DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1

Especificación y WP

18 de septiembre de 2023 — Algoritmos y Estructuras de Datos

Grupo losJavalíes

Integrante	LU	Correo electrónico
Cuneo, Lautaro	1581/21	`lautarocuneo@gmail.com`
Otazua Arce, Mateo	88/23	`tazuarce@gmail.com`
Pego, Micaela Giselle	380/22	`micaelapego@gmail.com`
Bonadykov, Felipe Igor	1942/21	`felipe.bonadykov@gmail.com`

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

1. Funciones útiles

`pred` `esMatriz` (m: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})(0 \leq i < |m| \longrightarrow_L |m[0]| = |m[i]|)$
}
`aux` `totalEscrutinio` (escrutinio: $seq\langle \mathbb{Z}\rangle$) : $\mathbb{Z}$ =
$\sum\limits_{i=0}^{|escrutinio|-1} escrutinio[i]$;
`pred` `sinRepetidosExcepto0` (matriz: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})$ $((\forall j : \mathbb{Z})$ $(0 \leq i < |matriz| \wedge 0 \leq j < |matriz[0]| \wedge matriz[i][j] \neq 0 \longrightarrow_L$
$(\forall k : \mathbb{Z})$ $((\forall l : \mathbb{Z})$ $(0 \leq k < |matriz| \wedge 0 \leq l < |matriz[0]| \wedge matriz[k][l] \neq 0 \wedge (k \neq i \vee l \neq j) \longrightarrow_L$
$matriz[i][j] \neq matriz[k][l])))))$
}
`pred` `todosDistintosLista` (lista: $seq\langle \mathbb{Z}\rangle$) {
$(\forall i : \mathbb{Z})$ $((\forall j : \mathbb{Z})$ $(0 \leq i < |lista| \wedge 0 \leq j < |lista| \wedge (i \neq j) \longrightarrow_L lista[i] \neq lista[j]))$
}
`aux` `porcentajeDeVotos` (partido : $\mathbb{Z}$, escrutinio: $seq\langle \mathbb{Z}\rangle$) : $\mathbb{Z}$ =
$(escrutinio[partido]/totalEscrutinio(escrutinio)) * 100$;
`pred` `escrutinioValido` (escrutinio : $seq\langle \mathbb{Z}\rangle$) {
$|escrutinio| \geq 2 \wedge todosPositivoso0(escrutinio) \wedge todosDistintosLista(subseq(escrutinio, 0, |escrutinio| - 1))$
}
`pred` `todosPositivoso0` (lista : $seq\langle \mathbb{Z}\rangle$) {
$(\forall elem : \mathbb{Z})$ $(0 \leq elem < |lista| \longrightarrow_L lista[elem] \geq 0)$
}
`pred` `esMatrizdHondt` (dHondt: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, cant_bancas : $\mathbb{Z}$) {
$esMatriz(dHondt) \wedge |dHondt| > 0 \wedge |dHondt[0]| = cant\_bancas \wedge$
$(\forall i, j : \mathbb{Z})$ $(0 \leq j < |dHondt| \wedge 0 \leq i < cant\_bancas \longrightarrow_L res[j][i] = divEntera(res[j][0], i + 1)])$
}
`pred` `esMatrizdHondtDe` (dHondt : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, escrutinio : $seq\langle \mathbb{Z}\rangle$, cant_bancas : $\mathbb{Z}$) {
$esMatrizdHondt(dHondt, cant\_bancas) \wedge |dHondt| = |escrutinio| - 1 \wedge_L$
$(\forall partido : \mathbb{Z})$ $(0 \leq partido < |escrutinio| - 1 \longrightarrow_L$
if $porcentajeMayorA3(partido, escrutinio)$ then
$dHondt[partido][0] = escrutinio[partido]$ else
$dHondt[partido][0] = 0$ fi)
}
`pred` `porcentajeMayorA3` (partido: $\mathbb{Z}$, escrutinio : $seq\langle \mathbb{Z}\rangle$) {
$3 \leq porcentajeDeVotos(partido, escrutinio)$
}

