PROGRAMAZIOAREN METODOLOGIA
Azterketa Partziala – 3. gaia – Egiaztapena
2014ko martxoaren 13a

ARIKETA (Egiaztapena) – (2 puntu)

Honako programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz. Espezifikazioaren arabera programak, sarrerako datu bezala zenbaki osoz osatutako A(1..n) eta B(1..n) bektoreak emanda eta A(1..n) bektorean balio negatiborik ez dagoela jakinda, B(1..n) bektoreko osagairen bat posizio bereko A(1..n) bektoreko elementuaren erro karratuaren azpitik egindako hurbilketa osoa al den erabaki behar du *q* aldagai boolearrean.

{φ} ≡ {n ≥ 1 ∧ ¬q ∧ ezneg(A(1..n))}
i : = 0;
while {INB} {E} i ≠ n **and not** q **loop**
  i : = i + 1;
  q : = (ekaho(A(i)) = B(i));
end loop;
{ψ} ≡ {q ↔ errorenbat(A(1..n), B(1..n), n)}

{INB} ≡ {ezneg(A(1..n)) ∧ (0 ≤ i ≤ n) ∧ (q ↔ errorenbat(A(1..n), B(1..n), i))} E = n – i ezneg(H(1..r)) ≡ ∀k(1 ≤ k ≤ r → H(k) ≥ 0) errorenbat(F(1..r), G(1..r), pos) ≡ ∃k(1 ≤ k ≤ pos ∧ ekaho(F(k)) = G(k))

Hor **ekaho** funtzioa, negatiboa ez den zenbaki oso bat emanda, zenbaki horren **e**rro **k**arratuaren **a**zpitik egindako **h**urbilketa **o**soa kalkulatzen duen funtzioa da (Adibideak: ekaho(9) = 3, ekaho(10) = 3, ekaho(8) = 2).

Egiaztapena egiterakoan, **ekaho** funtzioa inplementatuta dagoela suposatu behar da eta erabiltzeaz bakarrik arduratu beharko dugu, mod, div, eta antzeko beste funtzioekin egiten dugun bezala. Gainera, negatiboa den argumentu bat emanez gero, **ekaho** funtzioak errorea sortuko du. Hori dela eta, ekaho(x) erako dei batek errorerik ez sortzeko x balioak 0 edo handiagoa izan beharko luke.

Programa zuzena baldin bada, zuzentasunaren froga eman behar da.

Puntuazioa:

a) Hasierako zatiketa eta eskema: 0,200
b) Hasierako esleipenaren egiaztapena: 0,150 (Kalkulua: 0,050. Inplikazioa: 0,100)
c) While-aren erregelako (I) puntua: 0,010
d) While-aren erregelako (II) puntua: 0,040
e) While-aren erregelako (III) puntua: 0,700 (Kalkulua: 0,200. Inplikazioa: 0,500)
f) While-aren erregelako (IV) puntua: 0,350
   (Inplikazio erraza: 0,100. Inplikazio zaila: 0,250)
g) While-aren erregelako (V) puntua: 0,100
h) While-aren erregelako (VI) puntua: 0,200 (Kalkulua: 0,050. Inplikazioa: 0,150)
i) Zuzentasunaren froga: 0,250

Inplikazio bat zergatik betetzen den ez bada azaltzen, zero kontatuko da, hau da, inplikazio bat betetzen dela esateak zergatik betetzen den azaldu gabe, zero balio du.

PROGRAMAZIOAREN METODOLOGIA
2014ko martxoaren 13ko partzialaren soluzioa
3. gaia -- Egiaztapena
Kudeaketaren eta Informazio Sistemen Informatikaren Ingeniaritzako Gradua
Bilboko IITUE (UPV / EHU)
Lengoaia eta Sistema Informatikoak Saila

Ariketa (Egiaztapena)

Honako programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz. Espezifikazioaren arabera programak, sarrerako datu bezala zenbaki osoz osatutako A(1..n) eta B(1..n) bektoreak emanda eta A(1..n) bektorean balio negatiborik ez dagoela jakinda, B(1..n) bektoreko osagairen bat posizio bereko A(1..n) bektoreko elementuaren erro karratuaren azpitik egindako hurbilketa osoa al den erabaki behar du *q* aldagai boolearrean.

