3. GAIKO ARIKETAK
PROGRAMEN EGIAZTAPENA

AURKIBIDEA

a) Esleipena eta konposizio sekuentziala [#]

1. Esleipena (s : = s + A(i);)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{1 \le i \le n \wedge s = \sum_{k=1}^{i-1} A(k)\}$$

$$s := s + A(i);$$

$$\{\psi\} \equiv \{1 \le i \le n \wedge s = \sum_{k=1}^{i} A(k)\}$$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

2. Esleipena (k : = k div 2;)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{bikoitia(k) \wedge y * z^k = c\}$$

$$k := k \text{ div } 2;$$

$$\{\psi\} \equiv \{y * z^{2 * k} = c\}$$

div zatiketa osoa da (Adibideak: 30 div 2 = 15; 9 div 2 = 4).

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

3. Esleipena (zerorik_ez : = true;)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{n \ge 1 \wedge i = 0\}$$

$$zerorik\_ez := true;$$

$$\{\psi\} \equiv \{0 \le i \le n \wedge (zerorik\_ez \leftrightarrow \forall k(1 \le k \le i \rightarrow A(k) \ne 0))\}$$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

[#] Ariketa denetan boolearrak ez diren aldagai denak *integer* motakoak direla kontsideratuko da eta bektoreen kasuan (A, B eta abar) zenbaki osozkoak direla eta n elementu dituztela kontsideratuko da.

4. Esleipena (i : = 1;)

Honako programa hau zuzena al den egiaztatu:

$\{\varphi\} \equiv \{n \geq 1 \wedge A(1) \neq 0 \wedge zerorik\_ez\}$
i : = 1;
$\{\psi\} \equiv \{1 \leq i \leq n \wedge (zerorik\_ez \leftrightarrow \forall k(1 \leq k \leq i \rightarrow A(k) \neq 0))\}$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

5. Esleipena (i : = i * j;)

Honako programa hau zuzena al den egiaztatu:

$\{\varphi\} \equiv \{i \geq j^k \wedge i < w\}$
i : = i * j;
$\{\psi\} \equiv \{i = j^{k+1}\}$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

6. Esleipena (z : = x;)

Honako programa hau zuzena al den egiaztatu:

$\{\varphi\} \equiv \{x \geq y\}$
z : = x;
$\{\psi\} \equiv \{z = handiena(x, y)\}$

handiena(x, y) funtzioak x itzuliko du $x \geq y$ betetzen baldin bada eta y itzuliko du $x < y$ betetzen baldin bada.

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

7. Esleipena (m : = A(i + 1);)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{m = \text{handiena}(A(1..i)) \wedge (1 \leq i \leq n - 1) \wedge A(i + 1) > m\}$$

$$m : = A(i + 1);$$

$$\{\psi\} \equiv \{m = \text{handiena}(A(1..i + 1)) \wedge (1 \leq i \leq n - 1)\}$$

handiena(Q(1..r)) funtzioak Q(1..r) bektoreko balio handiena itzuliko du. Adibidez, handiena((4, 0, 10, 6)) espresioaren emaitza edo balioa 10 izango litzateke.

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

8. Esleipena (ez_zero : = neg + pos;)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{neg = N\ i\ (1 \leq i \leq n \wedge A(i) < 0) \wedge pos = N\ i\ (1 \leq i \leq n \wedge A(i) > 0)\}$$

$$ez\_zero : = neg + pos;$$

$$\{\psi\} \equiv \{ez\_zero = N\ i\ (1 \leq i \leq n \wedge A(i) \neq 0)\}$$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

9. Konposizio sekuentziala (s : = s + A(i); i : = i + 1;)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{1 \leq i \leq n \wedge s = \sum_{k=1}^{i-1} A(k)\}$$

$$s : = s + A(i);$$
$$i : = i + 1;$$

$$\{\psi\} \equiv \{1 \leq i \leq n + 1 \wedge s = \sum_{k=1}^{i-1} A(k)\}$$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

10. Konposizio sekuentziala (k : = k div 2; z : = z ∗ z;)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{\text{bikoitia}(k) \wedge y * z^k = c\}$$

$$k := k \text{ div } 2;$$
$$z := z * z;$$

$$\{\psi\} \equiv \{y * z^k = c\}$$

div zatiketa osoa da (Adibideak: 30 div 2 = 15; 9 div 2 = 4).

