n \text{ div } 2)}_{\alpha} \wedge \underbrace{(sim \leftrightarrow simetrico(A(1..n), i))}_{\beta} \wedge \underbrace{(i = (n \text{ div } 2)}_{\gamma} \vee \underbrace{\neg sim)}_{\delta}\}$$

$$\downarrow ?$$

$$sim \leftrightarrow simetrico(A(1..n), n \text{ div } 2)$$

Como tenemos una disyunción hay que tener en cuenta las tres posibilidades de que (i = (n div 2) ∨ ¬sim) sea True:

i = (n div 2)	**¬sim**
True	True
True	False
False	True

- ✓ En los dos primeros casos al ser **i = n div 2**, ocurre que β y ψ son iguales y por tanto se cumple la implicación.
- ✓ En el tercer caso tenemos **i ≠ (n div 2)** y **sim = False**. De α deducimos que i < (n div 2) y de β deducimos que ¬simetrico(A(1..n), i). Es decir, simetrico(A(1..n), i) es False. Al ser simetrico(A(1..n), i) False e i < (n div 2), también simetrico(A(1..n), n div 2) es False. Por tanto en ψ nos queda False ↔ False y ello supone que ψ es True.

Esto todo quiere decir que la implicación se cumple.

V. ¿(INV ∧ B) → E > 0?

$$\{(0 \le \underbrace{i \le n \text{ div } 2)}_{\alpha} \wedge (sim \leftrightarrow simetrico(A(1..n), i)) \wedge \underbrace{i \ne n \text{ div } 2}_{\beta} \wedge sim\}$$

$$\downarrow ?$$

$$\underbrace{n \text{ div } 2 - i > 0}_{\text{por } \alpha \text{ y } \beta}$$

Por α y β sabemos que i < n div 2. Si transformamos la expresión para que nos quede (n div 2) – i en la parte derecha, obtenemos la expresión i – i < (n div 2) – i, que simplificando queda 0 < (n div 2) – i, justo lo que queríamos probar.

VI.

- $\{\varphi_4\} \equiv \{def(A(i) = A(n-i+1)) \wedge (E < v)_{sim}{}^{(A(i) = A(n-i+1))}\} \equiv$
  $\equiv \{(1 \le i \le n) \wedge (1 \le n-i+1 \le n) \wedge (n \text{ div } 2 - i < v)\} \equiv^{(simplificación)}$
  $\equiv \{(1 \le i \le n) \wedge (1 - n - 1 \le n - i + 1 - n - 1 \le n - n - 1) \wedge$
  $(n \text{ div } 2 - i < v)\} \equiv^{(simplificación)}$
  $\equiv \{(1 \le i \le n) \wedge (-n \le -i \le -1) \wedge (n \text{ div } 2 - i < v)\} \equiv^{(simplificación)}$
  $\equiv \{(1 \le i \le n) \wedge (1 \le i \le n) \wedge (n \text{ div } 2 - i < v)\} \equiv^{(simplificación)}$
  $\equiv \{(1 \le i \le n) \wedge (n \text{ div } 2 - i < v)\}$

- $\{\varphi_5\} \equiv \{def(i+1) \wedge (\varphi_4)_i^{i+1}\} \equiv$
  $\equiv \{true \wedge (1 \le i+1 \le n) \wedge (n \text{ div } 2 - (i+1) < v)\} \equiv^{(simplificación)}$
  $\equiv \{(0 \le i \le n-1) \wedge (((n \text{ div } 2) - i - 1) < v)\}$

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

$$\underbrace{(0 \le i \le n \text{ div } 2)}_{\alpha} \wedge (sim \leftrightarrow simetrico(A(1..n), i)) \wedge \underbrace{i \ne (n \text{ div } 2)}_{\beta} \textbf{ and } sim \wedge \underbrace{(n \text{ div } 2) - i = v}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{(0 \le i \le n-1)}_{\text{por } \alpha \text{ y } \beta} \wedge \underbrace{(((n \text{ div } 2) - i - 1) < v)}_{\text{por } \delta}$$

Por α y β deducimos que $0 \le i \le (n \text{ div } 2) - 1$. Por α se sabe que n tiene que ser mayor o igual que 0 y para cualquier número n que sea ≥ 0 se cumple $n - 1 \ge (n \text{ div } 2)$. Por tanto, si $0 \le i \le (n \text{ div } 2) - 1$, entonces también se cumple $0 \le i \le n - 1$.
Por δ sabemos que $(n \text{ div } 2) - i = v$ y como consecuencia de ello $(n \text{ div } 2) - i - 1 = v - 1$. Por tanto, como $v - 1$ es menor que v, tenemos que $(((n \text{ div } 2) - i - 1) < v)$
En consecuencia $(INV \wedge B \wedge E < v) \to \varphi_5$ se cumple.

- **Demostración formal:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ sim : = true; {INV} (AA)
3. $\{\varphi\}$ sim : = true; {INV} (RCN 1, 2)

I

4. INV $\rightarrow$ INV

II

5. INV $\rightarrow$ def(B)

III

6. (INV $\wedge$ B) $\rightarrow \varphi_3$
7. $\{\varphi_3\}$ i : = i + 1; $\{\varphi_2\}$ (AA)
8. {INV $\wedge$ B} i : = i + 1; $\{\varphi_2\}$ (RCN 6, 7)
9. $\{\varphi_2\}$ sim : = (A(i) = A(n – i + 1)); {INV} (AA)
10. {INV $\wedge$ B}
    i : = i + 1;
    sim : = (A(i) = A(n – i + 1));
    {INV} (RCP 8, 9)

IV

11. (INV $\wedge$ ¬B) $\rightarrow \psi$

V

12. (INV $\wedge$ B) $\rightarrow$ E > 0

VI

13. (INV $\wedge$ B $\wedge$ E = v) $\rightarrow \varphi_5$
14. $\{\varphi_5\}$ i : = i + 1; $\{\varphi_4\}$ (AA)
15. {INV $\wedge$ B $\wedge$ E = v} i : = i + 1; $\{\varphi_4\}$
    (RCN 13, 14)
16. $\{\varphi_4\}$ sim : = (A(i) = A(n – i + 1)); {E < v} (AA)
17. {INV $\wedge$ B $\wedge$ E = v}
    i : = i + 1;
    sim : = (A(i) = A(n – i + 1));
    {E < v} (RCP 15, 16)