$\{\varphi\} \equiv \{n \geq 1 \wedge \neg q \wedge \text{ezneg}(A(1..n))\}$
i : = 0;
while {INB} {E} i ≠ n **and not** q **loop**
  i : = i + 1;
  q : = (ekaho(A(i)) = B(i));
end loop;
$\{\psi\} \equiv \{q \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), n)\}$

$\{INB\} \equiv \{\text{ezneg}(A(1..n)) \wedge (0 \leq i \leq n) \wedge (q \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), i))\}$ $E = n - i$ $\text{ezneg}(H(1..r)) \equiv \forall k(1 \leq k \leq r \rightarrow H(k) \geq 0)$ $\text{errorenbat}(F(1..r), G(1..r), pos) \equiv \exists k(1 \leq k \leq pos \wedge \text{ekaho}(F(k)) = H(k))$

Hor **ekaho** funtzioa **e**rro **k**arratuaren **a**zpitik egindako **h**urbilketa **o**soa kalkulatzen duen funtzioa da (Adibideak: ekaho(9) = 3, ekaho(10) = 3, ekaho(8) = 2)

Egiaztapena egiterakoan, **ekaho** funtzioa inplementatuta dagoela suposatu behar da eta erabiltzeaz bakarrik arduratu beharko dugu, mod, div, eta antzeko beste funtzioekin egiten dugun bezala. Gainera, argumentu negatiboa den argumentu bat emanez gero, **ekaho** funtzioak errorea sortuko du. Hori dela eta, ekaho(x) erako dei batek errorerik ez sortzeko x balioak 0 edo handiagoa izan beharko luke.

Hasierako zatiketa eta eskema:

- While aginduaren aurretik esleipen bat dagoenez, while-aren aurrebaldintza edo hasierako baldintza bezala {INB} ipini behar da.

```
{φ}
i : = 0;
{INB}
while {INB} {E} i ≠ n and not q loop
  i : = i + 1;
  q : = (ekaho(A(i)) = B(i));
end loop;
{ψ}
```

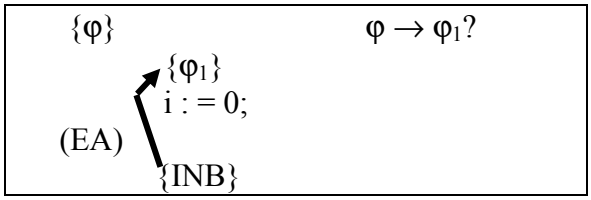
- Bi azpiprograma bereiztu behar dira:

```
{φ}
i : = 0;
{INB}
```

eta

```
{INB}
while {INB} {E} i ≠ n and not q loop
  i : = i + 1;
  q : = (ekaho(A(i)) = B(i));
end loop;
{ψ}
```

➢ Hasteko lehenengo azpiprograma zuzena dela egiaztatuko da



- $\{\varphi_1\} \equiv \{def(0) \wedge (INB)_i^0\} \equiv$

$\equiv \{true \wedge ezneg(A(1..n)) \wedge (0 \le 0 \le n) \wedge (q \leftrightarrow errorenbat(A(1..n), B(1..n), 0))\} \equiv$ sinplifikazioa

$\equiv \{ezneg(A(1..n)) \wedge (0 \le n) \wedge (q \leftrightarrow errorenbat(A(1..n), B(1..n), 0))\} \equiv$ formula

$\equiv \{ezneg(A(1..n)) \wedge (0 \le n) \wedge (q \leftrightarrow \exists k(1 \le k \le 0 \wedge ekaho(A(k)) = B(k)))\} \equiv$ eremu hutsa

$\equiv \{ezneg(A(1..n)) \wedge (0 \le n) \wedge (q \leftrightarrow false)\} \equiv$ sinplifikazioa

$\equiv \{ezneg(A(1..n)) \wedge (0 \le n) \wedge \neg q\}$

- Lehenengo sinplifikazioan alde batetik δ edozein formula izanda, *true* ∧ δ ≡ δ betetzen dela hartu da kontuan. Beste aldetik 0 ≤ 0 beti betetzen denez, (0 ≤ 0 ≤ n) ipintzea (0 ≤ n) ipintzearen berdina da.

- Jarraian errorenbat(A(1..n), B(1..n), 0) predikatua dagokion formulaz ordezkatu da, hau da, ∃k(1 ≤ k ≤ 0 ∧ ekaho(A(k)) = B(k)) formulaz. Hor eremu hutsa duen formula existentzial bat geratzen zaigunez, badakigu bere balioa false dela.

