3. GAIA
PROGRAMEN EGIAZTAPENA

3.1. Programen egiaztapenera sarrera

3.1.1. Helburua

Programa bat eta hasierako eta bukaerako baldintzen bidez emandako espezifikazioa emanda, programa hori espezifikazio horrekiko zuzena al den erabakitzea da helburua.

Beste era batera esanda, honako galdera honi erantzun behar zaio:

Hasierako baldintza betetzen duen egoera batetik abiatzen bagara, P programa burutu ondoren bukaerako badintza beteko al da?

{φ}
P
{ψ}

Hasierako baldintza {φ} bezala adieraziko da normalean eta bukaerako baldintza {ψ} bezala adieraziko da.

3.1.2. Adibidea: Programa zuzen bat

{φ} ≡ {x = *a* ∧ y = b}
lag : = x;
x : = y;
y : = lag;
{ψ} ≡ {x = b ∧ y = *a*}

{φ} eta {ψ} formulen bidez emandako espezifikazioak programaren hasieran x aldagaiaren balioa *a* baldin bada eta y-rena b baldin bada, programa burutu ondoren x-en balioa b eta y-rena *a* izango direla adierazten du.

Gai honetan programa hori espezifikazio horrekiko zuzena dela nola frogatzen den erakutsiko da.

3.1.3. Adibidea: Zuzena ez den programa bat

{φ} ≡ {x = *a* ∧ y = b}
y : = x;
x : = y;
{ψ} ≡ {x = b ∧ y = *a*}

Adibide honetan ere {φ} eta {ψ} formulen bidez emandako espezifikazioak programaren hasieran x aldagaiaren balioa *a* baldin bada eta y-rena b baldin bada, programa burutu ondoren x-en balioa b eta y-rena *a* izango direla adierazten du.

Gai honetan ikusiko dugun metodoa erabiliz, programa hori espezifikazio horrekiko zuzena dela frogatu ahal izango dugu.

3.1.4. Zuzentasun partziala, amaiera eta zuzentasun osoa

P programa {φ} eta {ψ} formulen bidez emandako espezifikazioarekiko **partzialki zuzena** dela esateak honako hau adierazten du:

P programaren exekuzioa hasierako baldintza den {φ} formula betetzen duen egoera batetik abiatzen bada eta P programa bukatzen baldin bada, orduan amaieran bukaerako baldintza den {ψ} formula beteko da.

Beraz P programa {φ} eta {ψ} formulen bidez emandako espezifikazioarekiko partzialki zuzena dela esaten denean, ez da bermatzen P bukatuko denik. Gerta daiteke P inoiz ez bukatzea.

Esate baterako jarraian datorren programa hor agertzen diren {φ} eta {ψ} formulekiko partzialki zuzena da baina gerta daiteke inoiz ez bukatzea ere:

```
{φ} ≡ {x = a ∧ y = b}
while y ≠ 0 loop
        x : = x + 1;
        y : = y – 1;
end loop;
{ψ} ≡ {x = a + b ∧ y = 0}
```

Partzialki zuzena dela froga daiteke, hau da, hasieran $\{x = a \wedge y = b\}$ betetzen baldin bada eta while hori exekutatzen badugu eta while hori bukatzen baldin bada, orduan $\{x = a + b \wedge y = 0\}$ beteko da. Baina gerta daiteke while hori inoiz ez bukatzea, izan ere, hasieran y-ren balioa negatiboa bada, hau da, b negatiboa bada, orduan while-a ez da inoiz amaituko.

P programa {φ} eta {ψ} formulen bidez emandako espezifikazioarekiko **guztiz zuzena** dela esateak honako hau adierazten du:

P programaren exekuzioa hasierako baldintza den {φ} formula betetzen duen egoera batetik abiatzen baldin bada, orduan P programa amaitu egingo da eta amaieran bukaerako baldintza den {ψ} formula beteko da.

Beraz P programa {φ} eta {ψ} formulen bidez emandako espezifikazioarekiko guztiz zuzena dela esaten denean, P programa bukatu egingo dela bermatzen da.

Esate baterako jarraian datorren programa hor agertzen diren {φ} eta {ψ} formulekiko guztiz zuzena da:

$\{\varphi\} \equiv \{x = a \wedge y = b \wedge b \geq 0\}$
<u>while</u> y ≠ 0 <u>loop</u>
    x : = x + 1;
    y : = y – 1;
<u>end</u> <u>loop</u>;
$\{\psi\} \equiv \{x = a + b \wedge y = 0\}$

Programa hori $\{x = a \wedge y = b \wedge b \geq 0\}$ betetzen den egoera betetik abiatzen baldin bada eta while hori exekutatzen badugu, orduan programa bukatu egingo dela ziurta dezakegu eta gainera $\{x = a + b \wedge y = 0\}$ beteko da.

P programa {φ} eta {ψ} formulen bidez emandako espezifikazioarekiko guztiz zuzena dela frogatzeko, partzialki zuzena dela eta gainera beti bukatuko dela frogatu beharko da.

Zuzentasun osoa = zuzentasun partziala + bukatzea

Esleipenaren eta baldintzazko aginduen kasuan **zuzentasun partziala** eta **zuzentasun osoa** gauza bera dira, esleipena eta baldintzazko aginduak beti bukatzen baitira. Agindu iteratiboen kasuan (while-loop, loop-exit-when, ...) iterazioa inoiz ez bukatzea gerta daitekeenez, zuzentasun partziala frogatzeaz gain iterazioa bukatu egiten dela ere frogatu egin beharko da.

3.2. Hoare-ren kalkulua

Hoare-ren kalkuluan programen zuzentasuna frogatzeko balio duten axiomak eta erregelak erabiltzen dira.

Jarraian datozen ataletan programazio-lengoaietako agindu ohikoenentzat Hoare-ren kalkula nola erabiltzen den erakutsiko da:

- Esleipena.
- Konposizioa.
- Iterazioa (while).

3.2.1. Jarraitu beharreko urratsak

P programa bat {φ} eta {ψ} formulen bidez emandako espezifikazioarekiko guztiz zuzena al den galdetzen digutenean, Hoare-ren kalkulua erabiliz programa guztiz zuzena al den erabakitzen saiatuko gara:

- Guztiz zuzena dela ateratzen bada, zuzentasun osoaren froga formala eman beharko da.

- Partzialki zuzena dela ateratzen bada, hasteko zuzentasun partzialaren froga formala eman beharko da eta gero batzutan programa ez dela bukatuko erakusten duen adibide konkretu bat eman beharko da.

- Partzialki zuzena ez dela ateratzen bada, $\{\varphi\}$ baldintza betetzen duen egoera batetik abiatuta, $\{\psi\}$ betetzen ez duen egoera batean bukatzen den adibide bat eman beharko da.

3.2.2. Esleipenaren Axioma (EA) eta Ondorioaren Erregela (OE)

3.2.2.1. Esleipenaren Axioma (EA)

Hasteko, Hoare-ren kalkulua erabiliz esleipenaren zuzentasuna nola egiaztatzen den ikusiko dugu.

{φ}
x : = t;
{ψ}

x : = t; erako esleipen bat $\{\varphi\}$ hasierako baldintza eta $\{\psi\}$ bukaerako baldintzarekiko zuzena izango da honako fhau betetzen bada

$$\{\varphi\} \equiv \{def(t) \wedge \psi_x^t\}.$$

Beste era batera esanda, honako hau zuzena da:

Esleipenaren Axioma (EA)	**{def(t) ∧ ψ_x^t}** x : = t; **{ψ}**

def(t) formula bat izango da eta t espresioa definituta egoteko zein baldintza bete behar den adieraziko du. Bestalde, $\boldsymbol{\psi_x^t}$ formula $\boldsymbol{\psi}$ formulan **x**-en agerpen denak **t** espresioaz ordezkatuz lortzen den formula da.

Esleipenaren axiomak honako hau dio:

Bukaeran x-ek $\boldsymbol{\psi}$ formulak dioena betetzea nahi baldin badugu eta bukatu baino lehen x algaiari t espresioaren balioa esleitu behar badiogu, t espresioak kalkulagarria izan beharko duela (**def(t)**) eta gainera hasieran, esleipena burutu baino lehen, t espresioak bukaeran x aldagaiak bete beharko duena bete dezala ($\boldsymbol{\psi_x^t}$).

3.2.2.2. Lehenengo adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{1 \le i \le n \wedge A(i) > 0\}$
x : = A(i);
$\{\psi\} \equiv \{x > 0\}$

Bi gauza egin behar dira:

- $\{def(t) \wedge \psi_x^t\}$ formula kalkulatu. Formula berri horri $\{\varphi_1\}$ deituko diogu. Hor teoriako t espresioaren ordez A(i) daukagu.
- $\{\varphi\}$ eta $\{def(t) \wedge \psi_x^t\}$ berdinak al diren egiaztatu.

EA

$\{\varphi\} \equiv \{1 \le i \le n \wedge A(i) > 0\}$
$\{\varphi_1\} \equiv \{def(A(i)) \wedge \psi_x^{A(i)}\} \equiv \{1 \le i \le n \wedge A(i) > 0\}$
x : = A(i);
$\{\psi\} \equiv \{x > 0\}$

Kasu honetan def(A(i)) formula $1 \le i \le n$ da, A(i) espresioak erroririk eman ez dezan edo kalkulagarria izan dadin i indizeak A(1..n) bektorearen mugen barruan egon behar duelako, hau da i aldagairen balioak 1 eta n-ren artean egon beharko du.

φ eta φ_1 formulak berdinak direnez, esleipen bakarra duen programa hori zuzena da φ eta ψ formulen bidez emandako espezifikazioarekiko.

Hoare-ren kalkulua erabiltzerakoan, behetik, hau da, bukaerako baldintzatik ($\{\psi\}$ formulatik) abiatzen gara eta esleipenaren aurrean bete beharko den $\{\varphi_1\}$ formula kalkulatuko da $\{\psi\}$ formula eta x : = A(i); esleipena erabiliz. Bukatzeko φ eta φ_1 formulak berdinak al diren ala ez begiratu beharko da.

Esleipenaren axiomak kasu honetan honako hau dio:

Bukaeran x-en balioa 0 baino handiagoa izatea nahi badugu, esleipenaren bidez x aldagaiak A(i) balioa hartuko duenez, bukaeran x-en balioa 0 baino handiagoa izan dadin, hasieran A(i)-ren balioak kalkulagarria izan beharko du eta gainera 0 baino handiagoa izan beharko du. Horrela, x aldagaiari A(i) balioa esleitu ondoren x-en balioa 0 baino handiagoa izatea lortuko dugu.

Adibide honetako programa zuzena denez, Zuzentasunaren froga eman beharko da:

- **Zuzentasunaren froga:**

    1. $\{\varphi\}$ x : = A(i); $\{\psi\}$ (EA)

Froga horren bidez esleipen hori Esleipenaren Axiomagatik zuzena dela adierazten dugu.

3.2.2.3. Bigarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{z \neq 0 \wedge y / z \neq 1\}$
$x := y / z;$
$\{\psi\} \equiv \{x \neq 1\}$

Bi gauza egin behar dira:

- $\{def(t) \wedge \psi_x^t\}$ formula kalkulatu. Formula berri horri $\{\varphi_1\}$ deituko diogu. Hor teoriako t espresioaren ordez A(i) daukagu.
- $\{\varphi\}$ eta $\{def(t) \wedge \psi_x^t\}$ berdinak al diren egiaztatu.

AA

$\{\varphi\} \equiv \{z \neq 0 \wedge y / z \neq 1\}$
$\{\varphi_1\} \equiv \{def(y / z) \wedge \psi_x^{y/z}\} \equiv \{z \neq 0 \wedge y / z \neq 1\}$
$x := y / z;$
$\{\psi\} \equiv \{x \neq 1\}$

Kasu honetan def(y / z) formularen $z \neq 0$ da, y / z espresioa kalkulatu ahal izateko z aldagaiaren balioak 0 ezin duelako izan, hau da, 0-ren desberdina izan behar du.

φ eta φ_1 formulak berdinak direnez, esleipen bakarra duen programa hori zuzena da φ eta ψ formulen bidez emandako espezifikazioarekiko.

Hoare-ren kalkulua erabiltzerakoan, behetik, hau da, bukaerako baldintzatik ($\{\psi\}$ formulatik) abiatzen gara eta esleipenaren aurrean bete beharko den $\{\varphi_1\}$ formula kalkulatuko da $\{\psi\}$ formula eta $x := y / z$; esleipena erabiliz. Bukatzeko φ eta φ_1 formulak berdinak al diren ala ez begiratu beharko da.

Esleipenaren axiomak kasu honetan honako hau dio:

Bukaeran x-en balioa 1-en desberdina izatea nahi badugu, esleipenaren bidez x aldagaiak y / z balioa hartuko duenez, bukaeran x-en balioa 1en desberdina izan dadin, hasieran y / z-ren balioak kalkulagarria izan beharko du eta gainera 1en desberdina izan beharko du. Horrela, x aldagaiari y / z balioa esleitu ondoren x-en balioa 1en desberdina izatea lortuko dugu.

Adibide honetako programa zuzena denez, Zuzentasunaren froga eman beharko da:

- **<u>Zuzentasunaren froga:</u>**

    1. $\{\varphi\}\ x := y / z;\ \{\psi\}$ (EA)

Froga horren bidez esleipen hori Esleipenaren Axiomagatik zuzena dela adierazten dugu.

3.2.2.4. Hirugarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{z \neq 0 \wedge y = 0\}$
$x := y / z;$
$\{\psi\} \equiv \{x \neq 1\}$

Bi gauza egin behar dira:

- $\{def(t) \wedge \psi_x^t\}$ formula kalkulatu. Formula berri horri $\{\varphi_1\}$ deituko diogu. Hor teoriako t espresioaren ordez y / z daukagu.
- $\{\varphi\}$ eta $\{def(t) \wedge \psi_x^t\}$ berdinak al diren egiaztatu.

EA

$\{\varphi\} \equiv \{z \neq 0 \wedge y = 0\}$
$\{\varphi_1\} \equiv \{def(y / z) \wedge \psi_x^{y/z}\} \equiv \{z \neq 0 \wedge y / z \neq 1\}$
$x := y / z;$
$\{\psi\} \equiv \{x \neq 1\}$

Adibide honetan def(y / z) espresioa $z \neq 0$ da, izan ere y / z kalkulatu ahal izateko z aldagaiak zeroren desberdina izan beharko du.