18. {INV}
    while {INV} i $\neq$ (n div 2) **and** sim **loop**
        i : = i + 1;
        sim : = (A(i) = A(n – i + 1));
    end loop;
    $\{\psi\}$ (RWH 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
    sim : = true;
    while {INV} i $\neq$ (n div 2) **and** sim **loop**
        i : = i + 1;
        sim : = (A(i) = A(n – i + 1));
    end loop;
    $\{\psi\}$ (RCP 3, 18)

10. (Septiembre 2009) Programa que decide en la variable booleana mult si A(1..n) es B(1..n) multiplicado por x. -- #

$\{\varphi\} \equiv \{n \geq 1 \wedge i = 1\}$
mult : = true;
while {INV} $i \neq (n + 1)$ **and** mult **loop**
    mult : = (A(i) = B(i) * x);
    i : = i + 1;
end loop;
$\{\psi\} \equiv \{mult \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k) * x)\}$

$\{INV\} \equiv \{(1 \leq i \leq n + 1) \wedge (mult \leftrightarrow \forall k(1 \leq k \leq i - 1 \rightarrow A(k) = B(k) * x))\}$

$E = n + 1 - i$

Esquema:

- Como aparte del while hay una asignación previa del while, primero ponemos {INV} como precondición del while.

- Hay que distinguir dos subprogramas:

$\{\varphi\}$
mult : = true;
{INV}

y

{INV}
while {INV} $i \neq (n + 1)$ **and** mult **loop**
    mult : = (A(i) = B(i) * x);
    i : = i + 1;
end loop;
$\{\psi\}$

- Primero se ha de verificar el primer subprograma

$$\begin{array}{lll} \{\varphi\} & & ¿(\varphi) \rightarrow \varphi_1? \\ & \{\varphi_1\} & \\ & \text{mult} := \text{true}; & \\ (\text{AA}) & & \\ & \{\text{INV}\} & \end{array}$$

- $\{\varphi_1\} \equiv \{\text{def}(\text{true}) \wedge (\text{INV})_{\text{mult}}^{\text{true}}\} \equiv$

$$\equiv \{\text{true} \wedge (1 \le i \le n + 1) \wedge (\text{true} \leftrightarrow \forall k(1 \le k \le i - 1 \rightarrow A(k) = B(k) * x))\} \equiv^{\text{simplificación}}$$
$$\equiv \{(1 \le i \le n + 1) \wedge \forall k(1 \le k \le i - 1 \rightarrow A(k) = B(k) * x)\}$$

La simplificación está basada en que para cualquier fórmula δ se cumple por una parte que *true* $\wedge\ \delta \equiv \delta$ y por otra parte que *true* $\leftrightarrow \delta \equiv \delta$.

- $¿\varphi \rightarrow \varphi_1?$

$$\{\underbrace{n \ge 1}_{\alpha} \wedge \underbrace{i = 1}_{\beta}\}$$
$$\downarrow ?$$
$$\underbrace{(1 \le i}_{\text{por } \beta} \underbrace{\le n + 1)}_{\text{por } \alpha \text{ y } \beta} \wedge \underbrace{\forall k(1 \le k \le i - 1 \rightarrow A(k) = B(k) * x)}_{\text{por } \beta}$$

Hay que tener en cuenta que al ser $n \ge 1$, sabemos que se cumplirá $n + 1 \ge 1$. Por otra parte, al ser i igual a 1, la formula universal $\forall k(1 \le k \le i - 1 \rightarrow A(k) = B(k) * x)$ viene a ser $\forall k(1 \le k \le 0 \rightarrow A(k) = B(k) * x)$ y como ahí el dominio es vacío, la fórmula es cierta.

- La aplicación de la Regla del While (RWH) al **<u>segundo subprograma</u>** supondrá realizar los siguientes cálculos y comprobaciones:

  Aplicamos la Regla del While (RWH) considerando que la precondición del while es $\{INV\}$ y que la poscondición del while es $\{\psi\}$.

I. ¿$INV \rightarrow INV$?
II. ¿$INV \rightarrow def(B)$?

III. $\{INV \wedge B\}$ ¿$(INV \wedge B) \rightarrow \varphi_3$?
$\{\varphi_3\}$
mult : = (A(i) = B(i) * x);
(AA)
$\{\varphi_2\}$
i : = i + 1;
(AA)
$\{INV\}$

IV. ¿$(INV \wedge \neg B) \rightarrow \psi$?
V. ¿$(INV \wedge B) \rightarrow E > 0$?
VI. $\{INV \wedge B \wedge E = v\}$ ¿$(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?
$\{\varphi_5\}$
mult : = (A(i) = B(i) * x);
(AA)
$\{\varphi_4\}$
i : = i + 1;
(AA)
$\{E < v\}$

I. ¿$INV \rightarrow INV$? Sí por ser iguales.

II. ¿$INV \rightarrow def(B)$?
¿$INV \rightarrow def(i \neq (n + 1)$ **and** $mult)$?
¿$INV \rightarrow true$? Sí, porque la segunda parte de la implicación es true.

III.

- $\{\varphi_2\} \equiv \{def(i + 1) \wedge (INV)_i^{\,i+1}\} \equiv$
  $\equiv \{true \wedge (1 \leq i + 1 \leq n + 1) \wedge$
  $(mult \leftrightarrow \forall k(1 \leq k \leq i + 1 - 1 \rightarrow A(k) = B(k) * x))\} \equiv^{(simplificación)}$
  $\equiv \{(0 \leq i \leq n) \wedge$
  $(mult \leftrightarrow \forall k(1 \leq k \leq i \rightarrow A(k) = B(k) * x))\}$

- $\{\varphi_3\} \equiv \{def(A(i) = B(i) * x) \wedge (\varphi_2)_{mult}^{A(i) = B(i) * x}\} \equiv$
  $\equiv \{(1 \leq i \leq n) \wedge (1 \leq i \leq n) \wedge (0 \leq i \leq n) \wedge$
  $((A(i) = B(i) * x) \leftrightarrow \forall k(1 \leq k \leq i \rightarrow A(k) = B(k) * x))\} \equiv^{(simplificación)}$
  $\equiv \{(1 \leq i \leq n) \wedge ((A(i) = B(i) * x) \leftrightarrow \forall k(1 \leq k \leq i \rightarrow A(k) = B(k) * x))\}$

- ¿(INV $\wedge$ B) $\rightarrow \varphi_3$?

$$\underbrace{(1 \le i \le n + 1)}_{\alpha} \wedge \underbrace{(mult \leftrightarrow \forall k(1 \le k \le i - 1 \rightarrow A(k) = B(k) * x))}_{\beta} \wedge \underbrace{i \ne (n + 1)}_{\gamma} \textbf{ and } \underbrace{mult}_{\delta}$$

$$\downarrow\textbf{?}$$

$$\underbrace{(1 \le i \le n)}_{\text{por } \alpha \text{ y } \gamma} \wedge \underbrace{(\underbrace{(A(i) = B(i) * x)}_{\lambda_1} \leftrightarrow \underbrace{\forall k(1 \le k \le i \rightarrow A(k) = B(k) * x)}_{\lambda_2})}_{\text{por } \beta \text{ y } \delta}$$

Por α y γ deducimos que $1 \le i \le n$.
En la doble implicación de φ_3 lo mejor es ver la doble implicación como las siguientes dos implicaciones:

- ¿$\lambda_1 \rightarrow \lambda_2$?
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?