- Azkeneko sinplifikazioa δ edozein formula izanda ere, δ ↔ *false* ≡ ¬δ betetzen delako egin da.

- $\varphi \rightarrow \varphi_1$?

$$\underbrace{\{n \geq 1}_{\alpha} \wedge \underbrace{\neg q}_{\beta} \wedge \underbrace{\text{ezneg}(A(1..n))\}}_{\gamma}$$

$$\downarrow ?$$

$$\underbrace{\{\text{ezneg}(A(1..n))}_{\gamma\text{-gatik}} \wedge \underbrace{0 \leq n}_{\alpha\text{-gatik}} \wedge \underbrace{\neg q\}}_{\beta\text{-gatik}}$$

$n \geq 1$ betetzen denez, $n \geq 0$ ere beteko da, hau da, $0 \leq n$ beteko da. Bestalde, φ formulan $\neg q \wedge \text{ezneg}(A(1..n))$ true dela esaten zaigunez, badakigu φ_1 formulako $\neg q \wedge \text{ezneg}(A(1..n))$ zatia true dela.

Orain bigarren azpiprograma hartu eta While-aren erregela (WE) aplikatuko dugu, aurrebaldintza {INB} dela eta bukaerako baldintza $\{\psi\}$ dela kontsideratuz. Erregela hori kontuan hartuz azpiprograma hau zuzena dela erabakitzeko jarraian zehazten diren kalkuluak eta egiaztapenak burutu beharko dira.

Eskema:

I. $\text{INB} \rightarrow \text{INB}$?

II. $\text{INB} \rightarrow \text{def(B)}$?

III. $\{\text{INB} \wedge \text{B}\}$ $(\text{INB} \wedge \text{B}) \rightarrow \varphi_3$?

$\{\varphi_3\}$
i : = i + 1;
(EA)
$\{\varphi_2\}$
q : = (ekaho(A(i)) = B(i));
(EA)
{INB}

IV. $(\text{INB} \wedge \neg\text{B}) \rightarrow \psi$?

V. $(\text{INB} \wedge \text{B}) \rightarrow \text{E} > 0$?

VI. $\{\text{INB} \wedge \text{B} \wedge \text{E} = \text{v}\}$ $(\text{INB} \wedge \text{B} \wedge \text{E} = \text{v}) \rightarrow \varphi_5$?

$\{\varphi_5\}$
i : = i + 1;
(EA)
$\{\varphi_4\}$
q : = (ekaho(A(i)) = B(i));
(EA)
$\{\text{E} < \text{v}\}$

I. $\text{INB} \rightarrow \text{INB}$? Bai, alde bietan gauza bera daukagulako.

II. INB → def(B)?
INB → def(i ≠ n **and not** q)?
INB → true? Bai, inplikazioaren bigarren zatian true daukagulako.

III.