Kasu honetan φ eta φ_1 ez dira berdinak baina programa begiratuz programa hori φ eta ψ formulen bidez emandako espezifikazioarekiko zuzena dela ikus daiteke. Beraz adibide honen bidez esleipen bat zuzena al den ala ez erabakitzeko kasu batzuetan Esleipenaren Axioma ez dela nahikoa ikus dezakegu. Ondorioaren Erregela behar dugu.

3.2.2.5. Ondorioaren Erregela (OE)

Ondorioaren Erregela (OE)

$\varphi \rightarrow \varphi_1,\ \{\varphi_1\}\ P\ \{\psi\}$
$\{\varphi\}\ P\ \{\psi\}$

φ formulak φ_1 inplikatzen badu (hau da, φ_1 formula φ formularen ondorioa baldin bada) eta $\{\varphi_1\}\ P\ \{\psi\}$ zuzena baldin bada, orduan $\{\varphi\}\ P\ \{\psi\}$ere zuzena iznago da.

Beste era batera esanda:

$\varphi \rightarrow \varphi_1$	betetzen baldin bada eta	$\{\varphi_1\}$ P $\{\psi\}$	zuzena baldin bada, orduan	$\{\varphi\}$ P $\{\psi\}$	ere zuzena izango da

Esleipenaren Axioma erabiliz eraikitzen den φ_1 formula φ formularen berdina ez denean, φ formulak φ_1 formula inplikatzen al duen begiratu beharko da. Beraz, esleipen bat φ hasierako baldintzarekiko eta ψ bukaerako baldintzarekiko zuzena al den erabakitzeko honako urratsak jarraitu beharko dira:

- $\{def(t) \wedge \psi_x^t\}$ kalkulatu. Formula horri φ_1 deituko diogu.
- $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den begiratu.

3.2.2.6. Hirugarren adibidea (jarraipena)

$\{\varphi\} \equiv \{z \neq 0 \wedge y = 0\}$
$\{\varphi_1\} \equiv \{z \neq 0 \wedge y / z \neq 1\}$

$\varphi \rightarrow \varphi_1$?

$$\underbrace{z \neq 0}_{\alpha} \wedge \underbrace{y = 0}_{\beta} \rightarrow \underbrace{z \neq 0}_{\text{bai, }\alpha\text{-gatik}} \wedge \underbrace{y / z \neq 1}_{\text{bai, }\alpha\text{ eta }\beta\text{-gatik}}?$$

Hor erabaki beharreko honako hau da:
$z \neq 0 \wedge y = 0$ betetzen bada, orduan $z \neq 0 \wedge y / z \neq 1$ ere beteko al da?

Erantzuna baiezkoa da eta ondorioz 3. adibideko programa zuzena da. Eskuineko aldeko espresioa egiazkoa dela ziurtatzeko ezkerreko aldeko zein zati behar diren letra grekoen bidez zehazten da. Kasu honetan $z \neq 0$ denez eta $y = 0$ denez, $y / z \neq 1$ egia dela esan dezakegu.

- **<u>Zuzentasunaren froga:</u>**

    1. $\varphi \rightarrow \varphi_1$
    2. $\{\varphi_1\}$ x : = y / z; $\{\psi\}$ (EA)
    3. $\{\varphi\}$ x : = y / z; $\{\psi\}$ (OE 1, 2)

  Zuzentasunaren frogan programa zuzena dela ziurtatzeko zein axioma eta erregela erabili diren adierazi behar da. Kasu honetan 3. urratsean programa osoa zuzena dela esaten da Ondorioaren Erregela erabiliz, izan ere aurretik $\varphi \rightarrow \varphi_1$ inplikazioa betetzen dela eta Esleipenaren Axiomagatik $\{\varphi_1\}$ x : = y / z; $\{\psi\}$ programa zuzena dela ikusi baitugu.

Programa zuzena dela erabakitzeko honako hau egin da:

- $\{def(t) \wedge \psi_x^t\}$ kalkulatu x : = y / z; esleipena eta $\{\psi\}$ bukaerako baldintza erabiliz.
- $\{\varphi\}$ eta $\{def(t) \wedge \psi_x^t\}$ berdinak al diren begiratu.
- Desberdinak direnez, $\varphi \rightarrow (def(t) \wedge \psi_x^t)$ inplikazioa betetzen al den aztertu
- Inplikazio hori bete egiten denez, programa zuzena da eta bukatzeko zuzentasunaren froga eman dugu.

3.2.2.7. Laugarren adibidea

Honako programa hau zuzena al da?

{φ} ≡ {1 ≤ i ≤ n ∧ posit(A(1..n))}
x : = A(i);
{ψ} ≡ {x > 0}

Bertan posit(A(1..n)) predikatua honela definituta dagoela kontsideratu behar da:

$$\text{posit}(A(1..n)) \equiv \forall k\ (1 \le k \le n \rightarrow A(k) > 0)$$

Beraz posit(A(1..n)) predikatuak A(1..n) taulako elementu denak positiboak (> 0) direla adierazten du.

Orain honako bi urrats hauek jarraitu behar ditugu:

- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^{t}\}$ kalkulatu t-ren balioa A(i) dela kontsideratuz.
- {φ} eta $\{\varphi_1\}$ berdinak al diren begiratu eta berdinak ez badira, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu.

EA

{φ} ≡ {1 ≤ i ≤ n ∧ posit(A(1..n))}
$\{\varphi_1\} \equiv \{def(A(i)) \wedge \psi_x^{A(i)}\} \equiv \{1 \le i \le n \wedge A(i) > 0\}$
x : = A(i);
{ψ} ≡ {x > 0}

Kasu honetan def(A(i)) espresioaren balioa 1 ≤ i ≤ n da, izan ereA(i) kalkulatzeko edo A(i) balioa erabiltzeko i indizeak A(1..n) bektorearen mugen barruan egon beharko du, hau da, i aldagaiak 1 eta n-ren arteko balio bat eduki behar du.

φ eta φ_1 ez direnez berdinak, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den ala ez begiratu beharko da, hau da, φ_1 formula φ formularen ondorioa al den aztertu beharko da.

$\varphi \rightarrow \varphi_1$?

$$\underbrace{1 \le i \le n}_{\alpha} \wedge \underbrace{\text{posit}(A(1..n))}_{\beta} \rightarrow \underbrace{1 \le i \le n}_{\alpha\text{-gatik}} \wedge \underbrace{A(i) > 0}_{\alpha \text{ eta } \beta\text{-gatik}}?$$

1 ≤ i ≤ n ∧ posit(A(1..n)) betetzen baldin bada, 1 ≤ i ≤ n ∧ A(i) > 0 ere beteko al da?

Erantzuna baiezkoa da eta ondorioz 4. adibideko programa zuzena da. Eskuineko aldeko espresioa egiazkoa dela ziurtatzeko ezkerreko aldeko zein zati behar diren letra grekoen bidez zehaztu da. Adibidez, A(i) > 0 betetzen da posit(A(1..n)) betetzen delako.

- **Zuzentasunaren froga:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ x : = A(i); {ψ} (EA)
3. {φ} x : = A(i); {ψ} (OE 1, 2)

Zuzentasunaren frogan programa zuzena dela ziurtatzeko zein axioma eta erregela erabili diren adierazi behar da. Kasu honetan 3. urratsean programa osoa zuzena dela esaten da Ondorioaren Erregela erabiliz, izan ere aurretik $\varphi \rightarrow \varphi_1$ inplikazioa betetzen dela eta Esleipenaren Axiomagatik $\{\varphi_1\}$ x : = A(i); $\{\psi\}$ programa zuzena dela ikusi baitugu.

Programa zuzena dela erabakitzeko honako hau egin da:

- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^t\}$ kalkulatu x : = A(i); esleipena eta $\{\psi\}$ bukaerako baldintza erabiliz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu.
- Desberdinak direnez, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu
- Inplikazio hori bete egiten denez, programa zuzena da eta bukatzeko zuzentasunaren froga eman dugu.

3.2.2.8. Bosgarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{1 \leq i < n\}$
i : = i + 1;
$\{\psi\} \equiv \{1 \leq i \leq n\}$

Orain honako bi urrats hauek jarraitu behar ditugu:

- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^t\}$ kalkulatu t-ren balioa i + 1 dela kontsideratuz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu eta berdinak ez badira, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu.

EA

$\{\varphi\} \equiv \{1 \leq i \leq n - 1\}$
$\{\varphi_1\} \equiv \{def(i + 1) \wedge \psi_x^{i+1}\} \equiv \{true \wedge 1 \leq i + 1 \leq n\} \equiv^{\text{sinplifikazioa}} \{0 \leq i \leq n - 1\}$
i : = i + 1;
$\{\psi\} \equiv \{1 \leq i \leq n\}$

def(i + 1) espresioaren balioa *true* da i + 1 beti kalkula daitekeelako, i + 1 kalkulatu ahal izateko ez da inolako baldintzarik bete behar aurretik.

$\{def(i + 1) \wedge \psi_x^{i+1}\}$ kalkulatu ondoren sinplifikatu egin da, ulertzeko errazagoa izan dadin. Sinplifikazioa egiteko $\delta \equiv true \wedge \delta$ baliokidetasuna erabili da, edozein formula δ-rentzat hori horrela baita. Bestalde, **$1 \leq i + 1 \leq n$** espresioa i + 1 ipiniz baino bakarrik i ipiniz hobeto ulertzen da. **$1 \leq i + 1 \leq n$** espresioa i + 1 –en ordez i erabiliz adierazteko, bere hiru osagaiei 1 balioa kendu behar zaie kenketaren bidez: **$1 - 1 \leq i + 1 - 1 \leq n - 1$**. eragiketak burutu ondoren **$0 \leq i \leq n - 1$** gelditzen da.

φ eta φ_1 ez direnez berdinak, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu behar da.

$\varphi \rightarrow \varphi_1$?

$$\underbrace{(1 \leq i \leq n - 1)}_{\alpha} \rightarrow \underbrace{0 \leq i \leq n - 1?}_{\text{bai, } \alpha\text{-gatik}}$$

Beraz, $1 \le i \le n - 1$ betetzen baldin bada, $0 \le i \le n - 1$ betetzen al da?

Erantzuna baiezkoa da eta ondorioz 5. adibideko programa zuzena da. Eskuineko aldeko espresioa egiazkoa dela ziurtatzeko ezkerreko aldeko zein zati behar diren letra grekoen bidez zehaztu da. Adibidez, $0 \le i \le n - 1$ betetzen da $1 \le i \le n - 1$ betetzen delako.

- **Zuzentasunaren froga:**

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

Zuzentasunaren frogan programa zuzena dela ziurtatzeko zein axioma eta erregela erabili diren adierazi behar da. Kasu honetan 3. urratsean programa osoa zuzena dela esaten da Ondorioaren Erregela erabiliz, izan ere aurretik $\varphi \rightarrow \varphi_1$ inplikazioa betetzen dela eta Esleipenaren Axiomagatik $\{\varphi_1\}$ i : = i + 1; $\{\psi\}$ programa zuzena dela ikusi baitugu.

Programa zuzena dela erabakitzeko honako hau egin da:
- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^{\,t}\}$ kalkulatu i : = i + 1; esleipena eta $\{\psi\}$ bukaerako baldintza erabiliz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu.
- Desberdinak direnez, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu
- Inplikazio hori bete egiten denez, programa zuzena da eta bukatzeko zuzentasunaren froga eman dugu.

3.2.2.9. Seigarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{z = 1\}$
y : = 0;
$\{\psi\} \equiv \{z = x^y \wedge y \ge 0\}$

Orain honako bi urrats hauek jarraitu behar ditugu:
- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^{\,t}\}$ kalkulatu t-ren balioa 0 dela kontsideratuz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu eta berdinak ez badira, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu.

EA

$\{\varphi\} \equiv \{z = 1\}$
$\{\varphi_1\} \equiv \{def(0) \wedge \psi_y^{\,0}\} \equiv \{true \wedge z = x^0 \wedge 0 \ge 0\} \equiv^{sinplifikazioa} \{z = x^0\}$
y : = 0;
$\{\psi\} \equiv \{z = x^y \wedge y \ge 0\}$

def(0) espresioaren balioa *true* da 0 beti kalkula daitekeelako, 0 kalkulatu ahal izateko ez da inolako baldintzarik bete behar aurretik.

$\{def(0) \wedge \psi_x^0\}$ kalkulatu ondoren sinplifikatu egin da ulrtzeko errazagoa izan dadin. Sinplifikazioa egiteko alde batetik $\delta \equiv$ *true* $\wedge\ \delta$ baliokidetasuna erabili da, edozein formula δ-rentzat hori horrela baita. Bestalde, $0 \geq 0$ beti betetzen denez, *true* edukitzea bezala da eta ken daiteke $\delta \equiv \delta \wedge$ *true* baliokidetasuna erabiliz, edozein formula δ hartuta hori horrela baita. Bukatzeko, x^0 ezin daitekeela sinplifikatu konturatzea komeni da. x^0 espresioaren balioa 1 da 0^0 kasuan izan ezik, hau da $x = 0$ denean x^0 ez da 1, indeterminazioa da eta kasu horrek errorea sortzen du programetan.

φ eta φ_1 ez direnez berdinak, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu behar da:
$\varphi \rightarrow \varphi_1$?
$z = 1 \rightarrow z = x^0$?

Beraz, $z = 1$ baldin bada, $z = x^0$ betetzen al da?

Erantzuna ezezkoa da, hori betetzen ez duen kasu bat baitago, x aldagaiaren balioa 0 denean 0^0 gelditzen da eta hori indeterminazioa da.

Inplikazioa ez denez betetzen, programa ez da zuzena. Programa bat zuzena ez denean, zuzentasunaren froga eman beharrean, programa ez dela zuzena erakusten duen adibide bat eman behar da.

- **Programa zuzena ez dela erakusten duen adibidea**:

  x aldagaiaren balioa 0 denean φ hasierako baldintza bete egiten da baina esleipena burutu ondoren ψ bukaerako baldintza ez da betetzen.

3.2.2.10. Zazpigarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{x \neq 0 \wedge z = 1\}$
y : = 0;
$\{\psi\} \equiv \{z = x^y \wedge y \geq 0\}$

Orain honako bi urrats hauek jarraitu behar ditugu:
- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^t\}$ kalkulatu t-ren balioa 0 dela kontsideratuz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu eta berdinak ez badira, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu.

EA

$\{\varphi\} \equiv \{x \neq 0 \wedge z = 1\}$
$\{\varphi_1\} \equiv \{def(0) \wedge \psi_y^0\} \equiv \{true \wedge z = x^0 \wedge 0 \geq 0\} \equiv^{sinplifikazioa} \{z = x^0\}$
y : = 0;
$\{\psi\} \equiv \{z = x^y \wedge y \geq 0\}$

def(0) espresioaren balioa *true* da 0 beti kalkula daitekeelako, 0 kalkulatu ahal izateko ez da inolako baldintzarik bete behar aurretik.