Pasamos a desarrollar las dos preguntas:

- ¿$\lambda_1 \rightarrow \lambda_2$?
  por δ sabemos que mult vale *true* y como consecuencia de ello y por β sabemos que se cumple $\forall k(1 \le k \le i - 1 \rightarrow A(k) = B(k) * x)$. Por tanto si λ_1 es true, es decir, si se cumple $A(i) = B(i) * x$, entonces se cumple $\forall k(1 \le k \le i \rightarrow A(k) = B(k) * x)$.
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?
  Si λ_1 es false, es decir, si se cumple $A(i) \ne B(i) * x$, entonces no se cumple $\forall k(1 \le k \le i \rightarrow A(k) = B(k) * x)$ porque justo el elemento de la posición i de A no es igual al elemento que ocupa la misma posición en B multiplicado por x.

En consecuencia (INV $\wedge$ B) $\rightarrow \varphi_3$ se cumple.

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

$$\{\underbrace{(1 \leq i \leq n + 1)}_{\alpha} \wedge \underbrace{(mult \leftrightarrow \forall k(1 \leq k \leq i - 1 \rightarrow A(k) = B(k) * x))}_{\beta} \wedge (\underbrace{i = (n + 1)}_{\gamma} \vee \underbrace{\neg mult)}_{\delta}\}$$

$$\downarrow ?$$

$$mult \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k) * x)$$

Como tenemos una disyunción hay que tener en cuenta las tres posibilidades de que $(i = (n + 1) \vee \neg mult)$ sea True:

i = (n + 1)	**¬mult**
True	True
True	False
False	True

- ✓ En los dos primeros casos al ser **i = n + 1**, ocurre que β y ψ son iguales y por tanto se cumple la implicación.
- ✓ En el tercer caso tenemos **i ≠ (n + 1)** y **mult = False**. De α deducimos que $i < (n + 1)$ y de β deducimos que $\neg\forall k(1 \leq k \leq i - 1 \rightarrow A(k) = B(k) * x)$. Es decir, $\forall k(1 \leq k \leq i - 1 \rightarrow A(k) = B(k) * x)$ es False. Al ser $\forall k(1 \leq k \leq i - 1 \rightarrow A(k) = B(k) * x)$ False e $i < (n + 1)$, también $\forall k(1 \leq k \leq n \rightarrow A(k) = B(k) * x)$ es False porque si entre las posiciones 1 e $i - 1$ no todos los elementos cumplen la propiedad, entonces no todos los elementos del vector cumplen la propiedad. Por tanto en ψ nos queda False ↔ False y ello supone que ψ es True.

Esto todo quiere decir que la implicación se cumple.

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

$$\{(\underbrace{1 \leq i \leq n + 1)}_{\alpha} \wedge (mult \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k) * x)) \wedge \underbrace{i \neq n + 1}_{\beta} \wedge mult\}$$

$$\downarrow ?$$

$$\underbrace{n + 1 - i > 0}_{\text{por } \alpha \text{ y } \beta}$$

Por α y β sabemos que $i < n + 1$. Si transformamos la expresión para que nos quede $(n + 1) - i$ en la parte derecha, obtenemos la expresión $i - i < (n + 1) - i$, que simplificando queda $0 < (n + 1) - i$, justo lo que queríamos probar.

VI.

- $\{\varphi_4\} \equiv \{def(i + 1) \wedge (E < v)_i^{\,i+1}\} \equiv$
  $\equiv \{true \wedge (n + 1 - (i + 1) < v)\} \equiv^{(simplificación)}$
  $\equiv \{(n + 1 - i - 1 < v)\} \equiv^{(simplificación)}$
  $\equiv \{(n - i < v)\}$

- $\{\varphi_5\} \equiv \{def(A(i) = B(i) * x) \wedge (\varphi_4)_{mult}^{A(i) = B(i) * x}\} \equiv$
  $\equiv \{(1 \le i \le n) \wedge (1 \le i \le n) \wedge (n - i < v)\} \equiv^{(simplificación)}$
  $\equiv \{(1 \le i \le n) \wedge (n - i < v)\}$

- ¿$(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?

$$\underbrace{(1 \le i \le n + 1)}_{\alpha} \wedge (mult \leftrightarrow \forall k(1 \le k \le i - 1 \rightarrow A(k) = B(k) * x)) \wedge \underbrace{i \ne (n + 1)}_{\beta}\ \textbf{and}\ mult \wedge \underbrace{n + 1 - i = v}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{(1 \le i \le n)}_{\text{por } \alpha \text{ y } \beta} \wedge \underbrace{(n - i < v)}_{\text{por } \delta}$$

Por α y β deducimos que $1 \le i \le n$.
Por δ sabemos que $n + 1 - i = v$ y como consecuencia de ello $n + 1 - i - 1 = v - 1$, es decir, $n - i = v - 1$. Por tanto, como $v - 1$ es menor que v, tenemos que $n - i < v$
En consecuencia $(INV \wedge B \wedge E < v) \rightarrow \varphi_5$ se cumple.