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

11. Konposizio sekuentziala (k : = k + 1; i : = i ∗ j;)

Honako programa hau zuzena al den egiaztatu:

$$\{\varphi\} \equiv \{i = j^k\}$$

$$k := k + 1;$$
$$i := i * j;$$

$$\{\psi\} \equiv \{i = j^k\}$$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

b) Iterazioak[#]

1. Negatiboak ez diren x eta y-ren arteko biderkadura z aldagaian kalkulatzen duen programa

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera programak negatiboak ez diren x eta y-ren arteko biderkadura kalkulatzen du z aldagaian:

{φ} ≡ {x = *a* ≥ 0 ∧ y ≥ 0}
z : = 0;
<u>while</u> {INB} x ≠ 0 <u>loop</u>
    z : = z + y;
    x : = x – 1;
<u>end</u> <u>loop</u>;
{ψ} ≡ {z = *a* * y}
	{INB}≡ {0 ≤ x ≤ *a* ∧ z = (*a* – x) * y }
	
	E = x

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

2. bider aldagai boolearrean B(1..n) bektoreko balioak A(1..n) bektoreko balioak bider x al diren erabakitzen duen programa

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera programak *bider* aldagai boolearrean B(1..n) bektoreko balioak A(1..n) bektoreko balioak bider x al diren erabakiko du:

{φ} ≡ {n ≥ 1 ∧ bider}
i : = 1;
<u>while</u> **{INB}** i ≤ n <u>and</u> bider <u>loop</u>
    bider : =(A(i) * x = B(i));
    i : = i + 1;
<u>end</u> <u>loop</u>;
{ψ} ≡ {bider ↔ ∀k(1 ≤ k ≤ n → B(k) = A(k) * x)}

{INB}≡ {(1 ≤ i ≤ n + 1) ∧ (bider ↔ ∀k(1 ≤ k ≤ i – 1 → B(k) = A(k) * x))} **E = n + 1 – i**

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

[#] Ariketa denetan boolearrak ez diren aldagai denak *integer* motakoak direla kontsideratuko da eta bektoreen kasuan (A, B eta abar) zenbaki osozkoak direla eta n elementu dituztela kontsideratuko da.

3. batura aldagai boolearrean C(1..n) bektoreko elementuak A(1..n) eta B(1..n) bektoreetako elementuen batura al diren erabakitzen duen programa

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera programak *batura* aldagai boolearrean C(1..n) bektoreko elementuak A(1..n) eta B(1..n) bektoreetako elementuen batura al diren erabakiko du:

```
{φ} ≡ {n ≥ 1 ∧ i = 1}
batura : = true;
while {INB} i ≤ n and batura loop
      batura : = (C(i) = A(i) + B(i));
      i : = i + 1;
end loop;
{ψ} ≡ {batura ↔ ∀k(1 ≤ k ≤ n → C(k) = A(k) + B(k))}
```

{INB}≡ $\{(1 \le i \le n + 1) \wedge (batura \leftrightarrow \forall k(1 \le k \le i - 1 \rightarrow C(k) = A(k) + B(k)))\}$
E = $n + 1 - i$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

4. txik aldagai boolearrean A(1..n) bektoreko elementu denak B(1..n) bektoreko posizio bereko elementuak baino txikiagoak al diren erabakitzen duen programa

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera programak *txik* aldagai boolearrean A(1..n) bektoreko elementu denak B(1..n) bektoreko posizio bereko elementuak baino txikiagoak al diren erabakiko du:

```
{φ} ≡ {n ≥ 1 ∧ i = 1}
txik : = true;
while {INB} i ≤ n and txik loop
      i : = i + 1;
      txik : = (A(i) < B(i));
end loop;
{ψ} ≡ {txik ↔ ∀k(1 ≤ k ≤ n → A(k) < B(k))}
```