{def(0) $\wedge$ ψ_x^0} kalkulatu ondoren sinplifikatu egin da ulertzeko errazagoa izan dadin. Sinplifikazioa egiteko alde batetik $\delta \equiv$ *true* $\wedge\ \delta$ baliokidetasuna erabili da, edozein formula δ-rentzat hori horrela baita. Bestalde, $0 \geq 0$ beti betetzen denez, *true* edukitzea bezala da eta ken daiteke $\delta \equiv \delta \wedge$ *true* baliokidetasuna erabiliz, edozein formula δ hartuta hori horrela baita. Bukatzeko, x^0 ezin daitekeela sinplifikatu konturatzea komeni da. x^0 espresioaren balioa 1 da 0^0 kasuan izan ezik, hau da $x = 0$ denean x^0 ez da 1, indeterminazioa da eta kasu horrek errorea sortzen du programetan.

φ eta φ_1 ez direnez berdinak, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu behar da:

$\varphi \rightarrow \varphi_1$?

$$\underbrace{x \neq 0}_{\alpha} \wedge \underbrace{z = 1}_{\beta} \rightarrow \underbrace{z = x^0}_{\text{bai, } \alpha \text{ eta } \beta\text{-gatik}}?$$

Beraz, $x \neq 0$ baldin bada eta $z = 1$ baldin bada, $z = x^0$ betetzen al da?

$x \neq 0 \wedge z = 1$ betetzen dela jakinda, $z = x^0$ ere betetzen dela ziurta dezakegu izan ere $x \neq 0$ izanda, x^0 beti 1 izango baita. Beraz erantzuna baiezkoa da eta ondorioz 7. adibideko programa zuzena da. Eskuineko aldeko espresioa egiazkoa dela ziurtatzeko ezkerreko aldeko zein zati behar diren letra grekoen bidez zehaztu da.

Programa zuzena denez zuzentasunaren froga eman behar da.

- **Zuzentasunaren froga:**

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

Zuzentasunaren frogan programa zuzena dela ziurtatzeko zein axioma eta erregela erabili diren adierazi behar da. Kasu honetan 3. urratsean programa osoa zuzena dela esaten da Ondorioaren Erregela erabiliz, izan ere aurretik $\varphi \rightarrow \varphi_1$ inplikazioa betetzen dela eta Esleipenaren Axiomagatik $\{\varphi_1\}$ y : = 0; $\{\psi\}$ programa zuzena dela ikusi baitugu.

Programa zuzena dela erabakitzeko honako hau egin da:

- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^t\}$ kalkulatu y : = 0; esleipena eta $\{\psi\}$ bukaerako baldintza erabiliz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu.
- Desberdinak direnez, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu
- Inplikazio hori bete egiten denez, programa zuzena da eta bukatzeko zuzentasunaren froga eman dugu.

3.2.2.11. Zortzigarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{b \wedge posit(A(1..i)) \wedge 1 \le i \le n - 1 \wedge A(i + 1) < 0\}$
b: = false;
$\{\psi\} \equiv \{b \leftrightarrow posit(A(1..i + 1))\}$

Bertan posit(A(1..n)) predikatua honela definituta dagoela kontsideratu behar da:

$$posit(A(1..n)) \equiv \forall k\ (1 \le k \le n \rightarrow A(k) > 0)$$

Orain honako bi urrats hauek jarraitu behar ditugu:

- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^t\}$ kalkulatu t-ren balioa 0 dela kontsideratuz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu eta berdinak ez badira, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu.

EA

$\{\varphi\} \equiv \{b \wedge posit(A(1..i)) \wedge 1 \le i \le n - 1 \wedge A(i + 1) < 0\}$
$\{\varphi_1\} \equiv \{def(false) \wedge \psi_b^{false}\} \equiv \{true \wedge (false \leftrightarrow posit(A(1..i + 1)))\}$
$\equiv^{sinplifikazioa} \{\neg posit(A(1..i + 1))\}$
b: = false;
$\{\psi\} \equiv \{b \leftrightarrow posit(A(1..i + 1))\}$

def(false) espresioaren balioa *true* da false konstante bat delako eta beti kalkula daitekeelako, false kalkulatu ahal izateko ez da inolako baldintzarik bete behar aurretik.

$\{def(false) \wedge \psi_x^{false}\}$ kalkulatu ondoren sinplifikatu egin da ulertzeko errazagoa izan dadin. Sinplifikazioa egiteko alde batetik $\delta \equiv$ *true* $\wedge\ \delta$ baliokidetasuna erabili da, edozein formula δ-rentzat hori horrela baita. Bestalde, $\neg\delta \equiv$ *false* $\leftrightarrow \delta$ baliokidetasuna erabili da, edozein formula δ hartuta baliokidetasun hori betetzen baita.

φ eta φ_1 ez direnez berdinak, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu behar da:

$\varphi \rightarrow \varphi_1$?

$$b \wedge posit(A(1..i)) \wedge \underbrace{1 \le i \le n - 1}_{\alpha} \wedge \underbrace{A(i + 1) < 0}_{\beta} \rightarrow \underbrace{\neg posit(A(1..i + 1))}_{\text{Bai, } \alpha \text{ eta } \beta\text{-gatik}}?$$

Beraz, $b \wedge posit(A(1..i)) \wedge 1 \le i \le n - 1 \wedge A(i + 1) < 0$ betetzen baldin bada, $\neg posit(A(1..i + 1))$ ere beteko al da?

α-gatik i + 1 espresioaren balioa A(1..n) bektorearen mugen barruan egongo da. β-gatik A(1..n) bektoreko i + 1 posizioan balio negatibo bat daukagu. Ondorioz, posit(A(1..i + 1)) ez da beteko, hau da, false izango da eta $\neg posit(A(1..i + 1))$ egiazkoa izango da:

Eskuineko aldeko espresioa egiazkoa dela ziurtatzeko ezkerreko aldeko zein zati behar diren letra grekoen bidez zehaztu da.

Programa zuzena denez zuzentasunaren froga eman behar da.

- **<u>Zuzentasunaren froga:</u>**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ b : = false; $\{\psi\}$ (EA)
3. $\{\varphi\}$ b : = false; $\{\psi\}$ (OE 1, 2)

Zuzentasunaren frogan programa zuzena dela ziurtatzeko zein axioma eta erregela erabili diren adierazi behar da. Kasu honetan 3. urratsean programa osoa zuzena dela esaten da Ondorioaren Erregela erabiliz, izan ere aurretik $\varphi \rightarrow \varphi_1$ inplikazioa betetzen dela eta Esleipenaren Axiomagatik $\{\varphi_1\}$ b : = false; $\{\psi\}$ programa zuzena dela ikusi baitugu.

Programa zuzena dela erabakitzeko honako hau egin da:

- $\{\varphi_1\} \equiv \{def(t) \wedge \psi_x^t\}$ kalkulatu b : = false; esleipena eta $\{\psi\}$ bukaerako baldintza erabiliz.
- $\{\varphi\}$ eta $\{\varphi_1\}$ berdinak al diren begiratu.
- Desberdinak direnez, $\varphi \rightarrow \varphi_1$ inplikazioa betetzen al den aztertu
- Inplikazio hori bete egiten denez, programa zuzena da eta bukatzeko zuzentasunaren froga eman dugu.

3.2.3. Konposizioaren erregela (KE)

Orain arte agindu bakarra, eta zehatzago izateko, esleipen bakarra duten programen zuzentasuna egiaztatzeko zer egin egin behar den ikusi dugu. Jarraian agindu bat baino gehiago dituzten programen zuzentasuna nola egiaztatzen den ikusiko dugu. Landuko ditugun adibideek eta ariketek bakarrik esleipenak izango dituzte.

Konposizioaren erregela (KE)
Bi azpiprograma edo aginduen konposizioa $$\frac{\{\varphi\}\ P_1\ \{\varphi_1\},\ \{\varphi_1\}\ P_2\ \{\psi\}}{\{\varphi\}\ P_1\ P_2\ \{\psi\}}$$ Erregela honen arabera, $\{\varphi\}$ P_1 $\{\varphi_1\}$ zuzena baldin bada eta $\{\varphi_1\}$ P_2 $\{\psi\}$ ere zuzena baldin bada, orduan $\{\varphi\}$ P_1 P_2 $\{\psi\}$ zuzena da.

Beste era batera esanda:

$\{\varphi\}$ P_1 $\{\varphi_1\}$	zuzena bada eta	$\{\varphi_1\}$ P_2 $\{\psi\}$	zuzena bada, orduan	$\{\varphi\}$ P_1 P_2 $\{\psi\}$	zuzena da.

Lehenengo programaren bukaerako baldintzak bigarren programaren hasierako baldintzaren berdina izan behar du. Kasu honetan $\{\varphi_1\}$.

3.2.3.1. Bederatzigarren adibidea

Honako programa hau zuzena al da?

	$\{\varphi\} \equiv \{x = a \land y = b\}$
P1	x : = x + y;
P2	y : = x – y;
P3	x : = x – y;
	$\{\psi\} \equiv \{x = b \land y = a\}$

Programa hori zuzena al den erabakitzeko hiru asertzio kalkulatu behar dira Esleipenaren Axioma erabiliz.

	$\{\varphi\} \equiv \{x = a \land y = b\}$
EA	$\{\varphi_3\} \equiv \{def(x + y) \land (\varphi_2)_x^{x+y}\}$
	x : = x + y;
EA	$\{\varphi_2\} \equiv \{def(x - y) \land (\varphi_1)_y^{x-y}\}$
	y : = x – y;
EA	$\{\varphi_1\} \equiv \{def(x - y) \land \psi_x^{x-y}\}$
	x : = x – y;
	$\{\psi\} \equiv \{x = b \land y = a\}$

Honako ekintza hauek burutu beharko dira:

- $\{\varphi_1\} \equiv \{def(x - y) \land \psi_x^{x-y}\}$ kalkulatu
- $\{\varphi_2\} \equiv \{def(x - y) \land (\varphi_1)_y^{x-y}\}$ kalkulatu
- $\{\varphi_3\} \equiv \{def(x + y) \land (\varphi_2)_x^{x+y}\}$ kalkulatu
- φ eta φ_3 berdinak al diren begiratu eta ez badira berdinak, $\varphi \to \varphi_3$ inplikazioa betetzen al den egiaztatu. Inplikazio hori betetzen bada, orduan programa zuzena da eta bestela ez da zuzena.
- Programa zuzena bada, zuzentasunaren froga eman beharko da eta programa zuzena ez bada, programa zuzena ez dela erakusten duen adibide bat eman beharko da.

$\{\varphi_1\}$, $\{\varphi_2\}$ eta $\{\varphi_3\}$ kalkulatzerakoan kalkuluak behetik gora egiten dira, ψ bukaerako baldintzatik abiatuta.

Jarraian $\{\varphi_1\}$, $\{\varphi_2\}$ eta $\{\varphi_3\}$ formulak emango dira eta kasu bakoitzean sinplifikatu egingo dira:

- $\{\varphi_1\} \equiv \{def(x - y) \land \psi_x^{x-y}\} \equiv \{true \land x - y = b \land y = a\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{x - y = b \land y = a\}$

  $\{\varphi_1\} \equiv \{x - y = b \land y = a\}$

- $\{\varphi_2\} \equiv \{def(x - y) \land (\varphi_1)_y^{x-y}\} \equiv \{true \land x - (x - y) = b \land x - y = a\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{y = b \land x - y = a\}$

$\{\varphi_2\} \equiv \{y = b \wedge x - y = a\}$

- $\{\varphi_3\} \equiv \{def(x + y) \wedge (\varphi_2)_x^{x+y}\} \equiv \{true \wedge y = b \wedge x + y - y = a\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{y = b \wedge x = a\}$

$\{\varphi_3\} \equiv \{ y = b \wedge x = a \}$

def(x – y) eta def(x + y) espresioak *true* dira, zenbaki osoentzat batuketa eta kenketa beti kalkula daitezkeelako, hau da, eragiketa horiek burutzeko ez dago aurretik baldintzarik bete beharrik.

Goraino iritsi ondoren, hau da, φ_3 kalkulatu ondoren, φ eta φ_3 ez direnez berdinak, φ_3 formulak dioena φ formulak dioenaren ondorioa al den aztertu beharko da:

$$\varphi \rightarrow \varphi_3?$$

$$\underbrace{x = a}_{\alpha} \wedge \underbrace{y = b}_{\beta} \rightarrow \underbrace{y = b}_{\beta\text{-gatik}} \wedge \underbrace{x = a}_{\alpha\text{-gatik}}?$$

Hau da, $x = a \wedge y = b$ betetzen bada, orduan $y = b \wedge x = a$ ere betetzen al da? Erantzuna baiezkoa da, kasu bietan gauza bera baina orden desberdinean daukagulako eta edozein bi formula δ eta γ emanda, honako baliokidetasun hau betetzen da beti: $\delta \wedge \gamma \equiv \gamma \wedge \delta$.

α eta β letra grekoen bidez eskuineko aldeko atal bakoitza betetzen dela ziurtatzeko ezkerreko aldeko zein atal erabili behar den zehazten da.

Implikazio hori bete egiten denez 9. adibideko programa zuzena da eta orain zuzentasunaren froga eman behar da.

- **<u>Zuzentasunaren froga:</u>**

1. $\varphi \rightarrow \varphi_3$
2. $\{\varphi_3\}$ x : = x + y; $\{\varphi_2\}$ (EA)
3. $\{\varphi\}$ x : = x + y; $\{\varphi_2\}$ (OE 1, 2)
4. $\{\varphi_2\}$ y : = x – y; $\{\varphi_1\}$ (EA)
5. $\{\varphi\}$
   x : = x + y;
   y : = x – y;
   $\{\varphi_1\}$ (KE 3, 4)
6. $\{\varphi_1\}$ x : = x – y; $\{\psi\}$ (EA)
7. $\{\varphi\}$
   x : = x + y;
   y : = x – y;
   x : = x – y;
   $\{\psi\}$ (KE 5, 6)

Zuzentasunaren frogan programa zuzena dela ziurtatzeko zein axioma eta erregela erabili diren adierazi behar da. Kasu honetan 7. urratsean programa osoa zuzena dela esaten da konposizioaren Erregela erabiliz. Horretarako 5. eta 6. puntuetako programak elkartu dira. Aurretik 5. eta 6. puntuetako programak zuzenak direla frogatu behar izan da. Seigarren puntuko programa zuzena dela zuzenean frogatu da Esleipenaren Axioma erabiliz. Bosgarren puntuko programa zuzena dela esateko Konposizioaren Erregela eta hirugarren eta laugarren puntuetako programak hartu dira kontuan. Laugarren puntuko programa zuzena dela frogatzeko Esleipenaren Axioma erabili da. Hirugarren puntuko programa zuzena dela frogatzeko Ondorioaren erregela, bigarren puntuko programa eta lehenengo puntuko inplikazioa hartu dira kontuan. Eta bigarren puntuko programa zuzena dela frogatzeko Esleipenaren axioma erabili da.