2. Especificación

2.1. hayBallotage

`proc` `hayBallotage` (in escrutinio : $seq\langle \mathbb{Z}\rangle$) : Bool
`requiere` $\{escrutinioValido(escrutinio)\}$
`asegura` $\{res = true \iff (\forall partido : \mathbb{Z})$ $(0 \leq partido < |escrutinio| - 1 \longrightarrow_L$
$\neg ganaEnPrimeraVuelta(partido, escrutinio))\}$

`pred` `ganaEnPrimeraVuelta` (partido: $\mathbb{Z}$, escrutinio : $seq\langle \mathbb{Z}\rangle$) {
$(porcentajeDeVotos(partido, escrutinio) > 45 \vee$
$((porcentajeDeVotos(partido, escrutinio) > 40) \wedge$
$(\forall otro\_partido : \mathbb{Z})$ $(0 \leq otro\_partido < |escrutinio| - 1 \wedge partido \neq otro\_partido \longrightarrow_L$
$porcentajeDeVotos(partido, escrutinio) > porcentajeDeVotos(otro\_partido, escrutinio) + 10)))$
}

2.2. hayFraude

`proc` `hayFraude` (in escrutinio_presidencial : $seq\langle \mathbb{Z}\rangle$, in escrutinio_senadores : $seq\langle \mathbb{Z}\rangle$, in escrutinio_diputados : $seq\langle \mathbb{Z}\rangle$) : Bool
`requiere` $\{|escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados|\}$
`requiere` $\{|escrutinio\_senadores| > 2\}$
`requiere` $\{escrutinioValido(escrutinio\_presidencial)\}$
`requiere` $\{escrutinioValido(escrutinio\_senadores)\}$
`requiere` $\{escrutinioValido(escrutinio\_diputados)\}$

asegura $\{res = true \iff \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores)) \wedge$
$(totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados)))\}$

2.3. obtenerSenadoresEnProvincia

proc obtenerSenadoresEnProvincia (in escrutinio : $seq\langle \mathbb{Z} \rangle$) : $\mathbb{Z}$x$\mathbb{Z}$
requiere $\{|escrutinio| > 2\}$
requiere $\{escrutinioValido(escrutinio)\}$
asegura $\{0 \leq\ res_0,\ res_1 < |escrutinio| - 1\}$
asegura $\{escrutinio[res_0] > escrutinio[res_1]\}$
asegura $\{(\forall i : \mathbb{Z})\ (0 \leq i < |escrutinio| - 1\ \wedge\ \ i \neq res_0\ \wedge\ i \neq res_1 \longrightarrow_L escrutinio[i] < escrutinio[res_1])\}$

2.4. calcularDHondtEnProvincia

proc calcularDHondtEnProvincia (in cant_bancas : $\mathbb{Z}$, in escrutinio : $seq\langle \mathbb{Z} \rangle$) : $seq\langle seq\langle \mathbb{Z} \rangle \rangle$
requiere $\{0 < cant\_bancas\}$
requiere $\{escrutinioValido(escrutinio)\}$
requiere $\{noGeneraRepetidos1(escrutinio, cant\_bancas)\}$
requiere $\{noGeneraRepetidos2(escrutinio, cant\_bancas)\}$
asegura $\{esMatrizdHondtDe(res, escrutinio, cant\_bancas)\}$
asegura $\{sinRepetidosExcepto0(res)\}$

pred noGeneraRepetidos1 (escrutinio : $seq\langle \mathbb{Z} \rangle$, cant_bancas : $\mathbb{Z}$) {
$(\forall partido, otro\_partido : \mathbb{Z})\ (0 \leq partido, otro\_partido < |escrutinio| - 1 \wedge partido \neq otro\_partido \longrightarrow_L$
$(\forall i, j : \mathbb{Z})\ (1 \leq i, j \leq cant\_bancas \longrightarrow escrutinio[partido]\ div\ i\ \neq\ escrutinio[otro\_partido]\ div\ j)))$
}
pred noGeneraRepetidos2 (escrutinio : $seq\langle \mathbb{Z} \rangle$, cant_bancas : $\mathbb{Z}$) {
$(\forall partido : \mathbb{Z})\ (0 \leq partido < |escrutinio| - 1 \longrightarrow_L (\forall i, j : \mathbb{Z})\ (1 \leq i, j \leq cant\_bancas \wedge i \neq j \longrightarrow_L$
$divEntera(escrutinio[partido], i) \neq divEntera(escrutinio[partido], j)))$
}