- **Demostración formal:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ mult : = true; {INV} (AA)
3. $\{\varphi\}$ mult : = true; {INV} (RCN 1, 2)

I

4. INV $\rightarrow$ INV

II

5. INV $\rightarrow$ def(B)

III

6. (INV $\wedge$ B) $\rightarrow \varphi_3$
7. $\{\varphi_3\}$ mult : = (A(i) = B(i) * x); $\{\varphi_2\}$ (AA)
8. {INV $\wedge$ B} mult : = (A(i) = B(i) * x); $\{\varphi_2\}$ (RCN 6, 7)
9. $\{\varphi_2\}$ i : = i + 1; {INV} (AA)
10. {INV $\wedge$ B}
mult : = (A(i) = B(i) * x);
i : = i + 1;
{INV} (RCP 8, 9)

IV

11. (INV $\wedge$ ¬B) $\rightarrow \psi$

V

12. (INV $\wedge$ B) $\rightarrow$ E > 0

VI

13. (INV $\wedge$ B $\wedge$ E = v) $\rightarrow \varphi_5$
14. $\{\varphi_5\}$ mult : = (A(i) = B(i) * x); $\{\varphi_4\}$ (AA)
15. {INV $\wedge$ B $\wedge$ E = v} mult : = (A(i) = B(i) * x); $\{\varphi_4\}$
(RCN 13, 14)
16. $\{\varphi_4\}$ i : = i + 1; {E < v} (AA)
17. {INV $\wedge$ B $\wedge$ E = v}
mult : = (A(i) = B(i) * x);
i : = i + 1;
{E < v} (RCP 15, 16)

18. {INV}
while {INV} i ≠ (n + 1) **and** mult **loop**
mult : = (A(i) = B(i) * x);
i : = i + 1;
end loop;
$\{\psi\}$ (RWH 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
mult : = true;
while {INV} i ≠ (n + 1) **and** mult **loop**
mult : = (A(i) = B(i) * x);
i : = i + 1;
end loop;
$\{\psi\}$ (RCP 3, 18)

11. (Junio 2010) Programa que decide en la variable booleana multpos si cada elemento de A(1..n) es múltiplo de la posición que ocupa. -- #

$\{\varphi\} \equiv \{n \geq 1 \wedge i = 0\}$
multpos : = true;
<u>while</u> {INV} $i \neq n$ **and** multpos **<u>loop</u>**
  i : = i + 1;
  multpos : = ((A(i) **mod** i) = 0);
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{multpos \leftrightarrow \forall k(1 \leq k \leq n \rightarrow (A(k) \bmod k) = 0)\}$

$\{INV\} \equiv \{(0 \leq i \leq n) \wedge (multpos \leftrightarrow \forall k(1 \leq k \leq i \rightarrow (A(k) \bmod k) = 0))\}$

$E = n - i$

mod es el resto de la división entera
(Ejemplos: 10 mod 5 = 0; 10 mod 4 = 2; 10 mod 3 = 1)

<u>Esquema:</u>

- Como aparte del while hay una asignación previa del while, primero ponemos {INV} como precondición del while.

- Hay que distinguir dos subprogramas:

$\{\varphi\}$
multpos : = true;
{INV}

y

{INV}
<u>while</u> {INV} $i \neq n$ **<u>and</u>** multpos **<u>loop</u>**
  i : = i + 1;
  multpos : = ((A(i) mod i) = 0);
<u>end</u> **<u>loop</u>**;
$\{\psi\}$

➢ Primero se ha de verificar el primer subprograma

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

$\{\varphi_1\}$

multpos : = true;

(AA)

$\{INV\}$

- $\{\varphi_1\} \equiv \{def(true) \wedge (INV)_{multpos}^{true}\} \equiv$

  $\equiv \{true \wedge (0 \le i \le n) \wedge$

  $(true \leftrightarrow \forall k(1 \le k \le i \rightarrow (A(k) \bmod k) = 0))\} \equiv^{\text{simplificación}}$

  $\equiv \{(0 \le i \le n) \wedge \forall k(1 \le k \le i \rightarrow (A(k) \bmod k) = 0)\}$

La simplificación está basada en que para cualquier fórmula δ se cumple por una parte que *true* $\wedge\ \delta \equiv \delta$ y por otra parte que *true* $\leftrightarrow \delta \equiv \delta$.

- ¿$\varphi \rightarrow \varphi_1$?

$$\{\underbrace{n \ge 1}_{\alpha} \wedge \underbrace{i = 0}_{\beta}\}$$

$$\downarrow ?$$

$$(\underbrace{0 \le i}_{\text{por } \beta} \underbrace{\le n}_{\text{por } \alpha \text{ y } \beta}) \wedge \underbrace{\forall k(1 \le k \le i \rightarrow (A(k) \bmod k) = 0)}_{\text{por } \beta}$$

Hay que tener en cuenta que al ser $n \ge 1$ e $i = 0$, sabemos que se cumplirá $i \le n$. Por otra parte, al ser i igual a 0, en la formula universal $\forall k(1 \le k \le i \rightarrow (A(k) \bmod k) = 0)$ el dominio es vacío y por tanto la fórmula es cierta.

- La aplicación de la Regla del While (RWH) al **segundo subprograma** supondrá realizar los siguientes cálculos y comprobaciones:

  Aplicamos la Regla del While (RWH) considerando que la precondición del while es $\{INV\}$ y que la poscondición del while es $\{\psi\}$.

I. ¿$INV \rightarrow INV$?
II. ¿$INV \rightarrow def(B)$?

III. $\{INV \wedge B\}$ ¿$(INV \wedge B) \rightarrow \varphi_3$?
$\{\varphi_3\}$
$i := i + 1;$
(AA)
$\{\varphi_2\}$
multpos := ((A(i) **mod** i) = 0);
(AA)
$\{INV\}$

IV. ¿$(INV \wedge \neg B) \rightarrow \psi$?
V. ¿$(INV \wedge B) \rightarrow E > 0$?
VI. $\{INV \wedge B \wedge E = v\}$ ¿$(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?
$\{\varphi_5\}$
$i := i + 1;$
(AA)
$\{\varphi_4\}$
multpos := ((A(i) **mod** i) = 0);
(AA)
$\{E < v\}$

I. ¿$INV \rightarrow INV$? Sí por ser iguales.

II. ¿$INV \rightarrow def(B)$?
¿$INV \rightarrow def(i \neq n$ **and** $multpos)$?
¿$INV \rightarrow true$? Sí, porque la segunda parte de la implicación es true.

III.