Programa zuzena dela erabakitzeko honako hau egin da:

- φ_1, φ_2 eta φ_3 kalkulatu ψ bukaerako baldintzatik abiatuta eta programan dauden esleipenak kontuan hartuz.
- $\{\varphi\}$ eta $\{\varphi_3\}$ berdinak al diren begiratu.
- Desberdinak direnez, $\varphi \rightarrow \varphi_3$ inplikazioa betetzen al den aztertu
- Inplikazio hori bete egiten denez, programa zuzena da eta bukatzeko zuzentasunaren froga eman dugu.

3.2.4. While-aren Erregela (WE)

Atal honetan **While** agindua duen programa baten zuzentasuna nola egiaztatzen den azalduko da.

While batez osatutako programa batek honako egitura izango du:

{φ} while {INB} B loop Aginduak end loop {ψ}	bertan φ hasierako baldintza da, ψ bukaerako baldintza da, INB inbariantea da, B while-aren baldintza da eta "Aginduak" while-aren barruko aginduak dira.

Honelako while programa bat emanda,

```
{φ}
while {INB} B loop
        Aginduak
end loop
{ψ}
```

Programa hori **partzialki zuzena** dela esango da honako lau puntu hauek betetzen badira:

I. $\varphi \rightarrow INB$ inplikazioa betetzen da
II. $INB \rightarrow def(B)$ inplikazioa betetzen da
III. Honako programa hau zuzena da

```
{INB ∧ B}
  Aginduak
 {INB}
```

IV. $(INB \wedge \neg B) \rightarrow \psi$ inplikazioa betetzen da

While programa hori **guztiz zuzena** izan dadin, aurreko lau puntu horietaz gain honako beste bi puntu hauek ere bete beharko ditu.

V. $(INB \wedge B) \rightarrow E > 0$ betetzen da
VI. Honako programa hau zuzena da

```
{INB ∧ B ∧ E = v}
  Aginduak
 {E < v}
```

v aldagaia programan agertzen ez den aldagai bat dela kontsideratuz.

Lehenengo lau puntuen bidez zuzentasun partziala frogatzen da, hau da, programa bukatzen bada, bukaerako baldintza betez bukatuko dela. Baina ez da frogatzen programa bukatuko denik.

Beste bi puntuen bidez (V eta VI puntuen bidez) while-a bukatu egingo dela frogatzen da.

Beraz, sei puntuak betetzen direla frogatzen bada, while-a beti bukatuko dela eta gainera bukaerako baldintza betez bukatuko dela frogatzen da. Kasu horretan programa guztiz zuzena dela esango genuke.

3.2.4.1. Hamargarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{i = 0 \wedge s = 0 \wedge n \geq 1\}$
while {INB} $i \neq n$ **loop**
    $i := i + 1;$
    $s := s + A(i);$
end loop;
$\{\psi\} \equiv \{s = \sum_{k=1}^{n} A(k)\}$

$\{INB\} \equiv \{(0 \leq i \leq n) \wedge s = \sum_{k=1}^{i} A(k)\}$

$E = n - i$

Urratsak:

- While-aren erregela aplikatzean honako kalkuluak eta egiaztapenak egin beharko dira:

I. $\varphi \rightarrow INB?$
II. $INB \rightarrow def(B)?$

III. $\{INB \wedge B\}$ $(INB \wedge B) \rightarrow \varphi_2?$
$\{\varphi_2\}$
(EA) i : = i + 1;
$\{\varphi_1\}$
(EA) s : = s + A(i);
$\{INB\}$

IV. $(INB \wedge \neg B) \rightarrow \psi?$
V. $(INB \wedge B) \rightarrow E > 0?$
VI. $\{INB \wedge B \wedge E = v\}$ $¿(INB \wedge B \wedge E = v) \rightarrow \varphi_4?$
$\{\varphi_4\}$
(EA) i : = i + 1;
$\{\varphi_3\}$
(EA) s : = s + A(i);
$\{E < v\}$

I. $\varphi \rightarrow INB?$

$$i = 0 \wedge s = 0 \wedge n \geq 0 \rightarrow (0 \leq i \leq n) \wedge s = \sum_{k=1}^{i} A(k)?$$

Hau da, $i = 0 \wedge s = 0 \wedge n \geq 0$ betetzen dela suposatuz, $(0 \leq i \leq n) \wedge s = \sum_{k=1}^{i} A(k)$ betetzen al den ala ez erabaki behar da.

$$\underbrace{i = 0}_{\alpha} \wedge \underbrace{s = 0}_{\beta} \wedge \underbrace{n \geq 1}_{\gamma} \rightarrow \underbrace{(0 \leq i \leq n)}_{\text{bai, } \alpha \text{ eta } \gamma\text{-gatik}} \wedge \underbrace{s = \sum_{k=1}^{i} A(k)}_{\text{bai, } \alpha \text{ eta } \beta\text{-gatik}}$$

Kasu honetan inplikazioa bete egiten da. Bigarren zatiko atal bakoitza bete egiten dela ziurta dezakegu lehenengo zatiko atalak kontuan hartuz, α, β eta γ letra grekoen bidez erakusten den bezala.

II. $INB \rightarrow def(B)?$
$INB \rightarrow true?$ Bai, inplikazioaren bigarren zatia true delako.

III.

- $\{\varphi_1\} \equiv \{\text{def}(s + A(i)) \wedge (INB)_s^{s + A(i)}\} \equiv$

  $\equiv \{(1 \le i \le n) \wedge (0 \le i \le n) \wedge s + A(i) = \sum_{k=1}^{i} A(k)\} \equiv$ sinplifikazioa

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

(1 ≤ i ≤ n) betetzen bada orduan (0 ≤ i ≤ n) ere beteko denez, (0 ≤ i ≤ n) ken daiteke (1 ≤ i ≤ n) lagaz. Baina kontuz, ezingo genuke (1 ≤ i ≤ n) kendu eta (0 ≤ i ≤ n) laga, (0 ≤ i ≤ n) betzen bada ere gerta daitekeelako (1 ≤ i ≤ n) ez betetzea.

- $\{\varphi_2\} \equiv \{\text{def}(i + 1) \wedge (\varphi_1)_i^{i+1}\} \equiv$

  $\equiv \{\text{true} \wedge (1 \le i + 1 \le n) \wedge s + A(i + 1) = \sum_{k=1}^{i+1} A(k)\} \equiv$ sinplifikazioa

  $\equiv \{(0 \le i \le n - 1) \wedge s + A(i + 1) = \sum_{k=1}^{i+1} A(k)\}$

Edozein formula δ hartuta, *true* ∧ δ ≡ δ baliokidetasuna betetzen denez, *true* ken daiteke.

Bestalde, (1 ≤ i + 1 ≤ n) formula i + 1 erabiliz ediku beharrean hobe da i erabiliz ipintzea. Horretarako 1 balioa kendu behar zaie kenketaren bidez hiru osagaiei: (1 – 1 ≤ i + 1 – 1 ≤ n – 1). Eragiketak burutu ondoren honako hau gelditukoda (0 ≤ i ≤ n – 1).

- $(INB \wedge B) \rightarrow \varphi_2$?

$$\underbrace{\mathbf{(0 \le i \le n)}}_{\alpha} \wedge \underbrace{\mathbf{s} = \sum_{k=1}^{i} A(k)}_{\beta} \wedge \underbrace{\mathbf{i \ne n}}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{\mathbf{(0 \le i \le n - 1)}}_{\alpha\text{ eta }\delta\text{-gatik}} \wedge \underbrace{\mathbf{s + A(i + 1)} = \sum_{k=1}^{i+1} A(k)}_{\alpha,\ \beta\text{ eta }\delta\text{-gatik}}$$

Beheko bigarren atalean α eta δ-gatik badakigu i + 1 indize bezala erabiliz ez garela A(1..n) bektorearen mugetatik aterako eta gero β-gatik "**s + A(i + 1) =** $\sum_{k=1}^{i+1} A(k)$" betetzen dela ziurta dezakegu.

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

$$\underbrace{\mathbf{(0 \le i \le n)}}_{\alpha} \wedge \mathbf{s} = \underbrace{\sum_{k=1}^{i} A(k)}_{\beta} \wedge \underbrace{\mathbf{i = n}}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{\{s = \sum_{k=1}^{n} A(k)\}}_{\beta \text{ eta } \delta\text{-gatik}}$$

i = n denez, sustituyendo i por n en β formulan i dagoen lekuan n ipiniz ψ lortzen da.

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

$$\underbrace{\mathbf{(0 \le i \le n)}}_{\alpha} \wedge \mathbf{s} = \sum_{k=1}^{i} A(k) \wedge \underbrace{\mathbf{i \ne n}}_{\beta}$$

$$\downarrow ?$$

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

α-gatik $n \ge i$ dela badakigu eta ondorioz $n - i \ge 0$ izango da, izan ere, $n \ge i$ betetzen bada, $n - i \ge i - i$ ere beteko da eta formula hori $n - i \ge 0$ formularen berdina da. Gainera β-gatik badakigu $\mathbf{i \ne n}$ dela. Beraz $n - i > 0$ betetzen da.

VI.

- $\{\varphi_3\} \equiv \{def(s + A(i)) \wedge (E < v)_s^{s + A(i)}\} \equiv$
  $\equiv \{(1 \le i \le n) \wedge n - i < v\} \equiv$

- $\{\varphi_4\} \equiv \{def(i + 1) \wedge (\varphi_3)_i^{i + 1}\} \equiv$
  $\equiv \{true \wedge (1 \le i + 1 \le n) \wedge n - (i + 1) < v\} \equiv$ sinplifikazioa
  $\equiv \{(0 \le i \le n - 1) \wedge n - i - 1 < v\}$

Edozein formula δ hartuta, *true* $\wedge\, \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.

Bestalde, $(1 \le i + 1 \le n)$ formula i + 1 erabiliz ediku beharrean hobe da i erabiliz ipintzea. Horretarako 1 balioa kendu behar zaie kenketaren bidez hiru osagaiei: $(1 - 1 \le i + 1 - 1 \le n - 1)$. Eragiketak burutu ondoren honako hau gelditurko da $(0 \le i \le n - 1)$.

- $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$?

$$\underbrace{\mathbf{(0 \le i \le n)}}_{\alpha} \wedge \mathbf{s} = \sum_{k=1}^{i} A(k) \wedge \underbrace{\mathbf{i \ne n}}_{\beta} \wedge \underbrace{\mathbf{n - i = v}}_{\delta}$$

$$\downarrow\mathbf{?}$$

$$\underbrace{\mathbf{(0 \le i \le n - 1)}}_{\alpha \text{ eta } \beta\text{-gatik}} \wedge \underbrace{\mathbf{n - i - 1 < v}}_{\delta\text{-gatik}}$$

Beheko lehenengo zatia, hau da, $(0 \le i \le n - 1)$, α eta β-gatik beteko da eta bigarren zatia, hau da, $n - i - 1 < v$, δ-gatik beteko da.

- **Zuzentasun osoaren froga:**

I
1. $\varphi \rightarrow INB$

II
2. $INB \rightarrow def(B)$

III
3. $(INB \wedge B) \rightarrow \varphi_2$
4. $\{\varphi_2\}$ i : = i + 1; $\{\varphi_1\}$ (AA)
5. $\{INB \wedge B\}$ i : = i + 1; $\{\varphi_1\}$ (OE 3, 4)
6. $\{\varphi_1\}$ s : = s + A(i); {INB} (EA)
7. $\{INB \wedge B\}$
   i : = i + 1;
   s : = s + A(i);
   {INB} (KE 5, 6)

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

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

VI
10. $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$
11. $\{\varphi_4\}$ i : = i + 1; $\{\varphi_3\}$ (EA)
12. $\{INB \wedge B \wedge E = v\}$ i : = i + 1; $\{\varphi_3\}$ (OE 10, 11)
13. $\{\varphi_3\}$ s : = s + A(i); $\{E < v\}$ (EA)
14. $\{INB \wedge B \wedge E = v\}$
    i : = i + 1;
    s : = s + A(i);
    $\{E < v\}$ (KE 12, 13)

15. $\{\varphi\}$
    while {INB} $i \ne n$ **loop**
        i : = i + 1;
        s : = s + A(i);
    end **loop**;
    $\{\psi\}$ (WE 1, 2, 7, 8, 9, 14)

3.2.4.2. Hamaikagarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{i = 0 \wedge s = 0 \wedge n \geq 1\}$
<u>while</u> {INB} $i \neq n$ **<u>loop</u>**
    $i := i + 1;$
    $s := s + A(i);$
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{s = \sum_{k=1}^{n} A(k)\}$

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

Programa honek aurreko programako adibidearekiko duen desberdintasun bakarra berdez ipinita dago.

<u>Urratsak:</u>

- While-aren erregela aplikatzean honako kalkuluak eta egiaztapenak egin beharko dira:

I. $\varphi \rightarrow INB$?
II. $INB \rightarrow def(B)$?

III. $\{INB \wedge B\}$ $(INB \wedge B) \rightarrow \varphi_2$?
  $\{\varphi_2\}$
  $i := i + 1;$
(EA)
  $\{\varphi_1\}$
  $s := s + A(i);$
(EA)
  $\{INB\}$

IV. $(INB \wedge \neg B) \rightarrow \psi$?
V. $(INB \wedge B) \rightarrow E > 0$?
VI. $\{INB \wedge B \wedge E = v\}$ $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$?
  $\{\varphi_4\}$
  $i := i + 1;$
(EA)
  $\{\varphi_3\}$
  $s := s + A(i);$
(EA)
  $\{E < v\}$

I. $\varphi \rightarrow$ INB?

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

$$\downarrow ?$$

$$(1 \leq \underbrace{i \leq n}_{\alpha \text{ eta } \beta\text{-gatik}}) \wedge s = \sum_{k=1}^{i} A(k)$$

φ formulak ez du inbariantea inplikatzen, izan ere gerta daiteke φ betetzea eta INB ez betetzea. Esate baterako φ formulak $i = 0$ betetzen dela dio eta INB formulak $1 \leq i \leq n$ betetzen dela dio, eta bi gauza horiek ezin dira batera bete. Beraz programa hau ez da zuzena eta ez dago prozesuarekin jarraitu beharrik.