2.5. obtenerDiputadosEnProvincia

proc obtenerDiputadosEnProvincia (in cant_bancas : $\mathbb{Z}$,in escrutinio : $seq\langle \mathbb{Z} \rangle$, in dHondt : $seq\langle seq\langle \mathbb{Z} \rangle\rangle$) : $seq\langle \mathbb{Z} \rangle$
requiere $\{cant\_bancas > 0\}$
requiere $\{escrutinioValido(escrutinio)\}$
requiere $\{esMatrizdHondtDe(dHondt, escrutinio, cant\_bancas)\}$
requiere $\{sinRepetidosExcepto0(dHondt)\}$
asegura $\{(\forall partido : \mathbb{Z})\ (0 \leq partido < |res| \longrightarrow_L res[partido] = bancasParaPartido(partido, dHondt, cant\_bancas))\}$
asegura $\{|res| = |dHondt|\}$

aux bancasParaPartido (partido : $\mathbb{Z}$, dHondt : $seq\langle seq\langle \mathbb{Z} \rangle\rangle$, cant_bancas : $\mathbb{Z}$) : $\mathbb{Z}$ =
$\sum_{columna=0}^{|dHondt[0]|-1}$ if $ganaBanca(dHondt, dHondt[partido][columna], cant\_bancas)$ then 1 else 0 fi ;
pred ganaBanca (dHondt : $seq\langle seq\langle \mathbb{Z} \rangle\rangle$, cociente : $\mathbb{Z}$, cant_bancas : $\mathbb{Z}$) {
$mayoresACociente(cociente, matriz) < cant\_bancas$
}
aux mayoresACociente (cociente: $\mathbb{Z}$, matriz: $seq\langle seq\langle \mathbb{Z} \rangle\rangle$) : $\mathbb{Z}$ =
$\sum_{i=0}^{|dHondt|-1} (\sum_{j=0}^{|dHondt[0]|-1}$ if $dHondt[i][j] > cociente$ then 1 else 0 fi)) ;

2.6. validarListasDiputadosEnProvincia

proc validarListasDiputadosEnProvincia (in cant_bancas : $\mathbb{Z}$, in listas : $seq\langle seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z} \rangle\rangle$) : Bool
requiere $\{0 < cant\_bancas\}$
requiere $\{0 < |listas|\}$
requiere $\{(\forall lista : \mathbb{Z})\ (0 \leq lista < |listas| \longrightarrow_L (\forall candidato : \mathbb{Z})\ (0 \leq candidato < |listas[lista]| \longrightarrow_L$
$listas[lista][candidato]_1 = 0 \vee listas[lista][candidato]_1 = 1))\}$
asegura $\{res = true \iff seRespetaParidadDeGenero(listas) \wedge candidatosJustos(listas, cant\_bancas)\}$

pred seRespetaParidadDeGenero (listas : $seq\langle seq\langle (\mathbb{Z} \times \mathbb{Z})\rangle\rangle$) {
$(\forall i : \mathbb{Z})\ (0 \leq i < |listas| \longrightarrow_L (|listas[i]| = 1 \vee_L generosIntercalados(listas[i])))$

}
`pred generosIntercalados` (lista : $seq\langle(\mathbb{Z} \times \mathbb{Z})\rangle$) {

$lista[0]_1 \neq lista[1]_1 \wedge (\forall i : \mathbb{Z})\ (0 \leq i < |lista| \longrightarrow_L lista[i]_1 = lista[imod2)]_1)$

}
`pred candidatosJustos` (listas : $seq\langle seq\langle(\mathbb{Z} \times \mathbb{Z})\rangle\rangle$, cant_bancas : $\mathbb{Z}$) {

$(\forall i : \mathbb{Z})\ (0 \leq i < |listas| \longrightarrow_L |listas[i]| = cant\_bancas)$