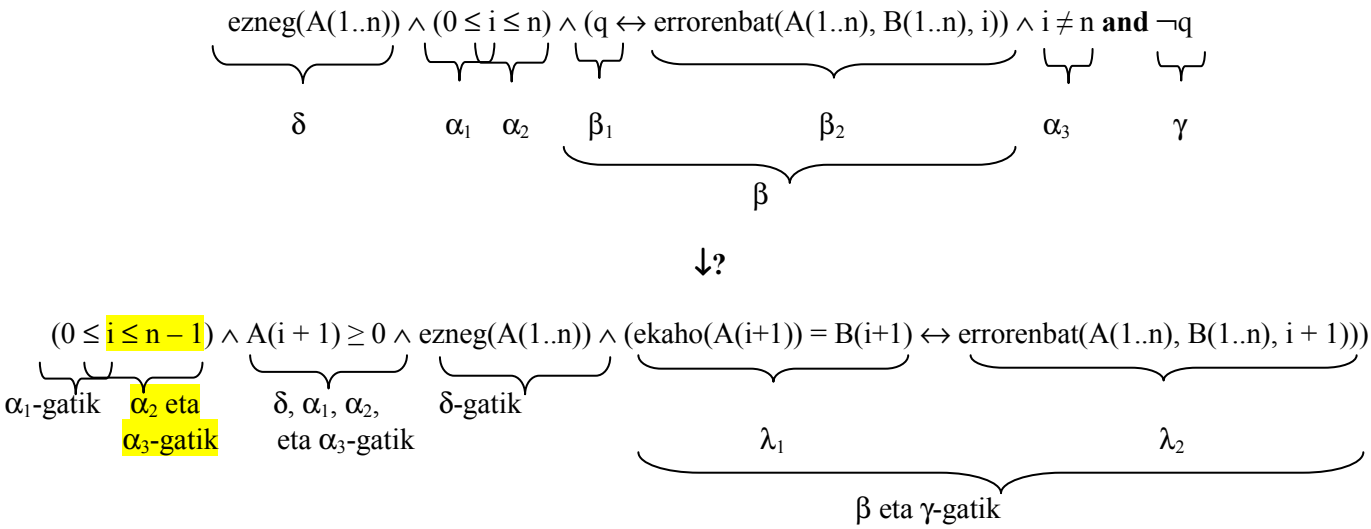
- $\{\varphi_2\} \equiv \{\text{def}(\text{ekaho}(A(i)) = B(i)) \wedge (\text{INB})_q^{\text{ekaho}(A(i)) = B(i)}\} \equiv$
  $\equiv \{(1 \le i \le n) \wedge (1 \le i \le n) \wedge A(i) \ge 0 \wedge$
  $\text{ezneg}(A(1..n)) \wedge (0 \le i \le n) \wedge$
  $(\text{ekaho}(A(i)) = B(i) \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), i))\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{(1 \le i \le n) \wedge A(i) \ge 0 \wedge$
  $\text{ezneg}(A(1..n)) \wedge (0 \le i \le n) \wedge$
  $(\text{ekaho}(A(i)) = B(i) \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), i))\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{(1 \le i \le n) \wedge A(i) \ge 0 \wedge$
  $\text{ezneg}(A(1..n)) \wedge$
  $(\text{ekaho}(A(i)) = B(i) \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), i))\}$

  Lehenengo sinplifikazioan $(1 \le i \le n)$ espresioa bi aldiz agertzen denez (bata A(i)-ren definizioagatik eta bestea B(i)-ren definizioagatik) bietako bat kendu egin da.
  Hala ere, bi eremu geratzen dira i-rentzat. Horregatik bigarren sinplifikazioan i aldagaiari dagokion eremua zein den erabakitzeko beheko mugetatik handiena (1 eta 0ren arteko handiena) eta goiko mugetatik txikiena (n eta n-ren arteko txikiena) hartu behar dira, beraz 1 eta n.

- $\{\varphi_3\} \equiv \{\text{def}(i + 1) \wedge (\varphi_2)_i^{i+1}\} \equiv$
  $\equiv \{\text{true} \wedge \text{ezneg}(A(1..n)) \wedge A(i + 1) \ge 0 \wedge (1 \le i + 1 \le n) \wedge$
  $(\text{ekaho}(A(i + 1)) = B(i + 1) \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), i + 1))$
  $\equiv^{\text{sinplifikazioa}}$
  $\equiv \{\text{ezneg}(A(1..n)) \wedge A(i + 1) \ge 0 \wedge (0 \le i \le n - 1) \wedge$
  $(\text{ekaho}(A(i + 1)) = B(i + 1) \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), i + 1))\}$

  Sinplifikazio urratsean alde batetik δ edozein formula izanda ere, *true* ∧ δ ≡ δ betetzen dela hartu da kontuan. Beste aldetik, (1 ≤ i + 1 ≤ n) eraldatu da, erdian i geldi dadin i + 1 balioaren ordez. Horretarako hiru osagaiei 1 kendu zaie: (1 – 1 ≤ i + 1 – 1 ≤ n – 1). Eragiketak burutu ondoren (0 ≤ i ≤ n – 1) gelditu da.

- $(INB \wedge B) \rightarrow \varphi_3$?



INB ∧ B betetzen dela jakinda, φ_3 betetzen al den jakin nahi da.
α_1-gatik $0 \leq i$ betetzen da.
α_2 eta α_3-gatik $i \leq n$ betetzen da.
α_1, α_2, eta α_3-gatik badakigu $i + 1$ balioa A bektorearen mugen barruan dagoela eta δ-gatik badakigu A bektoreko elementuak ez direla negatiboak, beraz $A(i + 1) \geq 0$ beteko da.