{INB}≡ $\{(1 \le i \le n + 1) \wedge (txik \leftrightarrow \forall k(1 \le k \le i - 1 \rightarrow A(k) < B(k)))\}$

E = $n + 1 - i$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

5. x aldagaian ≥ 0 den n zenbakiari Fibonacci-ren segidan dagokion s_n elementua kalkulatzen duen programa

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera, ≥ 0 den n zenbakia emanda programak Fibonacci-ren segidako s_n elementua kalkulatuko du x aldagaian ($s_0 = 0$, $s_1 = 1$ eta $s_k = s_{k-1} + s_{k-2}$, $k \geq 2$ denean):

```
{φ} ≡ {n ≥ 1 ∧ x = 0 ∧ z = 1 ∧ j = 1}
while {INB} j ≤ n loop
      z : = z + x;
      x : = z – x;
      j : = j + 1;
end loop;
{ψ} ≡ {x = sn}
```

{INB}≡ $\{(1 \leq j \leq n + 1) \wedge x = s_{j-1} \wedge z = s_j\}$
E = n + 1 – j

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

6. x zenbakiaren faktoriala f aldagaian kalkulatzen duen programa.

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera programak, ≥ 0 den x zenbakia emanda, programak x-en faktoriala kalkulatuko du f aldagaian:

```
{φ} ≡ {x ≥ 0 ∧ f = 1}
t : = x;
while {INB} t ≥ 1 loop
      t : = t – 1;
      f : = f ∗ t;
end loop;
```

{ψ} ≡ $\{f = \prod_{i=1}^{x} i\}$

{INB}≡ $\{(0 \leq t \leq x) \wedge f = \prod_{i=t+1}^{x} i\}$
E = t

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

7. ord aldagai boolearrean A(1..n) bektoreko elementu denak goranzko ordenean al dauden erabakitzen duen programa.

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera programak A(1..n) bektoreko elementu denak goranzko ordenean al dauden erabakiko du, emaitza *ord* aldagai boolearrean lagaz:

$\{\varphi\} \equiv \{n \geq 1 \wedge i = 1\}$
ord : = true;
<u>while</u> {INB} $i \neq n$ **<u>and</u>** ord **<u>loop</u>**
    ord : = $(A(i) \leq A(i + 1))$;
    i : = i + 1;
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{ord \leftrightarrow gorakorra(A(1..n))\}$

$\{INB\} \equiv \{(1 \leq i \leq n) \wedge (ord \leftrightarrow gorakorra(A(1..i)))\}$ $E = n - i$ $gorakorra(C(1..p)) \equiv \{\forall k(2 \leq k \leq p \rightarrow C(k - 1) \leq C(k))\}$

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

8. hond aldagai boolearrean R(1..n) bektorean A(1..n) bektoreko elementuak 2 zenbakiaz zatitzean lortutako hondarrak al dauden erabakitzen duen programa.

Honako programa hau zuzena al den egiaztatu emandako espezifikazioa, inbariantea eta E espresioa kontuan hartuz. Espezifikazioaren arabera programak R(1..n) bektorean A(1..n) bektoreko elementuak 2 zenbakiaz zatitzean lortutako hondarrak al dauden erabakiko du hond aldagaian:

$\{\varphi\} \equiv \{n \geq 1 \wedge hond\}$
i : = 0;
<u>while</u> {INB} $i \neq n$ **<u>and</u>** hond **<u>loop</u>**
    hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1));
    i : = i + 1;
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{hond \leftrightarrow hondarrak(A(1..n), R(1..n))\}$

$\{INB\} \equiv \{(0 \leq i \leq n) \wedge (hond \leftrightarrow hondarrak(A(1..i), R(1..i)))\}$ $E = n - i$ hondarrak(D(1..p), F(1..p)) ≡ {∀k(1 ≤ k ≤ p → D(k) <u>mod</u> 2 = F(k))} **mod** eragilea zatiketa osoaren hondarra da. (Adibideak: 10 mod 5 = 0; 10 mod 4 = 2; 10 mod 3 = 1)

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

9. (2009ko ekaina) sim aldagai boolearrean A(1..n) bektorea simetrikoa al den erabakitzen duen programa.