- $\{\varphi_2\} \equiv \{def((A(i) \bmod i) = 0) \wedge (INV)_{multpos}^{((A(i) \bmod i) = 0)}\} \equiv$
  $\equiv \{(1 \leq i \leq n) \wedge i \neq 0 \wedge (0 \leq i \leq n) \wedge$
  $((A(i)\ \mathbf{mod}\ i) = 0) \leftrightarrow \forall k(1 \leq k \leq i \rightarrow (A(k) \bmod k) = 0) \equiv^{(simplificación)}$
  $\equiv \{(1 \leq i \leq n) \wedge$
  $((A(i)\ \mathbf{mod}\ i) = 0) \leftrightarrow \forall k(1 \leq k \leq i \rightarrow (A(k) \bmod k) = 0)\}$

- $\{\varphi_3\} \equiv \{def(i + 1) \wedge (\varphi_2)_i^{i+1}\} \equiv$
  $\equiv \{true \wedge (1 \leq i + 1 \leq n) \wedge$
  $((A(i + 1)\ \mathbf{mod}\ (i + 1)) = 0) \leftrightarrow \forall k(1 \leq k \leq i + 1 \rightarrow (A(k) \bmod k) = 0)\} \equiv^{(simplificación)}$
  $\equiv \{(0 \leq i \leq n - 1) \wedge$
  $((A(i + 1)\ \mathbf{mod}\ (i + 1)) = 0) \leftrightarrow \forall k(1 \leq k \leq i + 1 \rightarrow (A(k) \bmod k) = 0)\}$

- ¿(INV $\wedge$ B) $\rightarrow \varphi_3$?

$$\underbrace{(0 \le i \le n)}_{\alpha} \wedge \underbrace{(\underbrace{\text{multpos}}_{\beta_1} \leftrightarrow \underbrace{\forall k(1 \le k \le i \rightarrow (A(k) \bmod k) = 0))}_{\beta_2}}_{\beta} \wedge \underbrace{i \ne n}_{\gamma} \textbf{ and } \underbrace{\text{multpos}}_{\delta}$$

$$\downarrow\textbf{?}$$

$$\underbrace{(0 \le i \le n-1)}_{\text{por } \alpha \text{ y } \gamma} \wedge \underbrace{\underbrace{((A(i+1) \textbf{ mod } (i+1)) = 0)}_{\lambda_1} \leftrightarrow \underbrace{\forall k(1 \le k \le i+1 \rightarrow (A(k) \bmod k) = 0)}_{\lambda_2}}_{\text{por } \beta \text{ y } \delta}$$

Por α y γ deducimos que $0 \le i \le n - 1$.
En la doble implicación de φ_3 lo mejor es ver la doble implicación como las siguientes dos implicaciones:

- ¿$\lambda_1 \rightarrow \lambda_2$?
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?

Pasamos a desarrollar las dos preguntas:

- ¿$\lambda_1 \rightarrow \lambda_2$?
  por δ sabemos que multpos vale *true* y como consecuencia de ello y por ser β true, sabemos que β_2 es true, es decir, hasta la posición i (incluida) los elementos de A son múltiplos de la posición que ocupan. Por tanto si λ_1 es true, es decir, si se cumple ((A(i + 1) **mod** (i + 1)) = 0), entonces se cumple que hasta la posición i + 1 (incluida) los elementos de A son múltiplos de la posición que ocupan. En resumen, λ_2 es true porque β_2 y λ_1 son true.
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?
  Si λ_1 es false, es decir, si no se cumple ((A(i + 1) **mod** (i + 1)) = 0), entonces no es verdad que hasta la posición i + 1 (incluida) los elementos de A son múltiplos de la posición que ocupan y por tanto λ_2 es false. Por ser β_2 true, hasta la posición i (incluida) los elementos de A son múltiplos de la posición que ocupan pero por considerar ahora que λ_1 es false, justo el elemento de la posición i + 1 no es múltiplo de la posición que ocupa y es por ello que λ_2 es false. En resumen, λ_2 es false porque λ_1 es false.

En consecuencia (INV $\wedge$ B) $\rightarrow \varphi_3$ se cumple.

IV. ¿(INV ∧ ¬B) → ψ?

$$\underbrace{(0 \le i \le n)}_{\alpha} \wedge \underbrace{(\underbrace{multpos}_{\beta_1} \leftrightarrow \underbrace{\forall k(1 \le k \le i \to (A(k) \bmod k) = 0)}_{\beta_2})}_{\beta} \wedge (\underbrace{i = n}_{\gamma} \vee \underbrace{\neg multpos}_{\delta})$$

$$\downarrow ?$$

$$\underbrace{multpos \leftrightarrow \forall k(1 \le k \le n \to (A(k) \bmod k) = 0)}_{\psi}$$

Como tenemos una disyunción hay que tener en cuenta las tres posibilidades de que (i = n ∨ ¬multpos) sea True:

i = n	**¬multpos**
True	True
True	False
False	True

- ✓ En los dos primeros casos al ser **i = n**, ocurre que β y ψ son iguales y por tanto se cumple la implicación.
- ✓ En el tercer caso tenemos **i ≠ n** y **multpos = False**. Como i es distinto de n, de α deducimos que i es menor que n. Para que ψ sea cierto, la fórmula ∀k(1 ≤ k ≤ n → (A(k) mod k) = 0) debería ser false. Como β es true y multpos es false, deducimos que β_2 es false y por tanto no es verdad que los primeros i elementos de A sean múltiplos de la posición que ocupan. Y si en el tramo 1..i no todos son múltiplos de la posición que ocupan, entonces no es verdad que todos los elementos del vector sean múltiplos de la posición que ocupan, es decir, ∀k(1 ≤ k ≤ n → (A(k) mod k) = 0) es false y como consecuencia ψ es true.

Esto todo quiere decir que la implicación se cumple.

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

$$\underbrace{(0 \leq i \leq n)}_{\alpha} \wedge (multpos \leftrightarrow \forall k(1 \leq k \leq i \rightarrow (A(k) \bmod k) = 0)) \wedge \underbrace{i \neq n}_{\beta} \wedge multpos$$

$$\downarrow ?$$

$$\underbrace{n - i > 0}_{\text{por } \alpha \text{ y } \beta}$$

Por α y β sabemos que $i < n$. Si transformamos la expresión para que nos quede $n - i$ en la parte derecha, obtenemos la expresión $i - i < n - i$, que simplificando queda $0 < n - i$, justo lo que queríamos probar.

VI.

- $\{\varphi_4\} \equiv \{def(((A(i) \textbf{ mod } i) = 0)) \wedge (E < v)_{multpos}^{((A(i) \textbf{ mod } i) = 0)}\} \equiv$
  $\equiv \{(1 \leq i \leq n) \wedge i \neq 0 \wedge (n - i < v)\} \equiv^{(simplificación)}$
  $\equiv \{(1 \leq i \leq n) \wedge (n - i < v)\}$

- $\{\varphi_5\} \equiv \{def(i + 1) \wedge (\varphi_4)_i^{i+1}\} \equiv$
  $\equiv \{true \wedge (1 \leq i + 1 \leq n) \wedge (n - (i + 1) < v)\} \equiv^{(simplificación)}$
  $\equiv \{(0 \leq i \leq n - 1) \wedge (n - i - 1 < v)\}$

- ¿$(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?

$$\underbrace{(0 \leq i \leq n)}_{\alpha} \wedge (multpos \leftrightarrow \forall k(1 \leq k \leq i \rightarrow (A(k) \bmod k) = 0)) \wedge \underbrace{i \neq n}_{\beta} \wedge multpos \wedge \underbrace{(n - i = v)}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{(0 \leq i \leq n - 1)}_{\text{por } \alpha \text{ y } \beta} \wedge \underbrace{(n - i - 1 < v)}_{\text{por } \delta}$$

Por α y β deducimos que $0 \leq i \leq n - 1$.