δ-gatik badakigu ezneg(A(1..n)) beteko dela.

φ_3 formulako $\lambda_1 \leftrightarrow \lambda_2$ inplikazio bikoitzari dagokionez, honako inplikazio hau egiaztatu behar da: $(INB \wedge B) \rightarrow (\lambda_1 \leftrightarrow \lambda_2)$?

β bezala izendatu dugun q ↔ denakerro(A(1..n), B(1..n), i) formularen esanahia honako hau da:

$q \leftrightarrow \exists k(1 \leq k \leq i \wedge ekaho(A(k)) = B(k))$

Disjuntzioa erabiliz, honela ere adieraz daiteke:

$q = (ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i))$

Bestetik $\lambda_1 \leftrightarrow \lambda_2$ bezala izendatu dugun
$(ekaho(A(i + 1)) = B(i + 1)) \leftrightarrow errorenbat(A(1..n), B(1..n), i + 1)$
formularen esanahia honako hau da:

$(ekaho(A(i + 1)) = B(i + 1)) \leftrightarrow \exists k(1 \leq k \leq i + 1 \wedge ekaho(A(k)) = B(k))$

Disjuntzioa erabiliz, honela ere adieraz daiteke:

$(ekaho(A(i + 1)) = B(i + 1)) = (ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i)) \vee (ekaho(A(i + 1)) = B(i + 1))$

Galdera honako hau da:

$q = (ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i))$

betetzen dela jakinda,

$(ekaho(A(i + 1)) = B(i + 1)) = (ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i)) \vee (ekaho(A(i + 1)) = B(i + 1))$

ere betetzen al da?

γ-gatik badakigu q-ren balioa false dela. Beraz, q eta $(ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i))$ formularen balioak berdinak direnez, $(ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i))$ formularen balioa ere false dela ondoriozta dezakegu.

π edozein formula izanda ere, $false \vee \pi \equiv \pi$ betetzen denez,

$(ekaho(A(i + 1)) = B(i + 1)) = \underbrace{(ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i))}_{false} \vee (ekaho(A(i + 1)) = B(i + 1))$

formulan false den zatia sinplifika daiteke eta galdera berria honako hau da

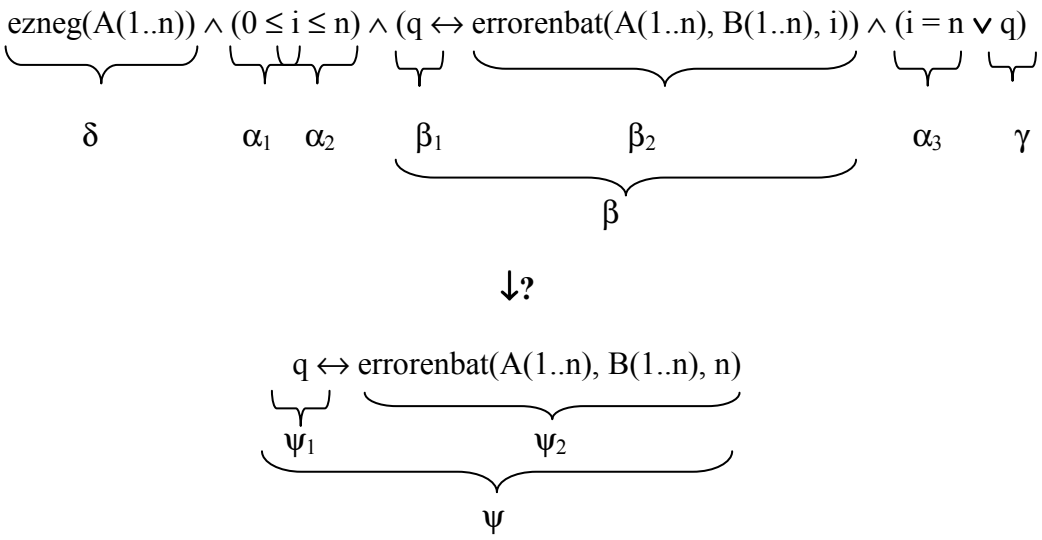
$(ekaho(A(i + 1)) = B(i + 1)) = (ekaho(A(i + 1)) = B(i + 1))$?

Eta erantzuna baiezkoa da alde biak berdinak direlako.