3.2.4.3. Hamabigarren adibidea

Programa hau 10. adibideko programaren antzekoa da baina while-aren barruko aginduak alderantzizko ordenean daude.

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{i = 0 \wedge s = 0 \wedge n \geq 1\}$
while {INB} $i \neq n$ **loop**
    $s := s + A(i + 1);$
    $i := i + 1;$
end loop;
$\{\psi\} \equiv \{s = \sum_{k=1}^{n} A(k)\}$

$\{INB\} \equiv \{(0 \leq i \leq n) \wedge s = \sum_{k=1}^{i} A(k)\}$

$E = n - i$

Urratsak:

- While-aren erregela aplikatzean honako kalkuluak eta egiaztapenak egin beharko dira:

I. $\varphi \rightarrow INB$?
II. $INB \rightarrow def(B)$?

III. $\{INB \wedge B\}$ $(INB \wedge B) \rightarrow \varphi_2$?
$\{\varphi_2\}$
s : = s + A(i + 1);
(EA)
$\{\varphi_1\}$
i : = i + 1;
(EA)
$\{INB\}$

IV. $(INB \wedge \neg B) \rightarrow \psi$?
V. $(INB \wedge B) \rightarrow E > 0$?
VI. $\{INB \wedge B \wedge E = v\}$ $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$?
$\{\varphi_4\}$
s : = s + A(i + 1);
(EA)
$\{\varphi_3\}$
i : = i + 1;
(EA)
$\{E < v\}$

I. $\varphi \rightarrow INB$?

$$i = 0 \wedge s = 0 \wedge n \geq 0 \rightarrow (0 \leq i \leq n) \wedge s = \sum_{k=1}^{i} A(k)\,?$$

$$\underbrace{i = 0}_{\alpha} \wedge \underbrace{s = 0}_{\beta} \wedge \underbrace{n \geq 1}_{\gamma} \rightarrow \underbrace{(0 \leq i \leq n)}_{\alpha \text{ eta } \gamma\text{-gatik}} \wedge \underbrace{s = \sum_{k=1}^{i} A(k)}_{\alpha \text{ eta } \beta\text{-gatik}}$$

Inplikazio hori bete egiten da eta α, β eta γ letra grekoen bidez eskuineko aldeko atal bakoitza betetzen dela ziurtatzeko ezkerreko aldeko zein atal erabili behar den zehazten da.

II. $INB \rightarrow def(B)$?
$INB \rightarrow true$? Bai, inplikazioko bigarren zatia true delako.

III.

- $\{\varphi_1\} \equiv \{\mathrm{def}(i + 1) \wedge (\mathrm{INB})_i^{i+1}\} \equiv$

  $\equiv \{\mathrm{true} \wedge (0 \leq i + 1 \leq n) \wedge s = \sum_{k=1}^{i+1} A(k)\} \equiv$ sinplifikazioa

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

Edozein formula δ hartuta, *true* $\wedge\ \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.

Bestalde, $(0 \leq i + 1 \leq n)$ formula $i + 1$ erabiliz ediku beharrean hobe da i erabiliz ipintzea. Horretarako 1 balioa kendu behar zaie kenketaren bidez hiru osagaiei: $(0 - 1 \leq i + 1 - 1 \leq n - 1)$. Eragiketak burutu ondoren honako hau geldituko da $(-1 \leq i \leq n - 1)$.

- $\{\varphi_2\} \equiv \{\mathrm{def}(s + A(i + 1)) \wedge (\varphi_1)_s^{s + A(i)}\} \equiv$

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

sinplifikazioa $\equiv \{(0 \leq i \leq n - 1) \wedge s + A(i + 1) = \sum_{k=1}^{i+1} A(k)\}$

Hasteko $(1 \leq i + 1 \leq n)$ espresioa $(0 \leq i \leq n - 1)$ bezala ipin daiteke. Gero $(-1 \leq i \leq n - 1)$ ken daiteke bakarrik $(0 \leq i \leq n - 1)$ lagaz, $(0 \leq i \leq n - 1)$ betetzen bada, $(-1 \leq i \leq n - 1)$ ere beteko delako. Baina kontuz, ezingo genuke $(0 \leq i \leq n - 1)$ kendu eta $(-1 \leq i \leq n - 1)$ laga, $(-1 \leq i \leq n - 1)$ betetzeak ez duelako ziurtatzen $(0 \leq i \leq n - 1)$ beteko denik.

- $(\mathrm{INB} \wedge B) \rightarrow \varphi_2$?

$$\underbrace{\mathbf{(0 \leq i \leq n)}}_{\alpha} \wedge \underbrace{\mathbf{s} = \sum_{k=1}^{i} A(k)}_{\beta} \wedge \underbrace{\mathbf{i \neq n}}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{(0 \leq i \leq n - 1)}_{\alpha \text{ eta } \delta\text{-gatik}} \wedge \underbrace{s + A(i + 1) = \sum_{k=1}^{i+1} A(k)}_{\alpha,\ \beta \text{ eta } \delta\text{-gatik}}$$

Beheko bigarren atalean α eta δ-gatik badakigu $i + 1$ indize bezala erabiliz ez garela $A(1..n)$ bektorearen mugetatik aterako eta gero β-gatik "$\mathbf{s + A(i + 1)} = \sum_{k=1}^{i+1} A(k)$" betetzen dela ziurta dezakegu.

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

$$\underbrace{\mathbf{(0 \le i \le n)}}_{\alpha} \wedge \underbrace{\mathbf{s =} \sum_{k=1}^{i} A(k)}_{\beta} \wedge \underbrace{\mathbf{i = n}}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{\{s = \sum_{k=1}^{n} A(k)\}}_{\beta \text{ eta } \delta\text{-gatik}}$$

i = n denez, sustituyendo i por n en β formulan i dagoen lekuan n ipiniz ψ lortzen da.

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

$$\underbrace{\mathbf{(0 \le i \le n)}}_{\alpha} \wedge \mathbf{s =} \sum_{k=1}^{i} A(k) \wedge \underbrace{\mathbf{i \ne n}}_{\beta}$$

$$\downarrow ?$$

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

α-gatik n ≥ i dela badakigu eta ondorioz n – i ≥ 0 izango da, izan ere, n ≥ i betetzen bada, n – i ≥ i – i ere beteko da eta formula hori n – i ≥ 0 formularen berdina da. Gainera β-gatik badakigu **i ≠ n** dela. Beraz n – i > 0 betetzen da.

VI.

- $\{\varphi_3\} \equiv \{\text{def}(i + 1) \wedge (E < v)_i^{i+1}\} \equiv$
  $\equiv \{\text{true} \wedge n - (i + 1) < v\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{n - i - 1 < v\}$

Edozein formula δ hartuta, *true* ∧ δ ≡ δ baliokidetasuna betetzen denez, *true* ken daiteke.

- $\{\varphi_4\} \equiv \{\text{def}(s + A(i + 1)) \wedge (\varphi_3)_s^{s + A(i)}\} \equiv$
  $\equiv \{(1 \le i + 1 \le n) \wedge n - i - 1 < v\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{(0 \le i \le n - 1) \wedge n - i - 1 < v\}$

Bestalde, $(1 \le i + 1 \le n)$ formula $i + 1$ erabiliz ediku beharrean hobe da i erabiliz ipintzea. Horretarako 1 balioa kendu behar zaie kenketaren bidez hiru osagaiei: $(1 - 1 \le i + 1 - 1 \le n - 1)$. Eragiketak burutu ondoren honako hau geldituko da $(0 \le i \le n - 1)$.

- $(INB \wedge B \wedge E = v) \to \varphi_4$?

$$\underbrace{\mathbf{(0 \le i \le n)}}_{\alpha} \wedge \mathbf{s} = \sum_{k=1}^{i} A(k) \wedge \underbrace{\mathbf{i \ne n}}_{\beta} \wedge \underbrace{\mathbf{n - i = v}}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{\mathbf{(0 \le i \le n - 1)}}_{\alpha \text{ eta } \beta\text{-gatik}} \wedge \underbrace{\mathbf{n - i - 1 < v}}_{\delta\text{-gatik}}$$

Beheko lehenengo zatia, hau da, $(0 \le i \le n - 1)$, α eta β-gatik beteko da eta bigarren zatia, hau da, $n - i - 1 < v$, δ-gatik beteko da.

- **Zuzentasun osoaren froga:**

I

1. $\varphi \rightarrow INB$

II

2. $INB \rightarrow def(B)$

III

3. $(INB \wedge B) \rightarrow \varphi_2$
4. $\{\varphi_2\}$ s : = s + A(i + 1); $\{\varphi_1\}$ (EA)
5. $\{INB \wedge B\}$ s : = s + A(i + 1); $\{\varphi_1\}$ (OE 3, 4)
6. $\{\varphi_1\}$ i : = i + 1; {INB} (EA)
7. $\{INB \wedge B\}$
   s : = s + A(i + 1);
   i : = i + 1;
   {INB} (KE 5, 6)

IV

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

V

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

VI

10. $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$
11. $\{\varphi_4\}$ s : = s + A(i + 1); $\{\varphi_3\}$ (EA)
12. $\{INB \wedge B \wedge E = v\}$ s : = s + A(i + 1); $\{\varphi_3\}$ (OE 10, 11)
13. $\{\varphi_3\}$ i : = i + 1; $\{E < v\}$ (EA)
14. $\{INB \wedge B \wedge E = v\}$
    s : = s + A(i + 1);
    i : = i + 1;
    $\{E < v\}$ (KE 12, 13)

15. $\{\varphi\}$
    while {INB} $i \neq n$ **loop**
      s : = s + A(i + 1);
      i : = i + 1;
    end loop;
    $\{\psi\}$ (WE 1, 2, 7, 8, 9, 14)

3.2.4.4. Hamairugarren adibidea

Programa hau 10. eta 12. adibideetako programen antzekoa da baina while-aren barruko aginduetan, hasierako baldintzan eta inbariantean desberdintasun txiki batzuk daude

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{i = 1 \wedge n \geq 1\}$
s : = 0;
while {INB} $i \neq n + 1$ **loop**
    s : = s + A(i);
    i : = i + 1;
end loop;
$\{\psi\} \equiv \{s = \sum_{k=1}^{n} A(k)\}$

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

$E = n + 1 - i$

Urratsak:

- While aginduaren aurretik esleipen bat dagoenez, bi azpiprograma bereiztu behar dira:

$\{\varphi\}$
s : = 0;
{INB}

eta

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

➢ Hasteko lehenengo azpiprograma zuzena al den egiaztatuko da

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

$\{\varphi_1\}$

s : = 0;

(EA)

$\{INB\}$

- $\{\varphi_1\} \equiv \{def(0) \land (INB)_s^0\} \equiv$

  $\equiv \{True \land (1 \leq i \leq n + 1) \land 0 = \sum_{k=1}^{i} A(k)\} \equiv$ sinplifikazioa

  $\equiv \{(1 \leq i \leq n + 1) \land 0 = \sum_{k=1}^{i} A(k)\}$

- $\varphi \rightarrow \varphi_1?$

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

$$\downarrow ?$$

$$\underbrace{(1 \leq i \leq n + 1)}_{\alpha \text{ eta } \beta\text{-gatik}} \land \underbrace{0 = \sum_{k=1}^{i-1} A(k)}_{\alpha\text{-gatik}}$$

Beheko bigarren atalean α-gatik badakigu i aldagaiaren balioa 1 izango dela eta ondorioz batukariaren balioa 0 izango da k 1etik 0ra doalako eta batukari batean beheko muga goikoa baino handiagoa denean batukariaren balioa 0 izaten da.

➢ While-aren erregela bigarren azpiprogramari aplikatzean honako kalkuluak eta egiaztapenak egin beharko dira:

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

I. $INB \rightarrow INB?$

II. $INB \rightarrow def(B)?$

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

$\{\varphi_3\}$

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

$\{\varphi_2\}$

(EA) $i := i + 1;$

$\{INB\}$

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

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

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

$\{\varphi_5\}$

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

$\{\varphi_4\}$

(EA) $i := i + 1;$

$\{E < v\}$

I. $INB \rightarrow INB?$

Inplikazio hori bete egiten da alde bietan gauza bera daukagulako.

II. $INB \rightarrow def(B)?$

$INB \rightarrow true?$ Bai, bigarren zatian true daukagulako.

III.

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

$$\equiv \{true \wedge (1 \le i + 1 \le n + 1) \wedge s = \sum_{k=1}^{i+1-1} A(k)\} \equiv^{\text{sinplifikazioa}}$$

$$\equiv \{(0 \le i \le n) \wedge s = \sum_{k=1}^{i} A(k)\}$$

Edozein formula δ hartuta, *true* $\wedge\ \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.

Bestalde, $(1 \le i + 1 \le n)$ formula $i + 1$ erabiliz ediku beharrean hobe da i erabiliz ipintzea. Horretarako 1 balioa kendu behar zaie kenketaren bidez hiru osagaiei: $(1 - 1 \le i + 1 - 1 \le n - 1)$. Eragiketak burutu ondoren honako hau geldituko da $(0 \le i \le n - 1)$.

- $\{\varphi_3\} \equiv \{def(s + A(i)) \wedge (\varphi_2)_s^{s + A(i)}\} \equiv$

  $\equiv \{(1 \le i \le n) \wedge (0 \le i \le n) \wedge s + A(i) = \sum_{k=1}^{i} A(k)\} \equiv$ sinplifikazioa

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

(0 ≤ i ≤ n) ken daiteke bakarrik (1 ≤ i ≤ n) lagaz, izan ere (1 ≤ i ≤ n) betetzen bada, (0 ≤ i ≤ n) ere beteko baita. Baina ezingo genuke (1 ≤ i ≤ n) kendu eta (0 ≤ i ≤ n) laga, (0 ≤ i ≤ n) betetzeak ez baitu ziurtatzen (1 ≤ i ≤ n) beteko denik.

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

$$\underbrace{\mathbf{(1 \le i \le n + 1)}}_{\alpha} \wedge \underbrace{\mathbf{s} = \sum_{k=1}^{i-1} A(k)}_{\beta} \wedge \underbrace{\mathbf{i \ne n + 1}}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{(1 \le i \le n)}_{\alpha \text{ eta } \delta\text{-gatik}} \wedge \underbrace{s + A(i) = \sum_{k=1}^{i} A(k)}_{\alpha,\ \beta \text{ eta } \delta\text{-gatik}}$$

Beheko bigarren atalean badakigu α eta δ-gatik i ez dela A(1..n) bektorearen mugetatik kanpo egongo eta gero β-gatik "$\mathbf{s + A(i)} = \sum_{k=1}^{i} A(k)$" beteko dela ziurta dezakegu.