}

3. Implementaciones

3.1. hayBallotage

```
res := true

totalVotos := 0
i := 0
while (i < escrutinio.size()) do
    totalVotos := totalVotos + escrutinio[i]
    i := i + 1
endwhile

i := 0
while (i < escrutinio.size()) do
    escrutinio[i] := (escrutinio[i] / totalVotos) ∗ 100
    i := i + 1
endwhile

i := 0
mayorP := 0
while (i < escrutinio.size()−1) do
    if (escrutinio[i] > mayorP) then
        mayorP := escrutinio[i]
    else
        skip
    endif
    i := i + 1
endwhile

if (mayorP > 45) then
    res := false
else
    skip
endif

i := 0
superaPor10 := true
while (i < escrutinio.size()−1) do
    if (mayorP < escrutinio[i] + 10 ∧ escrutinio[i]≠ mayorP) then
        superaPor10 := false
    else
        skip
    endif
    i := i + 1
endwhile


if (superaPor10 ∧ mayorP > 40) then
    res := false
else
    skip
endif

return res
```

3.2. hayFraude

```
totalEscP := 0
totalEscS := 0
totalEscD := 0
i := 0

while (i < escrutinioPresidencial.size()) do
    totalEscP := totalEscP + escrutinioPresidencial[i]
    totalEscS := totalEscS + escrutinioSenadores[i]
    totalEscD := totalEscD + escrutinioDiputados[i]
    i := i + 1
endwhile

res := false
if (totalEscP ≠ totalEscS ∨ totalEscP ≠ totalEscD) then
    res := true
else
    skip
endif
return res
```

3.3. obtenerSenadoresEnProvincia

```
i := 1
primero := 0
segundo := 1

while (i < escrutinio.size()−1) do
    if (escrutinio[i] > escrutinio[primero]) then
        segundo := primero
        primero := i
    else
        if (escrutinio[i] > escrutinio[segundo]) then
            segundo := i
        else
            skip
        endif
    endif
    i := i + 1
endwhile

res := (primero, segundo)
return res
```

3.4. validarListasDiputadosEnProvincia

```
candidatosJustos := true
i := 0
while (i < listas.size()) do
    if (listas[i].size() ≠ cant_bancas) then
        candidatosJustos := false
    else
        skip
    endif
i := i + 1
endwhile

intercaladoHMH := true
intercaladoMHM := true
i := 0
while (i < listas.size()) do
    j := 0
    while (j < listas[i].size()) do
        if (((j mod 2 = 0) ∧ (listas[i][j][1] ≠ 0)) ∨ ((j mod 2 = 1) ∧ (listas[i][j][1] ≠ 1))) then
            intercaladoHMH := false
        else
            skip
        endif
        if (((j mod 2 = 0) ∧ (listas[i][j][1] ≠ 1)) ∨ ((j mod 2 = 1) ∧ (listas[i][j][1] ≠ 0))) then
            intercaladoMHM := false
        else
            skip
        endif
        j : j + 1
    endwhile
    i := i + 1
endwhile

seRespetaParidadDeGenero := false
if (intercaladoHMH ∨ intercaladoMHM) then
    seRespetaParidadDeGenero := true
else
    skip
endif

res := false
if (candidatosJustos ∧ seRespetaParidadDeGenero) then
    res := true
else
    skip
endif
return res
```

4. Demostraciones de correctitud

4.1. obtenerSenadoresEnProvincia

Recordemos la especificación:

`proc obtenerSenadoresEnProvincia` (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$x$\mathbb{Z}$
 `requiere` $\{|escrutinio| > 2\}$
 `requiere` $\{escrutinioValido(escrutinio)\}$
 `asegura` $\{0 \leq\ res_0,\ res_1 < |escrutinio| - 1\}$
 `asegura` $\{escrutinio[res_0] > escrutinio[res_1]\}$
 `asegura` $\{(\forall i : \mathbb{Z})\ (0 \leq i < |escrutinio| - 1\ \wedge\ \ i \neq res_0\ \wedge\ i \neq res_1 \longrightarrow_L escrutinio[i] < escrutinio[res_1])\}$

El programa tiene tres partes:

S_1

```
i := 0
primero := 0
segundo := 1
```

C

```
while(i < escrutinio.size()-1) do
    if (escrutinio[i] > escrutinio[primero]) then
        segundo := primero
        primero := i
    else
        if (escrutinio[i]> escrutinio[segundo]) then
            segundo := i
        else
            skip
        endif
    endif
    i:= i + 1
endwhile
```