Ondorioz, $INB \wedge B$ formulak $\lambda_1 \leftrightarrow \lambda_2$ formula inplikatzen duela frogatu da, β eta γ zatietan zegoen informazioa erabiliz.

Beraz $(INB \wedge B) \rightarrow \varphi_3$ inplikazioa bete egiten da.

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



Helburua δ, α_1, α_2, β eta $(\alpha_3 \vee \gamma)$ egiazkoak direla jakinda, ψ egiazkoa al den erabakitzea da. Kasu honetan $\alpha_3 \vee \gamma$ disjuntzioa, hau da, $(i = n \vee q)$ disjuntzioa daukagunez, disjuntzio hori egia izateko dauden hiru aukerak hartu beharko dira kontuan:

i = n	**q**
True	True
True	False
False	True

- ✓ Lehenengo bi kasuetan **i = n** denez, β eta ψ formula bera dira. β betetzen dela dakigunez, ψ formula ere bete egiten da. Ondorioz inplikazioa ere bete egiten da.
- ✓ Hirugarren kasuan **i = n** false da eta **q = True** betetzen da. Ondorioz, α_2 kontuan hartuz, $i \leq n - 1$ betetzen da. Orain $i \leq n - 1$ betetzen denez, β eta ψ ez dira formula bera eta arrazonamendu desberdina jarraitu behar da. Gure helburua ψ betetzen al den erabakitzea da. β formularen esanahia honako hau da:

$$q \leftrightarrow \exists k(1 \leq k \leq i \wedge ekaho(A(k)) = B(k))$$

Disjuntzioa erabiliz, honela ere adieraz daiteke:

$$q = (ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i))$$

q true dela suposatzen ari garenez, badakigu

$$(ekaho(A(1)) = B(1)) \vee \ldots \vee (ekaho(A(i)) = B(i))$$

ere true dela.

ψ formularen esanahia honako hau da:

$$q \leftrightarrow \exists k(1 \leq k \leq n \wedge ekaho(A(k)) = B(k))$$

Disjuntzioa erabiliz, honela ere adieraz daiteke:

$$q = \underbrace{ekaho(A(1)) = B(1) \vee \ldots \vee ekaho(A(i)) = B(i)}_{true} \vee \ldots \vee ekaho(A(n)) = B(n)$$

β formula eta q = true informazioa erabiliz
$(ekaho(A(1)) = B(1)) \wedge \ldots \wedge (ekaho(A(i)) = B(i))$

formularen balioa true dela ikusi dugu. Gainera π edozein formula izanda ere $true \vee \pi \equiv true$ betetzen denez,

$(ekaho(A(1)) = B(1)) \wedge \ldots \wedge (ekaho(A(i)) = B(i)) \wedge \ldots \wedge (ekaho(A(n)) = B(n))$

formularen balioa ere true izango da $(ekaho(A(1)) = B(1)) \wedge \ldots \wedge (ekaho(A(i)) = B(i))$ true delako eta, ondorioz, q-ren balioaren berdina eta beraz, inplikazioa bete egiten da.

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

$$ezneg(A(1..n)) \wedge (0 \leq \underbrace{i \leq n}_{\alpha}) \wedge (q \leftrightarrow errorenbat(A(1..n), B(1..n), i)) \wedge \underbrace{i \neq n}_{\beta} \textbf{ and } \neg q$$

$$\downarrow ?$$

$$\underbrace{n - i > 0}_{\alpha \text{ eta } \beta\text{-gatik}}$$

Helburua $n - i > 0$ betetzen al den erabakitzea da.
α-gatik $i \leq n$ betetzen da, hau da, $n \geq i$ betetzen da. β-gatik $i \neq n$ dela ere badakigu. Beraz, $n > i$ betetzen da. Gure helburua $n - i$ espresioarekin zer gertatzen den jakitea denez, espresio horretako alde bietan i kenduko dugu: $n - i > i - i$. Sinplifikatuz $n - i > 0$ geratzen da, eta hori zen frogatu nahi genuena.

VI.