Por δ sabemos que $n - i = v$ y como consecuencia de ello $n - i - 1 = v - 1$. Por tanto, como $v - 1$ es menor que v, tenemos que $(n - i - 1) < v$.

En consecuencia $(INV \wedge B \wedge E < v) \rightarrow \varphi_5$ se cumple.

- **Demostración formal:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ multpos : = true; {INV} (AA)
3. $\{\varphi\}$ multpos : = true; {INV} (RCN 1, 2)

I

4. INV → INV

II

5. INV → def(B)

III

6. $(INV \wedge B) \rightarrow \varphi_3$
7. $\{\varphi_3\}$ i : = i + 1; $\{\varphi_2\}$ (AA)
8. $\{INV \wedge B\}$ i : = i + 1; $\{\varphi_2\}$ (RCN 6, 7)
9. $\{\varphi_2\}$ multpos : = ((A(i) **mod** i) = 0); {INV} (AA)
10. $\{INV \wedge B\}$
   i : = i + 1;
   multpos : = ((A(i) **mod** i) = 0);
   {INV} (RCP 8, 9)

IV

11. $(INV \wedge \neg B) \rightarrow \psi$

V

12. $(INV \wedge B) \rightarrow E > 0$

VI

13. $(INV \wedge B \wedge E = v) \rightarrow \varphi_5$
14. $\{\varphi_5\}$ i : = i + 1; $\{\varphi_4\}$ (AA)
15. $\{INV \wedge B \wedge E = v\}$ i : = i + 1; $\{\varphi_4\}$ (RCN 13, 14)
16. $\{\varphi_4\}$ multpos : = (A(i) = A(n – i + 1)); $\{E < v\}$ (AA)
17. $\{INV \wedge B \wedge E = v\}$
   i : = i + 1;
   multpos : = ((A(i) **mod** i) = 0);
   $\{E < v\}$ (RCP 15, 16)

18. {INV}
   while {INV} i ≠ n **and** multpos **loop**
      i : = i + 1;
      multpos : = ((A(i) **mod** i) = 0);
   end loop;
   $\{\psi\}$ (RWH 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
   multpos : = true;
   while {INV} i ≠ n **and** multpos **loop**
      i : = i + 1;
      multpos : = ((A(i) **mod** i) = 0);
   end loop;
   $\{\psi\}$ (RCP 3, 18)

12. (Septiembre 2010) Programa que decide en la variable booleana mult si algún elemento de A(1..n) es múltiplo de x. -- #

$\{\varphi\} \equiv \{n \geq 1 \land i = 1 \land x \neq 0\}$
mult : = false;
<u>while</u> {INV} $i \neq n + 1$ **and not** mult **<u>loop</u>**
    i : = i + 1;
    mult : = (A(i – 1) **mod** x = 0);
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{mult \leftrightarrow \exists k(1 \leq k \leq n \land (A(k) \bmod k) = 0)\}$

$\{INV\} \equiv \{x \neq 0 \land (1 \leq i \leq n + 1) \land (mult \leftrightarrow \exists k(1 \leq k \leq i - 1 \land A(k) \bmod x = 0))\}$

$E = n + 1 - i$

mod es el resto de la división entera
(Ejemplos: 10 mod 5 = 0; 10 mod 4 = 2; 10 mod 3 = 1)

<u>Esquema:</u>

- Como aparte del while hay una asignación previa del while, primero ponemos {INV} como precondición del while.

- Hay que distinguir dos subprogramas:

$\{\varphi\}$
mult : = false;
{INV}

y

{INV}
<u>while</u> {INV} $i \neq n + 1$ **<u>and</u> not** mult **<u>loop</u>**
    i : = i + 1;
    mult : = (A(i – 1) **mod** x) = 0);
<u>end</u> **<u>loop</u>**;
$\{\psi\}$

➢ Primero se ha de verificar el primer subprograma

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

$\{\varphi_1\}$

mult : = false;

(AA)

$\{INV\}$

- $\{\varphi_1\} \equiv \{def(false) \wedge (INV)_{mult}^{false}\} \equiv$

  $\equiv \{true \wedge x \neq 0 \wedge (1 \leq i \leq n + 1) \wedge$

  $(false \leftrightarrow \exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0))\} \equiv^{\text{simplificación}}$

  $\equiv \{x \neq 0 \wedge (1 \leq i \leq n + 1) \wedge \neg\exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0)\}$

La simplificación está basada en que para cualquier fórmula δ se cumple por una parte que *true* $\wedge\ \delta \equiv \delta$ y por otra parte que *false* $\leftrightarrow \delta \equiv \neg\delta$.

- ¿$\varphi \rightarrow \varphi_1$?

$$\{\underbrace{n \geq 1}_{\alpha} \wedge \underbrace{i = 1}_{\beta} \wedge \underbrace{x \neq 0}_{\gamma}\}$$

$$\downarrow \mathbf{?}$$

$$\underbrace{x \neq 0}_{\text{por } \gamma} \wedge (\underbrace{1 \leq i}_{\text{por } \beta} \underbrace{\leq n + 1}_{\text{por } \alpha \text{ y } \beta}) \wedge \underbrace{\neg\exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0)}_{\text{por } \beta}$$

Hay que tener en cuenta que al ser $i = 1$, sabemos que se cumplirá $1 \leq i$. Por ser $n \geq 1$ e $i = 1$, sabemos que se cumple $i \leq n + 1$. Por otra parte, al ser i igual a 1, en la formula existencial $\exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0)$ el dominio es vacío y por tanto la fórmula es falsa y por tanto $\neg\exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0)$ es cierta (true).

- La aplicación de la Regla del While (RWH) al **segundo subprograma** supondrá realizar los siguientes cálculos y comprobaciones:

  Aplicamos la Regla del While (RWH) considerando que la precondición del while es $\{INV\}$ y que la poscondición del while es $\{\psi\}$.