S_2

```
res := (primero, segundo)
return res
```

Para probar la correctitud del programa, debemos probar la siguiente tripla de Hoare:

$$\{P\}\ S_1; C; S_2\ \{Q\}$$

Lo que es equivalente a probar las siguientes tres cosas:

$$\{P\}\ S_1\ \{P_C\}$$
$$\{P_C\}\ C\ \{Q_C\}$$
$$\{Q_C\}\ S_2\ \{Q\}$$

A lo largo de esta demostración vamos a usar el siguiente predicado, especificado en la primera parte de este informe:

`pred escrutinioValido` (escrutinio : $seq\langle\mathbb{Z}\rangle$) {
 $|escrutinio| \geq 2 \wedge todosPositivoso0(escrutinio) \wedge todosDistintosLista(subseq(escrutinio, 0, |escrutinio| - 1))$
}

4.1.1. Principio del programa

Primero probamos la primera parte del programa:

$$\{P\}\ S_1\ \{P_C\}$$

Esto es equivalente a probar:

$$P \longrightarrow wp(S_1, P_C)$$

Calculamos $wp(S_1, P_C)$

$$P_C \equiv i = 1 \wedge primero = 0 \wedge segundo = 1 \wedge |escrutinio| > 2 \wedge escrutinioValido(escrutinio)$$

S_1

```
i := 1
primero := 0
segundo := 1
```

Aplicando el axioma 2, de asignación, sabemos que $wp(S_1, P_C) \equiv (((P_C)^{segundo}_1)^{primero}_0)^i_1$. Reemplazamos correspondientemente:

$$wp(S_1, P_C) \ \equiv \ 1 = 1 \wedge 0 = 0 \wedge 1 = 1 \wedge |escrutinio| > 2$$

$$wp(S_1, P_C) \ \equiv \ |escrutinio| > 2$$

Se ve a simple vista que $P \longrightarrow wp(S_1, P_C)$, ya que $|escrutinio| > 2$ está presente en el requiere del programa.

Queda probada la correctitud de la primera parte de nuestro programa.

$$\{P\}\ S_1\ \{P_C\}$$

4.1.2. Correctitud del ciclo:

Ahora continuamos probando la correctitud de la segunda parte de nuestro programa.

$$\{P_C\}\ C\ \{Q_C\}$$

Para probar la correctitud del ciclo, hacemos uso del **teorema del invariante** y del **teorema de terminación**. Primero vamos con el invariante. Para hallar un posible candidato, vemos P_C, B_C y Q_C.

$$P_C \ \equiv \ i = 1 \wedge primero = 0 \wedge segundo = 1 \wedge |escrutinio| > 2 \wedge escrutinioValido(escrutinio)$$

$$B_C \ \equiv \ i < |escrutinio| - 1$$

$$\begin{aligned} Q_C \ \equiv \ & i = |escrutinio| - 1 \wedge \\ & 0 \ \leq \ primero, segundo < |escrutinio| - 1 \wedge \\ & escrutinio[primero] > escrutinio[segundo] \wedge \\ & (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio| - 1 \wedge k \neq primero \wedge k \neq segundo \longrightarrow escrutinio[k] < escrutinio[segundo])) \end{aligned}$$

Nuestro posible invariante es el siguiente:

$$\begin{aligned} I \ \equiv \ & 0 \ \leq \ primero,\ segundo < |escrutinio| - 1 \wedge \\ & 1 \ \leq \ i \ \leq \ |escrutinio| - 1 \wedge \\ & (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero])) \wedge \\ & (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

Debe cumplir con las siguientes tres condiciones:

$$P_C \longrightarrow I$$
$$\{I \ \wedge \ B_C\} \ C \ \{I\}$$
$$I \ \wedge \ \neg B_C \ \longrightarrow \ Q_C$$

Procedemos a demostrar que las cumple.