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

$$\underbrace{\mathbf{(1 \le i \le n + 1)}}_{\alpha} \wedge \underbrace{\mathbf{s} = \sum_{k=1}^{i-1} A(k)}_{\beta} \wedge \underbrace{\mathbf{i = n + 1}}_{\delta}$$

$$\downarrow ?$$

$$\underbrace{\{s = \sum_{k=1}^{n} A(k)\}}_{\beta \text{ eta } \delta\text{-gatik}}$$

i = n + 1 denez, β formulan i-ren ordez n + 1 ipiniz ψ formula lortuko da.

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

$$\underbrace{\mathbf{(1 \le i \le n + 1)}}_{\alpha} \wedge \mathbf{s} = \sum_{k=1}^{i-1} A(k) \wedge \underbrace{\mathbf{i \ne n + 1}}_{\beta}$$

$$\downarrow ?$$

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

α-gatik badakigu $n + 1 \ge i$ dela eta obdorioz $n + 1 - i \ge 0$ izango da, izan ere $n + 1 \ge i$ betetzen bada, $n + 1 - i \ge i - i$ ere beteko da eta hori $n + 1 - i \ge 0$ da. Gainera β-gatik badakigu $\mathbf{i \ne n + 1}$ dela eta ondorioz $n + 1 - i > 0$ dela ziurta dezakegu.

VI.

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

Edozein formula δ hartuta, *true* $\wedge\ \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.

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

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

$$\underbrace{\mathbf{(1 \le i \le n + 1)}}_{\alpha} \wedge \mathbf{s} = \sum_{k=1}^{i-1} A(k) \wedge \underbrace{\mathbf{i \ne n + 1}}_{\beta} \wedge \underbrace{\mathbf{n + 1 - i = v}}_{\delta}$$

$$\downarrow ?$$

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

Beheko lehenengo atalean α eta β-gatik $1 \le i \le n$ beteko dela ziurta dezakegu eta beheko bigarren atalean δ-gatik $n - i = v - 1$ betetzen dela baiezta dezakegu $n + 1 - i = v$ berdintzako alde bietan 1 kenduz, eta ondorioz $n - i < v$ beteko da $n - i = v - 1$ baita.

- **Zuzentasun osoaren froga:**

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

I

4. INB → INB

II

5. INB → def(B)

III

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

IV

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

V

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

VI

13. $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$
14. $\{\varphi_4\}$ s : = s + A(i); $\{\varphi_3\}$ (EA)
15. $\{INB \wedge B \wedge E = v\}$ s : = s + A(i); $\{\varphi_3\}$ (OE 13, 14)
16. $\{\varphi_3\}$ i : = i + 1; $\{E < v\}$ (EA)
17. $\{INB \wedge B \wedge E = v\}$
    s : = s + A(i);
    i : = i + 1;
    $\{E < v\}$ (KE 15, 16)

18. {INB}
    while {INB} $i \neq n + 1$ **loop**
        s : = s + A(i);
        i : = i + 1;
    end loop;
    $\{\psi\}$ (WE 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
    s : = 0;
    while {INB} $i \neq n + 1$ **loop**
        s : = s + A(i);
        i : = i + 1;
    end loop;
    $\{\psi\}$ (KE 3, 18)

3.2.4.5. Hamalaugarren adibidea

Espezifikazioaren arabera ord aldagai boolearrean A(1..n) bektoreko elementuak goranzko ordenean al dauden, hau da, elementu bakoitza bere ondorengoa baino txikiagoa edo berdina al den erabaki beharko lukeen programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz:

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

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

$E = n - i$

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

Urratsak:

- While aginduaren aurretik esleipen bat dagoenez, bi azpiprograma bereiztu behar dira:

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

eta

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

➢ Hasteko lehenengo azpiprograma zuzena al den egiaztatuko da

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

$\{\varphi_1\}$

ord : = true;

(EA)

$\{INB\}$

- $\{\varphi_1\} \equiv \{def(true) \wedge (INB)_{ord}^{true}\} \equiv$

  $\equiv \{true \wedge (1 \leq i \leq n) \wedge (true \leftrightarrow gorakorra(A(1..i)))\} \equiv^{sinplifikazioa}$

  $\equiv \{(1 \leq i \leq n) \wedge gorakorra(A(1..i))\}$

Edozein formula δ hartuta, *true* $\wedge\ \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.Bestalde *true* $\leftrightarrow\ \delta \equiv \delta$ baliokidetasuna kontuan hartuz inplikazio bikoitza ere ken daiteke.

- $\varphi \rightarrow \varphi_1?$

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

$$\downarrow\mathbf{?}$$

$$\underbrace{(1 \leq i \leq n)}_{\alpha \text{ eta } \beta\text{-gatik}} \wedge \underbrace{gorakorra(A(1..i))}_{\beta\text{-gatik}}$$

i aldagaiaren balioa 1 denean, gorakorra(A(1..i)) formula eremu hutsa daukagu eta formula true da, hau da, bete egiten da.

- Bigarren azpiprogramari while-aren erregela aplikatu beharko zaio hasierako baldintza INB inbariantea dela kontsideratuz. While-aren erregela bigarren azpiprogramari aplikatzean honako kalkuluak eta egiaztapenak egin beharko dira:

I. $INB \to INV?$

II. $INB \to def(B)?$

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

$\{\varphi_3\}$

(EA) $ord := (A(i) \leq A(i + 1));$

$\{\varphi_2\}$

(EA) $i := i + 1;$

$\{INB\}$

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

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

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

$\{\varphi_5\}$

(EA) $ord := (A(i) \leq A(i + 1));$

$\{\varphi_4\}$

(EA) $i := i + 1;$

$\{E < v\}$

I. $INB \to INB?$
Inplikazio hori bete egiten da alde bietan gauza bera daukagulako.

II. $INB \to def(B)?$
$INB \to true?$ Bai, inplikazioaren bigarren zatia true delako.

III.

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

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

  Lehenengo sinplifikazioan errepikatuta zegoen tarte bat kendu da. Bigarren sinplifikazioan $(1 \le i + 1 \le n)$ tartea eraldatu egin da hiru osagaiei 1 kenduz eta $(0 \le i \le n - 1)$ tartea gelditu da. Hirugarren sinplifikazioan bi tarte desberdin ditugu i aldagaiarentzat eta beheko mugetako handienarekin (1 eta 0ren arteko handienarekin) eta goiko mugetako txikienarekin (n eta n – 1 balioen arteko txikienarekin) gelditu behar dugu. Beraz muga berriak 1 eta n – 1 dira eta tarte berria $(1 \le i \le n - 1)$.

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

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

$$\downarrow ?$$

$$\underbrace{(1 \le i \le n - 1)}_{\alpha \text{ eta } \gamma\text{-gatik}} \wedge \underbrace{(A(i) \le A(i + 1) \leftrightarrow gorakorra(A(1..i + 1)))}_{\substack{\pi \\ \beta \text{ eta } \delta\text{-gatik}}}$$

α eta γ-gatik $1 \le i \le n - 1$ dela ondoriozta dezakegu. φ_3 formulako inplikazio bikoitzaren bidez (hau da π-ren bidez) gorakorra(A(1..i + 1)) betetzea $A(i) \le A(i + 1)$ betetzearen menpe al dagoen galdetzen da. Inplikazio bikoitz hori betetzen al den jakiteko, honako bi inplikazio hauek betetzen al diren azter dezakegu:

✓ $(A(i) \le A(i + 1) \rightarrow gorakorra(A(1..i + 1)))$?
Erantzuna baiezkoa da, δ-gatik bai baitakigu ord aldagaiaren balioa *true* dela eta horregatik eta β-gatik gorakorra(A(1..i)) betetzen dela badakigu. Ondorioz $A(i) \le A(i + 1)$ betetzen al den jakitea falta da gorakorra(A(1..i + 1)) betetzen al den jakiteko.

✓ $gorakorra(A(1..i + 1))) \rightarrow (A(i) \le A(i + 1)$?

gorakorra(A(1..i + 1))) betetzen bada orduan badakigu 1..i + 1 tarteko elementu denak ordenatuta daudela eta ondorioz i eta i + 1 posiziokoak ere bai, beraz (A(i) ≤ A(i + 1) bete egiten da.

IV. (INB $\wedge$ ¬B) $\rightarrow$ ψ?

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

$$\downarrow ?$$

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

Disjuntzio bat daukagunez, (i = n $\vee$ ¬ord) formularen balioa true izateko hiru aukerak hartu behar dira kontuan:

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

- ✓ Lehenengo bietan i = n denez, β eta ψ berdinak dira eta inplkazioa bete egiten da.
- ✓ Hirugarrengoan i ≠ n denez eta ord = False denez, α-gatik i < n dela ziurta dezakegu eta β-gatik ¬gorakorra(A(1..i)) betetzen dela ziurta dezakegu. Beraz, gorakorra(A(1..i)) predikatuaren balioa False da. Eta gorakorra(A(1..i)) False izanda eta i < n izanda, gorakorra(A(1..n)) ere False izango da. Ondorioz ψ formulan False ↔ False gelditzen zaigu eta horrek esan nahi du ψ True dela.

Beraz inplikazioa bete egiten da.

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

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

$$\downarrow ?$$

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

VI.

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

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

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

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

$$\downarrow ?$$

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

- **Zuzentasun osoaren froga:**

1. $\varphi \rightarrow \varphi_1$
2. $\{\varphi_1\}$ ord : = true; {INB} (AA)
3. $\{\varphi\}$ ord : = true; {INB} (OE 1, 2)

I

4. INB $\rightarrow$ INB

II

5. INB $\rightarrow$ def(B)

III

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

IV

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

V

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

VI

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

18. {INB}
    while {INB} i $\neq$ n **and** ord **loop**
    ord : = (A(i) $\leq$ A(i + 1));
    i : = i + 1;
    end loop;
    $\{\psi\}$ (WE 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
    ord : = true;
    while {INB} i $\neq$ n **and** ord **loop**
    ord : = (A(i) $\leq$ A(i + 1));
    i : = i + 1;
    end loop;
    $\{\psi\}$ (KE 3, 18)

3.2.4.6. Hamabosgarren adibidea

Espezifikazioaren arabera hond aldagai boolearrean R(1..n) bektoreko posizio bakoitzean A(1..n) bektoreko elementua 2 zenbakiaz zatitzean lortzen den hondarra al dagoen erabaki beharko lukeen programa hau guztiz zuzena al den egiaztatu Hoare-ren kalkulua erabiliz

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

$\{INB\} \equiv \{(0 \leq i \leq n) \wedge (hond \leftrightarrow hondarrak(A(1..i), R(1..i)))\}$

$E = n - i$

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

Urratsak:

- While aginduaren aurretik esleipen bat dagoenez, bi azpiprograma bereiztu behar dira:

{φ}
i : = 0;
{INB}

y

{INB}
while {INB} i ≠ n **and** hond **loop**
    hond : = (A(i + 1) **mod** 2 = R(i + 1));
    i : = i + 1;
end loop;
{ψ}

➢ Hasteko lehenengo azpiprograma zuzena al den egiaztatuko da

{φ} $\varphi \rightarrow \varphi_1$?
$\{\varphi_1\}$
i : = 0;
(EA)
{INB}

- $\{\varphi_1\} \equiv \{def(0) \wedge (INB)_i^0\} \equiv$
  $\equiv \{true \wedge (0 \leq 0 \leq n) \wedge (hond \leftrightarrow hodarrak(A(1..0), R(1..0)))\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{(0 \leq n) \wedge (hond \leftrightarrow hondarrak(A(1..0), R(1..0)))\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{(0 \leq n) \wedge (hond \leftrightarrow true\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{(0 \leq n) \wedge hond\}$

hondarra(A(1..0), R(1..0)) predikatuan eremua hutsa denez true gelditzen da eta horregatik egin da bigarren sinplifikazioa.
Sinplifikazio prozesuan *true* $\wedge\ \delta \equiv \delta$ eta *true* $\leftrightarrow \delta \equiv \delta$ baliokidetasunak hartu dira kontuan.

- $\varphi \rightarrow \varphi_1$?

$$\{\underbrace{n \geq 1}_{\alpha} \wedge \underbrace{hond}_{\beta}\}$$
$$\downarrow ?$$
$$\underbrace{(0 \leq n)}_{\alpha\text{-gatik}} \wedge \underbrace{hond}_{\beta\text{-gatik}}$$

➢ Bigarren azpiprogramari while-aren erregela aplikatu beharko zaio hasierako baldintza INB inbariantea dela kontsideratuz. While-aren erregela bigarren azpiprogramari aplikatzean honako kalkuluak eta egiaztapenak egin beharko dira:

I. $INB \rightarrow INB$
II. $INB \rightarrow def(B)$?

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

$\{\varphi_3\}$
hond : = (A(i + 1) **mod** 2 = R(i + 1));
(EA)
$\{\varphi_2\}$
i : = i + 1;
(EA)
$\{INB\}$

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

$\{\varphi_5\}$
hond : = (A(i + 1) **mod** 2 = R(i + 1));
(EA)
$\{\varphi_4\}$
i : = i + 1;
(EA)
$\{E < v\}$

I. INB → INB? Inplikazio hau bete egiten da alde bietan gauza bera daukagulako.

II. INB → def(B)?

INB → true? Bai, inplikazioaren bigarren zatia true delako.

III.

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

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

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

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

$$\downarrow \mathbf{?}$$

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

β eta δ-gatik **hondarrak(A(1..i), R(1..i)))** betetzen dela ondoriozta dezakegu. φ_3 formulako inplikazio bikoitzean **hondarrak(A(1..i + 1), R(1..i + 1))** betetzea A(i + 1) **mod** 2 = R(i + 1) betetzearen menpe al dagoen galdetzen da. Erantzuna baiezkoa da bai baitakigu **hondarrak(A(1..i), R(1..i)))** bete egiten dela eta ondorioz A(i + 1) **mod** 2 = R(i + 1) betetzen al den jakitea falta da **hondarra(A(1..i + 1), R(1..i + 1))** betetzen al den jakiteko.