- $\{\varphi_4\} \equiv \{\text{def}(\text{ekaho}(A(i)) = B(i)) \wedge (E < v)_q^{\text{ekaho}(A(i)) = B(i)}\} \equiv$
  $\equiv \{(1 \le i \le n) \wedge (1 \le i \le n) \wedge A(i) \ge 0 \wedge (n - i < v)\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{(1 \le i \le n) \wedge A(i) \ge 0 \wedge (n - i < v)\}$

  $(n - i < v)$ espresioan q ez denez agertzen, q aldagaia ekaho(A(i)) = B(i) konparazioaz trukatzeak ez du eraginik.
  Sinplifikazioan $(1 \le i \le n)$ ezabatu da bi aldiz zegoelako.

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

- $(\text{INB} \wedge B \wedge E = v) \rightarrow \varphi_5$?

$$\underbrace{\text{ezneg}(A(1..n))}_{\delta} \wedge \underbrace{(0 \le i \le n)}_{\alpha_1\ \alpha_2} \wedge (q \leftrightarrow \text{errorenbat}(A(1..n), B(1..n), i - 1)) \wedge \underbrace{i \ne n}_{\alpha_3}\ \textbf{and}\ \neg q \wedge \underbrace{n - i = v}_{\beta}$$

$$\downarrow ?$$

$$\underbrace{(0 \le i \le n - 1)}_{\alpha_1\text{-gatik}\ \alpha_2\ \text{eta}\ \alpha_3\text{-gatik}} \wedge \underbrace{A(i + 1) \ge 0}_{\alpha_1, \alpha_2, \alpha_3\ \text{eta}\ \delta\text{-gatik}} \wedge \underbrace{(n - i - 1 < v)}_{\beta\text{-gatik}}$$

$0 \le i$ espresioa α_1-gatik betetzen da.
$i \le n - 1$ espresioa α_2 eta α_3-gatik betetzen da.
$A(i + 1) \ge 0$ espresioari dagokionez, α_1, α_2, α_3 espresioen ondorio bezala, badakigu $i + 1$ balioa A bektorearen mugen barruan dagoela eta δ-gatik $A(i + 1) \ge 0$ betetzen dela ziurta dezakegu.
β-gatik badakigu $n - i = v$ betetzen dela. Orain $n - i - 1 < v$ betetzen al den jakin nahi da. Hori dela eta, $n - i = v$ espresioan alde bietan 1 kenduz espresioa eraldatu egingo dugu, hau da, $n - i - 1 = v - 1$ geratzen da. Beraz orain badakigu $n - i - 1$ espresioaren balioa $v - 1$ dela eta galdera $n - i - 1$ espresioaren balioa v baino txikiagoa al den da. $v - 1$ beti v baino txikiagoa izaten denez, $n - i - 1$ espresioaren balioa v baino txikiagoa dela ziurta dezakegu.

Beraz $(\text{INB} \wedge B \wedge E < v) \rightarrow \varphi_5$ inplikazioa bete egiten da.

- **Zuzentasunaren froga:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ i : = 0; {INB} (EA)
3. $\{\varphi\}$ i : = 0; {INB} (OE 1, 2)

I

4. INB $\rightarrow$ INB

II

5. INB $\rightarrow$ def(B)

III

6. (INB $\wedge$ B) $\rightarrow \varphi_3$
7. $\{\varphi_3\}$ i : = i + 1; $\{\varphi_2\}$ (EA)
8. {INB $\wedge$ B} i : = i + 1; $\{\varphi_2\}$ (OE 6, 7)
9. $\{\varphi_2\}$ q : = (ekaho(A(i)) = B(i)); {INB} (EA)
10. {INB $\wedge$ B}
    i : = i + 1;
    q : = (ekaho(A(i)) = B(i));
    {INB} (KE 8, 9)

IV

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

V

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

VI

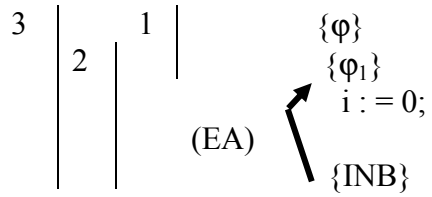
13. (INB $\wedge$ B $\wedge$ E = v) $\rightarrow \varphi_5$
14. $\{\varphi_5\}$ i : = i + 1; $\{\varphi_4\}$ (EA)
15. {INB $\wedge$ B $\wedge$ E = v} i : = i + 1; $\{\varphi_4\}$ (OE 13, 14)
16. $\{\varphi_4\}$ q : = (ekaho(A(i)) = B(i)); {E < v} (EA)
17. {INB $\wedge$ B $\wedge$ E = v}
    i : = i + 1;
    q : = (ekaho(A(i)) = B(i));
    {E < v} (KE 15, 16)