$\underline{P_C \longrightarrow I}$

Queremos ver que la precondición del ciclo implica el invariante.

$$\begin{aligned}
P_C \ &\equiv \ i = 1 \wedge primero = 0 \wedge segundo = 1 \wedge |escrutinio| > 2 \wedge escrutinioValido(escrutinio) \\
&\Longrightarrow \\
I \ &\equiv \ 0 \ \leq \ primero, \ segundo < |escrutinio| - 1 \wedge \\
&\quad 1 \ \leq \ i \ \leq \ |escrutinio| - 1 \wedge \\
&\quad (\forall k : \mathbb{Z}) \ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero])) \wedge \\
&\quad (\forall j : \mathbb{Z}) \ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
&\quad escrutinioValido(escrutinio)
\end{aligned}$$

Podemos ver la implicación por partes:

- $$i = 1 \longrightarrow 1 \leq i \leq |escrutinio| - 1 \checkmark$$

- $$primero = 0 \wedge segundo = 1 \wedge |escrutinio| > 2 \longrightarrow 0 \leq primero, \ segundo < |escrutinio| - 1 \checkmark$$

- $$escrutinioValido(escrutinio) \longrightarrow escrutinioValido(escrutinio) \checkmark$$

- $$\begin{aligned}
&i = 1 \wedge segundo = 1 \wedge primero = 0 \longrightarrow \\
&(\forall k : \mathbb{Z}) \ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
&(\forall j : \mathbb{Z}) \ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo])
\end{aligned}$$

  Reemplazando correspondientemente nos queda:

  $$\begin{aligned}
  &(\forall k : \mathbb{Z}) \ (0 \leq k < 1 \wedge k \neq 0 \longrightarrow escrutinio[0] < escrutinio[0]) \wedge \\
  &(\forall j : \mathbb{Z}) \ (0 \leq j < 1 \wedge j \neq 0 \wedge j \neq 1 \longrightarrow escrutinio[0] < escrutinio[1])
  \end{aligned}$$

  Como $(0 \leq k < 1 \wedge k \neq 0)$ y $(0 \leq j < 1 \wedge j \neq 0)$ son contradicciones, ambas implicaciones son tautologías. ✓

Queda así probado que la precondición del ciclo implica el invariante propuesto. $P_C \longrightarrow I$

$\underline{\{I \wedge B_C\} \ C \ \{I\}}$

Queremos probar que tras una iteración del ciclo, el invariante sigue siendo verdadero. Esto es equivalente a probar:

$$I \wedge B_C \longrightarrow wp(C, I)$$

Empezamos calculando $wp(C, I) \equiv wp(C_1, wp(C_2, I))$

C_1

```
if(escrutinio[i] > escrutinio[primero]) then
    F1
else
    F2
endif
```

C_2

```
i := i + 1
```

De adentro hacia afuera, empezamos con $wp(C_2, I)$. Por el axioma 1, sabemos que:

$$wp(C_2, I) \equiv I^i_{i+1}$$

Reemplazamos correspondientemente:

$$\begin{aligned} I \;\equiv\; & 0 \;\leq\; primero,\; segundo \;<\; |escrutinio| - 1 \wedge \\ & 1 \leq i \leq |escrutinio| - 1 \wedge \\ & (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

$$\begin{aligned} wp(C_2, I) \;\equiv\; & 0 \;\leq\; primero,\; segundo \;< |escrutinio| - 1 \wedge \\ & 1 \leq i + 1 \leq |escrutinio| - 1 \wedge \\ & (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

Podemos simplificar la expresión como:

$$\begin{aligned} wp(C_2, I) \;\equiv\; & 0 \;\leq\; primero,\; segundo \;< |escrutinio| - 1 \wedge \\ & 0 \leq i \leq |escrutinio| - 2 \wedge \\ & (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo] \wedge \\ & escrutinioValido(escrutinio)) \end{aligned}$$

Ahora que ya calculamos $wp(C_2, I)$, podemos pasar a calcular $wp(C_1, wp(C_2, I))$

C_1 es un if con guarda B_F. Llamamos F_1 y F_2 a sus dos casos:

$$B_F \equiv (escrutinio[i] > escrutinio[primero])$$

F_1

```
segundo := primero
primero := i
```

F_2

```
if (escrutinio[i] > escrutinio[segundo]) then
    segundo := i
else
    skip
endif
```

Como C_1 es un if, por el axioma 4 sabemos que:

$$wp(C_1, wp(C_2, I)) \equiv def(B_F) \wedge ((B_F \wedge wp(F_1, wp(C_2, I))) \vee (\neg B_F \wedge wp(F_2, wp(C_2, I))))$$

Entonces, para calcular $wp(C_1, wp(C_2, I))$, necesitamos calcular $wp(F_1,\ wp(C_2,\ I))$ y $wp(F_2,\ wp(C_2,\ I))$.