I. ¿$INV \rightarrow INV$?
II. ¿$INV \rightarrow def(B)$?

III. $\{INV \wedge B\}$ ¿$(INV \wedge B) \rightarrow \varphi_3$?
$\{\varphi_3\}$
$i := i + 1;$
(AA)
$\{\varphi_2\}$
$mult := (A(i-1)$ **mod** $x = 0);$
(AA)
$\{INV\}$

IV. ¿$(INV \wedge \neg B) \rightarrow \psi$?
V. ¿$(INV \wedge B) \rightarrow E > 0$?
VI. $\{INV \wedge B \wedge E = v\}$ ¿$(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?
$\{\varphi_5\}$
$i := i + 1;$
(AA)
$\{\varphi_4\}$
$mult := (A(i-1)$ **mod** $x = 0);$
(AA)
$\{E < v\}$

I. ¿$INV \rightarrow INV$? Sí por ser iguales.

II. ¿$INV \rightarrow def(B)$?
¿$INV \rightarrow def(i \neq n + 1$ **and not** $mult)$?
¿$INV \rightarrow true$? Sí, porque la segunda parte de la implicación es true.

III.

- $\{\varphi_2\} \equiv \{def(A(i-1) \bmod x = 0) \wedge (INV)_{mult}^{(A(i-1) \bmod x = 0)}\} \equiv$
  $\equiv \{(1 \leq i - 1 \leq n) \wedge x \neq 0 \wedge x \neq 0 \wedge (1 \leq i \leq n + 1) \wedge$
  $(A(i-1)\ \mathbf{mod}\ x = 0) \leftrightarrow \exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0))\} \equiv^{\text{simplificación}}$
  $\equiv \{(2 \leq i \leq n + 1) \wedge x \neq 0 \wedge (1 \leq i \leq n + 1) \wedge$
  $(A(i-1)\ \mathbf{mod}\ x = 0) \leftrightarrow \exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0))\} \equiv^{\text{simplificación}}$
  $\equiv \{(2 \leq i \leq n + 1) \wedge x \neq 0 \wedge$
  $(A(i-1)\ \mathbf{mod}\ x = 0) \leftrightarrow \exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0))\}$

- $\{\varphi_3\} \equiv \{def(i+1) \wedge (\varphi_2)_i^{i+1}\} \equiv$
  $\equiv \{true \wedge (2 \leq i+1 \leq n+1) \wedge x \neq 0 \wedge$
  $(A(i)\ \mathbf{mod}\ x = 0) \leftrightarrow \exists k(1 \leq k \leq i \wedge A(k) \bmod x = 0))\} \equiv^{\text{simplificación}}$
  $\equiv \{(2-1 \leq i+1-1 \leq n+1-1) \wedge x \neq 0 \wedge$
  $(A(i)\ \mathbf{mod}\ x = 0) \leftrightarrow \exists k(1 \leq k \leq i \wedge A(k) \bmod x = 0))\} \equiv^{\text{simplificación}}$
  $\equiv \{(1 \leq i \leq n) \wedge x \neq 0 \wedge$
  $(A(i)\ \mathbf{mod}\ x = 0) \leftrightarrow \exists k(1 \leq k \leq i \wedge A(k) \bmod x = 0))\}$

- ¿$(INV \wedge B) \rightarrow \varphi_3$?

- ¿$(INV \wedge B) \rightarrow \varphi_3$?

$$\underbrace{x \neq 0}_{\pi} \wedge \underbrace{(1 \leq i \leq n+1)}_{\alpha} \wedge \underbrace{(\underbrace{mult}_{\beta_1} \leftrightarrow \underbrace{\exists k(1 \leq k \leq i-1 \wedge A(k) \bmod x = 0))}_{\beta_2}}_{\beta} \wedge \underbrace{i \neq n+1}_{\gamma}\ \mathbf{and}\ \underbrace{\mathbf{not}\ mult}_{\delta}$$

$$\downarrow \mathbf{?}$$

$$\underbrace{(1 \leq i \leq n)}_{\text{por } \alpha \text{ y } \gamma} \wedge \underbrace{x \neq 0}_{\text{por } \pi} \wedge \underbrace{\underbrace{(A(i)\ \mathbf{mod}\ x = 0)}_{\lambda_1} \leftrightarrow \underbrace{\exists k(1 \leq k \leq i \wedge A(k) \bmod x = 0))}_{\lambda_2}}_{\lambda \text{ (por } \beta \text{ y } \delta)}$$

Por α y γ deducimos que $1 \leq i \leq n$.

$x \neq 0$ se cumple porque también aparece arriba (π).

En la doble implicación de φ_3 lo mejor es ver la doble implicación como las siguientes dos implicaciones:

- ¿$\lambda_1 \rightarrow \lambda_2$?
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?

Pasamos a desarrollar las dos preguntas:

- ¿$\lambda_1 \rightarrow \lambda_2$?
  por δ sabemos que *mult* vale *false* y como consecuencia de ello y por ser β true, sabemos que β_2 es false, es decir, hasta la posición i – 1 (incluida) ningún elemento de A es múltiplo de x. Por tanto, si λ_1 es true, es decir, si se cumple (A(i) **mod** x = 0), entonces se cumple que hasta la posición i (incluida) sí existe algún elemento de A que es múltiplo de x, justo el de la posición i. En resumen, λ_2 es true porque λ_1 es true.
- ¿$(\neg\lambda_1) \rightarrow (\neg\lambda_2)$?

Si λ_1 es false, es decir, si no se cumple (A(i) **mod** x = 0), entonces no es verdad que hasta la posición i (incluida) exista algún elemento de A que sea múltiplo de x y por tanto λ_2 es false. Por ser β_2 false, hasta la posición i – 1 (incluida) ningún elemento de A es múltiplo de x y por considerar ahora que λ_1 es false, el elemento de la posición i tampoco es múltiplo de la posición que ocupa y es por ello que λ_2 es false. En resumen, λ_2 es false porque λ_1 y β_2 son false.

En consecuencia $(INV \wedge B) \rightarrow \varphi_3$ se cumple.