18. {INB}
    while {INB} i $\neq$ n **and not** q **loop**
        i : = i + 1;
        q : = (ekaho(A(i)) = B(i));
    end loop;
    $\{\psi\}$ (WE 4, 5, 10, 11, 12, 17)

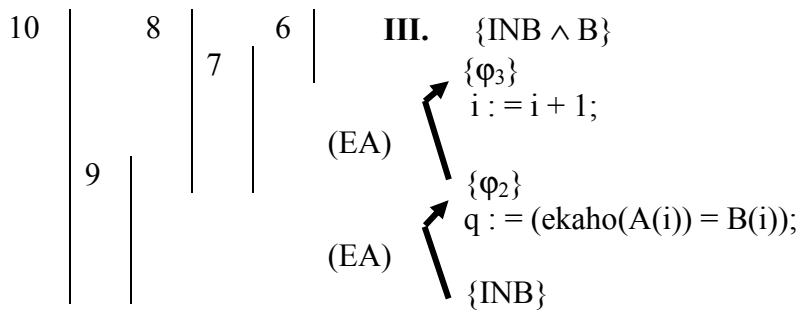
19. $\{\varphi\}$
    i : = 0;
    while {INB} i $\neq$ n **and not** q **loop**
        i : = i + 1;
        q : = (ekaho(A(i)) = B(i));
    end loop;
    $\{\psi\}$ (KE 3, 18)

Zuzentasunaren froga ematerakoan, While-aren aurreko esleipenari dagozkion hiru puntuak (1-3), while-aren erregelako III puntuari dagozkion bost puntuak (6-10) eta while-aren erregelako VI puntuari dagozkion bost puntuak (13-17) lortzeko, formulatik formulara doazen zatiak edo blokeak hartuz eta elkartuz joatea da onena, kasuan kasuko programa osatu arte.

While-aren aurretik dagoen esleipenaren kasuan, hasteko 1 eta 2 oinarrizko blokeak hartuko genituzke eta bi bloke horiek elkartzerakoan 3 blokea lortuko genuke. Oinarrizko blokeak (kasu honetan 1 eta 2) formulatik formulara joaten dira eta erdian gehienez esleipen bat izaten dute. Oinarrizko blokeak erdian esleipenik ez badu (1 oinarrizko blokeak bezala), orduan inplikazio bezala ipini beharko da. Oinarrizko blokeak esleipen bat badu (2 blokeak bezala), orduan oinarrizko bloke hori Esleipenaren Axiomagatik (EA) zuzena dela adierazi beharko da. Kasu honetan 1 blokeak esleipenik ez duenez, 1 eta 2 blokeak elkartzeko ondorioaren erregela (OE) erabili behar da.



While-aren erregelako III puntuari dagokion programan, hasteko 6 eta 7 oinarrizko blokeak hartuko genituzke eta bloke horiek elkartzean 8 blokea lortuko genuke. Oinarrizko blokeak (kasu honetan 6 eta 7) formulatik formulara joaten dira eta erdian gehienez esleipen bat izaten dute. Oinarrizko blokeak erdian esleipenik ez badu (6 oinarrizko blokeak bezala), orduan inplikazio bezala ipini beharko da. Oinarrizko blokeak esleipen bat badu (7 blokeak bezala), orduan oinarrizko bloke hori Esleipenaren Axiomagatik (EA) zuzena dela adierazi beharko da. Kasu honetan 6 blokeak esleipenik ez duenez, 6 eta 7 blokeak elkartzeko ondorioaren erregela (OE) erabili behar da. Jarraian 9 oinarrizko blokea hartuko genuke. 9 oinarrizko blokeak esleipen bat du eta oinarrizko bloke hori Esleipenaren Axiomagatik (EA) zuzena dela adierazi beharko da. Gero 8 eta 9 oinarrizko blokeak elkartu beharko dira. Bloke biek gutxienez esleipen bat dutenez, konposizioaren erregelaren bidez (KE) elkartuko dira. 8 eta 9 blokeak elkartzean 10 blokea lortzen da eta hori III puntuari dagokion programa da.



While-aren erregelako VI puntuari dagokion zatian planteamendua III puntuko planteamenduaren berdina da.