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

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

$$\downarrow \mathbf{?}$$

$$hond \leftrightarrow hondarrak(A(1..n), R(1..n))$$

Disjuntzio bat daukagunez, $(i = n \lor \neg hond)$ formula True izateko hiru aukerak aztertu behar dira:

i = n	**¬hond**
True	True
True	False
False	True

- Lehenengo bietan $i = n$ denez, β eta ψ formulak berdinak dira eta inplikazioa bete egiten da.
- Hirugarren kasuan $i \ne n$ eta hond = False dira. Alde batetik, $i \ne n$ denez $i < n$ dela ondoriozta dezakegu eta beste aldetik, h = False denez, ¬hondarrak(A(1..i), R(1..i)) betetzen dela ondoriozta dezakegu. Beraz hondarrak(A(1..i), R(1..i)) predikatuaren balioa False da. Beraz, hondarrak(A(1..i), R(1..i)) False dela jakinda eta $i <$ n dela jakinda, hondarra(A(1..n), R(1..n)) ere False dela baiezta dezakegu. Eta ψ formulan False $\leftrightarrow$ False gelditzen zaigu eta ondorioz ψ formula bete egiten da, True da.

Beraz inplikazioa bete egiten da.

V. $(INB \land B) \to E > 0?$

$$\{\underbrace{(0 \le i \le n)}_{\alpha} \land \underbrace{(hond \leftrightarrow hondarrak(A(1..i), R(1..i)))}_{\beta} \land \underbrace{i \ne n}_{\gamma} \land \underbrace{hond}_{\delta}\}$$

$$\downarrow ?$$

$$n - i > 0$$

α eta γ-gatik

VI.

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

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

- $(INB \land B \land E = v) \to \varphi_5?$

$$\{\underbrace{(0 \le i \le n)}_{\alpha} \land \underbrace{(hond \leftrightarrow hondarrak(A(1..i), R(1..i)))}_{\beta} \land \underbrace{i \ne n}_{\gamma} \land \underbrace{hond}_{\delta} \land \underbrace{(n - i = v)}_{\lambda}\}$$

$$\downarrow ?$$

$$\underbrace{(0 \le i \le n - 1)}_{\alpha \text{ eta } \gamma\text{-gatik}} \land \underbrace{(n - i - 1 < v)}_{\lambda\text{-gatik}}$$

- **<u>Zuzentasun osoaren froga:</u>**

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

I

4. INB → INB

II

5. INB → def(B)

III

6. $(INB \wedge B) \rightarrow \varphi_3$
7. $\{\varphi_3\}$ hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1)); $\{\varphi_2\}$ (EA)
8. $\{INB \wedge B\}$ hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1)); $\{\varphi_2\}$ (OE 6, 7)
9. $\{\varphi_2\}$ i : = i + 1; {INB} (AA)
10. $\{INB \wedge B\}$
    hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1));
    i : = i + 1;
    {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. $\{\varphi_5\}$ hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1)); $\{\varphi_4\}$ (EA)
15. $\{INB \wedge B \wedge E = v\}$ hond h : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1)); $\{\varphi_4\}$
    (OE 13, 14)
16. $\{\varphi_4\}$ i : = i + 1; $\{E < v\}$ (EA)
17. $\{INB \wedge B \wedge E = v\}$
    hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1));
    i : = i + 1;
    $\{E < v\}$ (KE 15, 16)

18. {INB}
    <u>while</u> {INB} $i \neq n$ **<u>and</u>** h **<u>loop</u>**
        hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1));
        i : = i + 1;
    <u>end</u> **<u>loop</u>**;
    $\{\psi\}$ (WE 4, 5, 10, 11, 12, 17)

19. $\{\varphi\}$
    i : = 0;
    <u>while</u> {INB} $i \neq n$ **<u>and</u>** h **<u>loop</u>**
        hond : = (A(i + 1) **<u>mod</u>** 2 = R(i + 1));
        i : = i + 1;
    <u>end</u> **<u>loop</u>**;
    $\{\psi\}$ (KE 3, 18)

3.2.4.7. Hamaseigarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{true\}$ **<u>while</u>** $\{INB\}$ $i \neq 0$ **<u>loop</u>** $i := i - 2;$ **<u>end</u>** **<u>loop</u>**; $\{\psi\} \equiv \{i = 0\}$
$\{INB\} \equiv \{(i \geq 0) \wedge bikoitia(i)\}$ $E = i$

<u>Urratsak:</u>

- While-aren erregela aplikatzerakoan honako kalkulu eta egiaztapen hauek burutu behar dira

I. $\varphi \rightarrow INB?$

II. $INB \rightarrow def(B)?$

III. $\{INB \wedge B\}$ $\quad (INB \wedge B) \rightarrow \varphi_1?$

$\{\varphi_1\}$

(EA) $i := i - 2;$

$\{INB\}$

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

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

VI. $\{INB \wedge B \wedge E = v\}$ $\quad (INB \wedge B \wedge E = v) \rightarrow \varphi_2?$

$\{\varphi_2\}$

(EA) $i := i - 2;$

$\{E < v\}$

I. $\varphi \rightarrow INB?$

$true \rightarrow (i \geq 0) \wedge bikoitia(i)?$

Inplikazio hau ez da betetzen. Adibidez i-ren balioa 3 denean inplikazioaren ezkerreko aldea betetzen da (true delako) baina eskuineko aldea ez da betetzen 3 ez delako bikoitia eta **true → false** gelditzen da eta hori false da.

Beraz **programa ez da zuzena**.

3.2.4.8. Hamazazpigarren adibidea

Honako programa hau zuzena al da?

Programa hau 16. adibideko programaren antzekoa da, baina inbariantea desberdina da.

$\{\varphi\} \equiv \{\text{true}\}$
<u>while</u> $\{INB\}$ $i \neq 0$ **<u>loop</u>**
  $i := i - 2;$
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{i = 0\}$

$\{INB\} \equiv \{\text{true}\}$

$E = i$

<u>Urratsak:</u>

- While-aren erregela aplikatzean honako kalkulu eta egiaztapen hauek egin behar dira

I. $\varphi \rightarrow INB$?
II. $INB \rightarrow def(B)$?

III. $\{INB \wedge B\}$ $\qquad$ $(INB \wedge B) \rightarrow \varphi_1$?
$\{\varphi_1\}$
$i := i - 2;$
(EA)
$\{INB\}$

IV. $(INB \wedge \neg B) \rightarrow \psi$?
V. $(INB \wedge B) \rightarrow E > 0$?
VI. $\{INB \wedge B \wedge E = v\}$ $\qquad$ $(INB \wedge B \wedge E = v) \rightarrow \varphi_2$?
$\{\varphi_2\}$
$i := i - 2;$
(EA)
$\{E < v\}$

I. $\varphi \rightarrow INB$?
true → true? Si

II. $INB \rightarrow def(B)$?
true → true? Bai, inplikazioaren bigarren zatia true delako.

III.

- $\{\varphi_1\} \equiv \{def(i - 2) \wedge (INB)_i^{i-2}\} \equiv$
  $\equiv \{true \wedge true\} \equiv$ sinplifikazioa
  $\equiv \{true\}$

- $(INB \wedge B) \rightarrow \varphi_1$?

  (true ∧ **i ≠ 0**) → true? Bai, bigarren zatia true delako.

IV. $(INB \wedge \neg B) \rightarrow \psi$?
(true ∧ ¬(**i ≠ 0**)) → i = 0? Bai.

V. $(INB \wedge B) \rightarrow E > 0$?
(true ∧ (**i ≠ 0**)) → i > 0? Ez.

Ez da beharrezkoa VI puntua begiratzea. Programa **partzialki zuzena da baina ez da guztiz zuzena**.

- **<u>Zuzentasun partzialaren froga:</u>**

I
1. $\varphi \rightarrow INB$

II
2. $INB \rightarrow def(B)$

III
3. $(INB \wedge B) \rightarrow \varphi_1$
4. $\{\varphi_1\}$ i : = i – 2; {INB} (EA)
5. {INB ∧ B}
   i : = i – 2;
   {INB} (KE 5, 6)

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

7. {φ}
   <u>while</u> {INB} i ≠ n **<u>loop</u>**
      i : = i – 2;
   <u>end</u> **<u>loop</u>**;
   {ψ} (WE 1, 2, 5, 6)

Froga honen bidez bakarrik zuzentasun partziala frogatzen da.

3.2.4.9. Hamazortzigarren adibidea

Honako programa hau zuzena al da?

Programa hau aurreko adibideetako programen antzekoa da. Desberdintasuna hasierako baldintzan eta inbariantean dago.

$\{\varphi\} \equiv \{i \geq 0 \wedge bikoitia(i)\}$ **while** {INB} $i \neq 0$ **loop** $i := i - 2;$ **end loop**; $\{\psi\} \equiv \{i = 0\}$
$\{INV\} \equiv \{i \geq 0 \wedge bikoitia(i)\}$ $E = i$

Urratsak:

- While-aren erregela aplikatzean honako kalkulu eta egiaztapen hauek egin behar dira

I.	$\varphi \rightarrow INB?$		
II.	$INB \rightarrow def(B)?$		
III.	$\{INB \wedge B\}$		$(INB \wedge B) \rightarrow \varphi_1?$
	$\{\varphi_1\}$		
	$i := i - 2;$		
(EA)	$\{INB\}$		
IV.	$(INB \wedge \neg B) \rightarrow \psi?$		
V.	$(INB \wedge B) \rightarrow E > 0?$		
VI.	$\{INB \wedge B \wedge E = v\}$		$(INB \wedge B \wedge E = v) \rightarrow \varphi_2?$
	$\{\varphi_2\}$		
	$i := i - 2;$		
(EA)	$\{E < v\}$		

I. $\varphi \rightarrow INB?$

$i \geq 0 \wedge bikoitia(i) \rightarrow i \geq 0 \wedge bikoitia(i)?$ Bai, inplikazioaren alde bietan gauza bera daukagulako.

II. $INB \rightarrow def(B)?$

$i \geq 0 \wedge bikoitia(i) \rightarrow true?$ Sbai, inplikazioaren bigarren zatian true daukagulako.

III.

- $\{\varphi_1\} \equiv \{\text{def}(i-2) \wedge (\text{INB})_i^{i-2}\} \equiv$
  $\equiv \{\text{true} \wedge i-2 \geq 0 \wedge \text{bikoitia}(i-2)\} \equiv_{\text{sinplifikazioa}}$
  $\equiv \{i-2 \geq 0 \wedge \text{bikoitia}(i-2)\}$

Edozein formula δ hartuta, *true* ∧ δ ≡ δ baliokidetasuna betetzen denez, *true* ken daiteke.

- (INB ∧ B) → φ_1?

$$\{\underbrace{i \geq 0 \wedge}_{\alpha}\underbrace{\text{bikoitia}(i)}_{\beta} \wedge \underbrace{\mathbf{i \neq 0}}_{\delta}\}$$

$$\downarrow\mathbf{?}$$

$$\underbrace{i-2 \geq 0}_{\alpha,\ \beta \text{ eta } \delta\text{-gatik}} \wedge \underbrace{\text{bikoitia}(i-2)}_{\beta\text{-gatik}}$$

IV. (INB ∧ ¬B) → ψ?

$(\, i \geq 0 \wedge \text{bikoitia}(i) \wedge \underbrace{\neg(\,\mathbf{i \neq 0})}_{\alpha}) \rightarrow i = 0$? Bai, α-gatik

V. (INB ∧ B) → E > 0?

$(\underbrace{i \geq 0}_{\alpha} \wedge \text{bikoitia}(i) \wedge \underbrace{(\mathbf{i \neq 0})}_{\beta}) \rightarrow \underbrace{i > 0}_{\alpha \text{ eta } \beta\text{-gatik.}}$? Bai

VI.

- $\{\varphi_2\} \equiv \{\text{def}(i-2) \wedge (E < v)_i^{i-2}\} \equiv$
  $\equiv \{\text{true} \wedge (i-2) < v\} \equiv_{\text{sinplifikazioa}}$
  $\equiv \{(i-2) < v\}$

Edozein formula δ hartuta, *true* ∧ δ ≡ δ baliokidetasuna betetzen denez, *true* ken daiteke.

- (INB ∧ B ∧ E = v) → φ_2?

$(i \geq 0 \wedge \text{bikoitia}(i) \wedge \mathbf{(i \neq 0)} \wedge \underbrace{(\mathbf{i = v})}_{\alpha}) \rightarrow \underbrace{(i-2) < v}_{\alpha\text{-gatik}}$? Bai.

- **Zuzentasun osoaren froga:**

I

1. $\varphi \rightarrow INB$

II

2. $INB \rightarrow def(B)$

III

3. $(INB \wedge B) \rightarrow \varphi_1$
4. $\{\varphi_1\}$ i : = i – 2; {INB} (EA)
5. $\{INB \wedge B\}$
   i : = i – 2;
   {INB} (KE 3, 4)

IV

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

V

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

VI

8. $(INB \wedge B \wedge E = v) \rightarrow \varphi_2$
9. $\{\varphi_2\}$ i : = i – 2; $\{E < v\}$ (EA)
10. $\{INB \wedge B \wedge E = v\}$
    i : = i – 2;
    $\{E < v\}$ (KE 8, 9)

11. $\{\varphi\}$

while {INB} $i \neq n$ **loop**
    i : = i – 2;
end **loop**;
$\{\psi\}$ (WE 1, 2, 5, 6, 7, 10)

3.2.4.10. Hemeretzigarren adibidea

Honako programa hau zuzena al da?

Programa hau aurreko adibideetako programen antzekoa da. Desberdintasuna hasierako baldintzan eta inbariantean dago.

$\{\varphi\} \equiv \{\ i \geq 0 \wedge \text{bikoitia}(i)\}$ **while** {INB} $i \neq 0$ **loop** $i := i - 2;$ **end loop**; $\{\psi\} \equiv \{i = 0\}$
$\{INB\} \equiv \{true\}$ $E = i$

Urratsak:

- While-aren erregela aplikatzean honako kalkulu eta egiaztapen hauek egin behar dira

	I.	$\varphi \rightarrow INB$?	
	II.	$INB \rightarrow def(B)$?	
	III.	$\{INB \wedge B\}$	$(INB \wedge B) \rightarrow \varphi_1$?
		$\{\varphi_1\}$	
		$i := i - 2;$	
(EA)			
		$\{INB\}$	
	IV.	$(INB \wedge \neg B) \rightarrow \psi$?	
	V.	$(INB \wedge B) \rightarrow E > 0$?	
	VI.	$\{INB \wedge B \wedge E = v\}$	$(INB \wedge B \wedge E = v) \rightarrow \varphi_2$?
		$\{\varphi_2\}$	
		$i := i - 2;$	
(EA)			
		$\{E < v\}$	

I. $\varphi \rightarrow INB$?

$i \geq 0 \wedge \text{bikoitia}(i) \rightarrow true$? Bai, inplikazioaren bigarren zatian true daukagulako.