Honako programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz. Espezifikazioaren arabera programak A(1..n) bektorea emanda, A(1..n) simetrikoa al den erabaki behar du *sim* aldagai boolearrean. A(1..n) bektorea simetrikoa dela esango da bere elementuak alderantzizko ordenean ipiniz lortzen den bektore berria A(1..n) bektorearen berdina bada. Adibidez (1, 8, 5, 8, 1) bektorea simetrikoa da eta baita (7, 2, 2, 7) bektorea ere:

$\{\varphi\} \equiv \{n \geq 1 \wedge i = 0\}$
sim : = true;
<u>while</u> {INB} $i \neq (n \text{ div } 2)$ **and** sim **<u>loop</u>**
    i : = i + 1;
    sim : = (A(i) = A(n – i + 1));
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{sim \leftrightarrow simetrikoa(A(1..n), n \text{ div } 2)\}$

$\{INB\} \equiv \{(0 \leq i \leq n \text{ div } 2) \wedge (sim \leftrightarrow simetrikoa(A(1..n), i))\}$ $E = n \text{ div } 2 - i$ $simetrikoa(A(1..n), pos) \equiv \forall k(1 \leq k \leq pos \rightarrow A(k) = A(n - k + 1))$ **div** zatiketa osoa da (Adibideak: 30 div 2 = 15; 9 div 2 = 4)

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

10. (2009ko iraila) aniz aldagai boolearrean A(1..n) bektorea B(1..n) bektorea bider x al den erabakitzen duen programa.

Honako programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz. Espezifikazioaren arabera programak zenbaki osoz osatutako A(1..n) eta B(1..n) bektoreak eta x zenbaki osoa emanda, A(1..n) bektorea B(1..n) bektorea bider x al den erabaki behar du *aniz* aldagai boolearrean. Adibidez (3, 12, 9, 15, 15) bektorea (1, 4, 3, 5, 5) bektorea bider 3 da:

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

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

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

11. (2010eko ekaina) anizpos aldagai boolearrean A(1..n) bektoreko posizio bakoitzean dagoen balioa posizioaren anizkoitza al den erabakitzen duen programa.

Honako programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz. Espezifikazioaren arabera programak, A(1..n) bektorea emanda, A(1..n) bektoreko posizio bakoitzean dagoen balioa posizioaren anizkoitza al den erabakitzen du *anizpos* aldagai boolearrean. Adibidez (1, 8, 15, 8, 20) bektorearen posizio bakoitzean posizioaren anizkoitza den balio bat dago (posizioak 1, 2, 3, 4 eta 5 dira). Bestalde, (1, 8, 7, 8, 20) bektorean hirugarren posizioko balioa ez da posizioaren anizkoitza:

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

$\{INB\} \equiv \{(0 \leq i \leq n) \wedge (anizpos \leftrightarrow \forall k(1 \leq k \leq i \rightarrow (A(k) \bmod k) = 0))\}$ $E = n - i$ **mod** eragilea zatiketa osoaren hondarra da. (Adibideak: 10 mod 5 = 0; 10 mod 4 = 2; 10 mod 3 = 1)

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

12. (2010eko iraila) aniz aldagai boolearrean A(1..n) bektoreko osagairen bat x zenbakiaren anizkoitza al den erabakitzen duen programa.

Honako programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz. Espezifikazioaren arabera, zenbaki osoz osatutako A(1..n) bektorea eta x zenbaki osoa emanda, programak A(1..n) bektoreko osagairen bat x zenbakiaren anizkoitza al den erabakitzen du *aniz* aldagai boolearrean:

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

$\{INB\} \equiv \{x \neq 0 \wedge (1 \leq i \leq n + 1) \wedge (aniz \leftrightarrow \exists k(1 \leq k \leq i - 1 \wedge A(k) \bmod x = 0))\}$ $E = n + 1 - i$ **mod** eragilea zatiketa osoaren hondarra da. (Adibideak: 10 mod 5 = 0; 10 mod 4 = 2; 10 mod 3 = 1)

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.