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

$$\underbrace{x \neq 0}_{\pi} \wedge \underbrace{(1 \le i \le n + 1)}_{\alpha} \wedge \underbrace{(\underbrace{mult}_{\beta_1} \leftrightarrow \underbrace{\exists k(1 \le k \le i - 1 \wedge A(k) \bmod x = 0)}_{\beta_2})}_{\beta} \wedge (\underbrace{i = n + 1}_{\gamma} \vee \underbrace{mult}_{\delta})$$

$$\downarrow ?$$

$$\underbrace{mult \leftrightarrow \exists k(1 \le k \le n \wedge A(k) \bmod x = 0)}_{\psi}$$

Como tenemos una disyunción hay que tener en cuenta las tres posibilidades de que $(i = n + 1 \vee mult)$ sea True:

i = n + 1	**mult**
True	True
True	False
False	True

- ✓ En los dos primeros casos al ser **i = n + 1**, ocurre que β y ψ son iguales y por tanto se cumple la implicación.
- ✓ En el tercer caso tenemos **i ≠ n + 1** y **mult = True**. Como i es distinto de n + 1, de α deducimos que i es menor que n + 1. Para que ψ sea cierto, la fórmula $\exists k(1 \le k \le n \wedge A(k) \bmod x = 0)$ debería ser true. Como β es true y mult es true, deducimos que β_2 es true y por tanto sabemos que en el intervalo 1..i – 1 de A algún elemento es múltiplo de x. Y si en el tramo 1..i – 1 algún elemento de A es múltiplo de x, entonces podemos asegurar que en el tramo 1..n algún elemento de A es múltiplo de x, es decir, $\exists k(1 \le k \le n \wedge A(k) \bmod x = 0)$ es true y como consecuencia ψ es true porque las dos parte de la doble implicación son true.

Esto todo quiere decir que la doble implicación se cumple.

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

$$x \neq 0 \wedge \underbrace{(1 \leq i \leq n+1)}_{\alpha} \wedge (mult \leftrightarrow \exists k(1 \leq k \leq i-1 \wedge A(k) \bmod x = 0)) \wedge \underbrace{i \neq n+1}_{\beta} \textbf{ and not } mult$$

$$\downarrow ?$$

$$\underbrace{n+1-i > 0}_{\text{por } \alpha \text{ y } \beta}$$

Por α y β sabemos que $i < n + 1$. Es decir, tenemos $n + 1 > i$. Si transformamos la expresión para que nos quede $n + 1 - i$ en la parte izquierda, obtenemos la expresión $n + 1 - i > i - i$, que simplificando queda $n + 1 - i > 0$, justo lo que queríamos probar.

VI.

- $\{\varphi_4\} \equiv \{def(A(i-1) \textbf{ mod } x = 0) \wedge (E < v)_{mult}^{A(i) \textbf{ mod } x = 0}\} \equiv$
  $\equiv \{(1 \leq i-1 \leq n) \wedge x \neq 0 \wedge (n+1-i < v)\} \equiv^{(\text{simplificación})}$
  $\equiv \{(2 \leq i \leq n+1) \wedge x \neq 0 \wedge (n+1-i < v)\}$

- $\{\varphi_5\} \equiv \{def(i+1) \wedge (\varphi_4)_i^{i+1}\} \equiv$
  $\equiv \{true \wedge (2 \leq i+1 \leq n+1) \wedge x \neq 0 \wedge (n+1-(i+1) < v)\} \equiv^{(\text{simplificación})}$
  $\equiv \{(2-1 \leq i+1-1 \leq n+1-1) \wedge x \neq 0 \wedge (n-i < v)\} \equiv^{(\text{simplificación})}$
  $\equiv \{(1 \leq i \leq n) \wedge x \neq 0 \wedge (n-i < v)\}$

- $(INV \wedge B \wedge E = v) \rightarrow \varphi_5$?

$$\underbrace{x \neq 0}_{\gamma} \wedge \underbrace{(1 \leq i \leq n+1)}_{\alpha} \wedge (mult \leftrightarrow \exists k(1 \leq k \leq i-1 \wedge A(k) \bmod x = 0)) \wedge \underbrace{i \neq n+1}_{\beta} \wedge \neg mult \wedge \underbrace{(n+1-i = v)}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{(1 \leq i \leq n)}_{\text{por } \alpha \text{ y } \beta} \wedge \underbrace{x \neq 0}_{\text{por } \gamma} \wedge \underbrace{(n-i < v)}_{\text{por } \delta}$$

Por α y β deducimos que $1 \leq i \leq n$.

$x \neq 0$ se cumple porque aparece también arriba (γ).

Por δ sabemos que $n + 1 - i = v$ y como consecuencia de ello $n + 1 - i - 1 = v - 1$. Por tanto, $n - i = v - 1$ y como $v - 1$ es menor que v, tenemos que $n - i < v$.

En consecuencia $(INV \wedge B \wedge E < v) \rightarrow \varphi_5$ se cumple.

- **Demostración formal:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ mult : = false; {INV} (AA)
3. $\{\varphi\}$ mult : = false; {INV} (RCN 1, 2)

I

4. INV → INV

II

5. INV → def(B)

III

6. $(INV \wedge B) \rightarrow \varphi_3$
7. $\{\varphi_3\}$ i : = i + 1; $\{\varphi_2\}$ (AA)
8. $\{INV \wedge B\}$ i : = i + 1; $\{\varphi_2\}$ (RCN 6, 7)
9. $\{\varphi_2\}$ mult : = (A(i – 1) **mod** x = 0); {INV} (AA)
10. $\{INV \wedge B\}$
    i : = i + 1;
    mult : = (A(i – 1) **mod** x = 0);
    {INV} (RCP 8, 9)

IV

11. $(INV \wedge \neg B) \rightarrow \psi$

V

12. $(INV \wedge B) \rightarrow E > 0$

VI

13. $(INV \wedge B \wedge E = v) \rightarrow \varphi_5$
14. $\{\varphi_5\}$ i : = i + 1; $\{\varphi_4\}$ (AA)
15. $\{INV \wedge B \wedge E = v\}$ i : = i + 1; $\{\varphi_4\}$ (RCN 13, 14)
16. $\{\varphi_4\}$ mult : = (A(i – 1) **mod** x = 0); $\{E < v\}$ (AA)
17. $\{INV \wedge B \wedge E = v\}$
    i : = i + 1;
    mult : = (A(i – 1) **mod** x = 0);
    $\{E < v\}$ (RCP 15, 16)

18. {INV}
    while {INV} $i \neq n + 1$ **and not** mult **loop**
        i : = i + 1;
        mult : = (A(i – 1) **mod** x = 0);
    end loop;
    $\{\psi\}$ (RWH 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
    mult : = false;
    while {INV} $i \neq n + 1$ **and not** mult **loop**
        i : = i + 1;
        mult : = (A(i – 1) **mod** x = 0);
    end loop;
    $\{\psi\}$ (RCP 3, 18)