II. $INB \rightarrow def(B)$?

$true \rightarrow true$? Bai, inplikazioaren bigarren zatian true daukagulako.

III.

- $\{\varphi_1\} \equiv \{def(i-2) \wedge (INB)_i^{i-2}\} \equiv$
  $\equiv \{true \wedge true\} \equiv^{\text{sinplifikazioa}}$
  $\equiv \{true\}$

Edozein formula δ hartuta, *true* $\wedge\ \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.

- $(INB \wedge B) \rightarrow \varphi_1$?

(true $\wedge$ **i ≠ 0**) → true? Bai, inplikazioaren bigarren zatia true delako.

IV. $(INB \wedge \neg B) \rightarrow \psi$?
$(true \wedge \neg\underbrace{(\mathbf{i \neq 0})}_{\alpha}) \rightarrow \underbrace{i = 0}_{\alpha\text{-gatik}}$? Bai.

V. $(INB \wedge B) \rightarrow E > 0$?
(true $\wedge$ (**i ≠ 0**)) → i > 0? Ez. Izan ere, i aldagaiaren balioa 0ren desberdina izateak ez du esan nahi i aldagaiaren balioa 0 baino handiagoa izango denik. Zero baino txikiagoa ere izan baidaiteke.

VI puntua begiratu beharrik ez dago. Programa hau ez da guztiz zuzena, bakarrik partzialki zuzena da.

- **Zuzentasun partzialaren froga:**

I	1.	$\varphi \rightarrow INB$
II	2.	$INB \rightarrow def(B)$
III	3.	$(INB \wedge B) \rightarrow \varphi_1$
	4.	$\{\varphi_1\}$ i : = i – 2; {INB} (EA)
	5.	$\{INB \wedge B\}$ i : = i – 2; {INB} (OE 3, 4)
IV	6.	$(INB \wedge \neg B) \rightarrow \psi$

Froga honen bidez zuzentasun partziala frogatzen da.

3.2.4.11. Hogeigarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{x = a \wedge y = b\}$
<u>while</u> {INB} $y \neq 0$ **<u>loop</u>**
    x : = x + 1;
    y : = y – 1;
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{x = a + b\}$

$\{INB\} \equiv \{x + y = a + b\}$ E = y

<u>Urratsak:</u>

- While-aren erregela aplikatzean honako kalkulu eta egiaztapen hauek egin behar dira

I. $\varphi \rightarrow INB$?

II. $INB \rightarrow def(B)$?

III. $\{INB \wedge B\}$ $\qquad$ $(INB \wedge B) \rightarrow \varphi_2$?

$\{\varphi_2\}$

x : = x + 1;

(EA)

$\{\varphi_1\}$

y : = y – 1;

(EA)

{INB}

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

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

VI. $\{INB \wedge B \wedge E = v\}$ $\qquad$ $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$?

$\{\varphi_4\}$

x : = x + 1;

(EA)

$\{\varphi_3\}$

y : = y – 1;

(EA)

$\{E < v\}$

I. $\varphi \rightarrow INB$?

$$\{\underbrace{x = a}_{\alpha} \wedge \underbrace{y = b}_{\beta}\}$$

$$\downarrow ?$$

$$\{\underbrace{x + y = a + b}_{\alpha \text{ eta } \beta\text{-gatik}}\}$$

x = *a* ∧ y = b betetzen dela suposatuz, x + y = *a* + b ere betetzen al den aztertu behar da.

II. INB → def(B)?
INB → true? Bai, inplikazioaren bigarren zatia true delako.

III.

- $\{\varphi_1\} \equiv \{def(y - 1) \wedge (INB)_y^{y-1}\} \equiv$
  $\equiv \{true \wedge x + y - 1 = a + b\} \equiv$ sinplifikazioa
  $\equiv \{x + y - 1 = a + b\}$

  Edozein formula δ hartuta, *true* ∧ δ ≡ δ baliokidetasuna betetzen denez, *true* ken daiteke.

.

- $\{\varphi_2\} \equiv \{def(x + 1) \wedge (\varphi_1)_x^{x+1}\} \equiv$
  $\equiv \{true \wedge x + 1 + y - 1 = a + b\} \equiv$ sinplifikazioa
  $\equiv \{x + y = a + b\}$

  Edozein formula δ hartuta, *true* ∧ δ ≡ δ baliokidetasuna betetzen denez, *true* ken daiteke.

  x + 1 + y – 1 = *a* + b formularen kasuan, ezkerreko aldean 1 batu eta 1 kentzen denez, batuketa hori eta kenketa hori ken daitezke eta x + y = *a* + b geldituko zaigu.

- $(INB \wedge B) \rightarrow \varphi_2$?

$$\underbrace{x + y = a + b}_{\alpha} \wedge \underbrace{y \neq 0}_{\beta}$$

$$\downarrow ?$$

$$\underbrace{x + y = a + b}_{\alpha \text{ eta } \beta\text{-gatik}}$$

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

$$\underbrace{x + y = a + b}_{\alpha} \wedge \underbrace{y = 0}_{\beta}$$
$$\downarrow \mathbf{?}$$
$$\underbrace{\{x = a + b\}}_{\alpha \text{ eta } \beta\text{-gatik}}$$

Goiko formulak y = 0 dela dioenez, β formulan y-ren ordez 0 ipiniz gelditzen dena ψ formularen berdina da eta beraz inplikazioa bete egiten da. Gogoratu ψ formula $x = a + b$ dela.

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

$$\underbrace{x + y = a + b}_{\alpha} \wedge \underbrace{y \neq 0}_{\beta}$$
$$\downarrow \mathbf{?}$$
$$y > 0$$

Inplikazio hau ez da betetzen eta ondorioz zuzentasun osoa ezin da frogatu, bakarrik zuzentasun partziala froga daiteke.

VI. puntua aztertu beharrik ez dago, dagoeneko bai baitakigu ez delas guztiz zuzena izango V puntua ez delako bete.

- **Zuzentasun partzialaren froga:**

I

1. $\varphi \rightarrow INB$

II

2. $INB \rightarrow def(B)$

III

3. $(INB \wedge B) \rightarrow \varphi_2$
4. $\{\varphi_2\}$ x : = x + 1; $\{\varphi_1\}$ (EA)
5. $\{INB \wedge B\}$ x : = x + 1; $\{\varphi_1\}$ (KE 3, 4)
6. $\{\varphi_1\}$ y : = y – 1; {INB} (EA)
7. $\{INB \wedge B\}$
   x : = x + 1;
   y : = y – 1;
   {INB} (KE 5, 6)

IV

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

9. $\{\varphi\}$
   while {INB} $i \neq n$ **loop**
      x : = x + 1;
      y : = y – 1;
   end loop;
   $\{\psi\}$ (RWE 1, 2, 7, 8)

3.2.4.12. Hogeita batgarren adibidea

Honako programa hau zuzena al da?

$\{\varphi\} \equiv \{x = a \wedge y = b \geq 0\}$
<u>while</u> $\{INB\}$ $y \neq 0$ **<u>loop</u>**
    $x := x + 1;$
    $y := y - 1;$
<u>end</u> **<u>loop</u>**;
$\{\psi\} \equiv \{x = a + b\}$

$\{INB\} \equiv \{(0 \leq y \leq b) \wedge x + y = a + b\}$ $E = y$

<u>Urratsak:</u>

- While-aren erregela aplikatzean honako kalkulu eta egiaztapen hauek egin behar dira

I. $\varphi \rightarrow INB$?

II. $INB \rightarrow def(B)$?

III. $\{INB \wedge B\}$ $\qquad$ $(INB \wedge B) \rightarrow \varphi_2$?

$\{\varphi_2\}$

$x := x + 1;$

(EA)

$\{\varphi_1\}$

$y := y - 1;$

(EA)

$\{INB\}$

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

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

VI. $\{INB \wedge B \wedge E = v\}$ $\qquad$ $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$?

$\{\varphi_4\}$

$x := x + 1;$

(EA)

$\{\varphi_3\}$

$y := y - 1;$

(EA)

$\{E < v\}$

I. $\varphi \rightarrow$ INB?

$$\{\underbrace{x = a}_{\alpha} \wedge \underbrace{y = b \geq 0}_{\beta}\}$$

$$\downarrow ?$$

$$\{\underbrace{(0 \leq y \leq b)}_{\beta\text{-gatik}} \wedge \underbrace{x + y = a + b}_{\alpha \text{ eta } \beta\text{-gatik}}\}$$

$x = a \wedge y = b \geq 0$ betetzen dela suposatuz, $(0 \leq y \leq b) \wedge x + y = a + b$ betetzen al den begiratu behar da.

Inplikazio hori bete egiten da. α eta β letra grekoen bidez beheko zatiko atal bakoitza bete egiten dela esateko goiko zatiko zein ataletan oinarritzen garen erakusten da.

II. INB → def(B)?
INB → true? Bai, inplikazioaren bigarren zatia true delako.

III.

- $\{\varphi_1\} \equiv \{\text{def}(y - 1) \wedge (\text{INB})_y^{y-1}\} \equiv$
  $\equiv \{\text{true} \wedge (0 \leq y - 1 \leq b) \wedge x + y - 1 = a + b\} \equiv$ sinplifikazioa
  $\equiv \{(1 \leq y \leq b + 1) \wedge x + y - 1 = a + b\}$

Edozein formula δ hartuta *true* $\wedge\ \delta \equiv \delta$ baliokidetasuna betetzen denez, true ken daiteke.

Bestalde, $(0 \leq y - 1 \leq b)$ formula $y - 1$ erabiliz ediku beharrean hobe da y erabiliz ipintzea. Horretarako 1 balioa batu behar zaie hiru osagaiei: $(0 + 1 \leq y - 1 + 1 \leq b + 1)$. Eragiketak burutu ondoren honako hau geldituko da $(1 \leq y \leq b + 1)$.

- $\{\varphi_2\} \equiv \{\text{def}(x + 1) \wedge (\varphi_1)_x^{x+1}\} \equiv$
  $\equiv \{\text{true} \wedge (1 \leq y \leq b + 1) \wedge x + 1 + y - 1 = a + b\} \equiv$ sinplifikazioa
  $\equiv \{(1 \leq y \leq b + 1) \wedge x + y = a + b\}$

Edozein formula δ hartuta, *true* $\wedge\ \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.

$x + 1 + y - 1 = a + b$ formularen kasuan, ezkerreko aldean 1 batu eta 1 kentzen denez, batuketa hori eta kenketa hori ken daitezke eta $x + y = a + b$ geldituko zaigu.

- $(INB \wedge B) \rightarrow \varphi_2$?

$$\underbrace{(0 \le y \le b)}_{\alpha} \wedge \underbrace{x + y = a + b}_{\beta} \wedge \underbrace{y \ne 0}_{\delta}$$
$$\Big\downarrow ?$$
$$\underbrace{(1 \le y \le b + 1)}_{\alpha \text{ eta } \delta\text{-gatik}} \wedge \underbrace{x + y = a + b}_{\beta\text{-gatik}}$$

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

$$\underbrace{(0 \le y \le b)}_{\alpha} \wedge \underbrace{x + y = a + b}_{\beta} \wedge \underbrace{y = 0}_{\delta}$$
$$\Big\downarrow ?$$
$$\underbrace{\{x = a + b\}}_{\beta \text{ eta } \delta\text{-gatik}}$$

y = 0 denez, β formulan y-ren ordez 0 ipiniz ψ formula lortzen da.

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

$$\underbrace{(0 \le y \le b)}_{\alpha} \wedge x + y = a + b \wedge \underbrace{y \ne 0}_{\beta}$$
$$\Big\downarrow ?$$
$$\underbrace{y > 0}_{\alpha \text{ eta } \beta\text{-gatik}}$$

α-gatik badakigu $y \ge 0$ betetzen dela eta β-gatik badakigu $\mathbf{y \ne 0}$ betetzen dela eta ondorioz $y > 0$ da.

VI.

- $\{\varphi_3\} \equiv \{def(y - 1) \wedge (E < v)_y^{y-1}\} \equiv$
  $\equiv \{true \wedge y - 1 < v\} \equiv$ sinplifikazioa
  $\equiv \{y - 1 < v\}$

- $\{\varphi_4\} \equiv \{def(x + 1) \wedge (\varphi_3)_x^{x+1}\} \equiv$
  $\equiv \{true \wedge y - 1 < v\} \equiv$ sinplifikazioa
  $\equiv \{y - 1 < v\}$

Edozein formula δ hartuta, $true \wedge \delta \equiv \delta$ baliokidetasuna betetzen denez, *true* ken daiteke.

- $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$?

$$(0 \le y \le b) \wedge x + y = a + b \wedge y \ne 0 \wedge \underbrace{y = v}_{\alpha}$$

$$\downarrow ?$$

$$\underbrace{y - 1 < v}_{\alpha\text{-gatik}}$$

y aldagaiaren balioa v aldagaiaren balioaren berdina bada, y – 1 balioa v – 1 balioaren berdina izango da eta ondorioz v baino txikiagoa izango da.

- **Zuzentasun osoaren froga:**

I

1. $\varphi \rightarrow INB$

II

2. $INB \rightarrow def(B)$

III

3. $(INB \wedge B) \rightarrow \varphi_2$
4. $\{\varphi_2\}$ x : = x + 1; $\{\varphi_1\}$ (EA)
5. $\{INB \wedge B\}$ x : = x + 1; $\{\varphi_1\}$ (OE 3, 4)
6. $\{\varphi_1\}$ y : = y – 1; {INB} (EA)
7. $\{INB \wedge B\}$
   x : = x + 1;
   y : = y – 1;
   {INB} (KE 5, 6)

IV

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

V

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

VI

10. $(INB \wedge B \wedge E = v) \rightarrow \varphi_4$
11. $\{\varphi_4\}$ x : = x + 1; $\{\varphi_3\}$ (EA)
12. $\{INB \wedge B \wedge E = v\}$ x : = x + 1; $\{\varphi_3\}$ (OE 10, 11)
13. $\{\varphi_3\}$ y : = y – 1; $\{E < v\}$ (EA)
14. $\{INB \wedge B \wedge E = v\}$
    x : = x + 1;
    y : = y – 1;
    $\{E < v\}$ (KE 12, 13)

15. $\{\varphi\}$
    while {INB} $i \ne n$ **loop**
      x : = x + 1;
      y : = y – 1;
    end loop;
    $\{\psi\}$ (WE 1, 2, 7, 8, 9, 14)