Empezamos calculando $wp(F_1, wp(C_2, I))$

Aplicando el axioma 1, de asignación, sabemos que:

$$wp(F_1, wp(C_2, I)) \equiv ((wp(C_2, I))^{primero}_{i})^{segundo}_{primero}$$

Reemplazamos correspondientemente:

$$\begin{aligned} wp(C_2, I) \ \equiv\ & 0 \ \leq \ primero, \ segundo \ < |escrutinio| - 1 \wedge \\ & 0 \leq i \leq |escrutinio| - 2 \wedge \\ & (\forall k : \mathbb{Z}) \ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z}) \ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

$$\begin{aligned} wp(F_1, wp(C_2, I)) \ \equiv\ & 0 \ \leq \ i, \ primero \ < \ |escrutinio| - 1 \wedge \\ & 0 \leq i \leq |escrutinio| - 2 \wedge \\ & (\forall k : \mathbb{Z}) \ (0 \leq k < i + 1 \wedge k \neq i \longrightarrow escrutinio[k] < escrutinio[i]) \wedge \\ & (\forall j : \mathbb{Z}) \ (0 \leq j < i + 1 \wedge j \neq i \wedge j \neq primero \longrightarrow escrutinio[j] < escrutinio[primero])) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

Ahora calculamos $wp(F_2, wp(C_2, I)$.

F_2 es un if con guarda B_Z. Llamamos Z_1 y Z_2 a sus dos casos.

$$B_Z \equiv (escrutinio[i] > escrutinio[segundo])$$

Z_1

```
segundo := i
```

Z_2

```
skip
```

Como F_2 es un if, por el axioma 4 sabemos que:

$$wp(F_2, wp(C_2, I)) \equiv def(B_Z) \wedge ((B_Z \wedge wp(Z_1, wp(C_2, I))) \vee (\neg B_Z \wedge wp(Z_2, wp(C_2, I))))$$

Entonces, para calcular $wp(F_2, wp(C_2, I)$, necesitamos calcular $wp(Z_1, wp(C_2, I)$ y $wp(Z_2, wp(C_2, I)$

Empezamos calculando $wp(Z_1, wp(C_2, I)$.

Por el axioma 1, de asignación, sabemos que:

$$wp(Z_1, wp(C_2, I) \ \equiv \ (wp(C_2, I))_i^{segundo}$$

Reemplazamos correspondientemente:

$$\begin{aligned} wp(C_2, I) \ \equiv\ & 0 \ \leq \ primero, \ segundo \ < |escrutinio| - 1 \wedge \\ & 0 \leq i \leq |escrutinio| - 2 \wedge \\ & (\forall k : \mathbb{Z}) \ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z}) \ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

$$\begin{aligned} wp(Z_1, wp(C_2, I) \ \equiv\ & 0 \ \leq \ primero, \ i \ < |escrutinio| - 1 \wedge \\ & 0 \leq i \leq |escrutinio| - 2 \wedge \\ & (\forall k : \mathbb{Z}) \ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z}) \ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq i \longrightarrow escrutinio[j] < escrutinio[i]))) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

Ahora calculamos $wp(Z_2, wp(C_2, I))$.

Como $Z_2 \ \equiv$ **skip**, aplicando el axioma 2, sabemos que:

$$wp(Z_2, wp(C_2, I)) \ \equiv \ wp(C_2, I)$$

$$\begin{aligned} wp(Z_2, wp(C_2, I)) \equiv\ & 0 \le primero,\ segundo < |escrutinio| - 1 \wedge \\ & 0 \le i \le |escrutinio| - 2 \wedge \\ & (\forall k : \mathbb{Z})\ (0 \le k < i + 1 \wedge k \ne primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z})\ (0 \le j < i + 1 \wedge j \ne primero \wedge j \ne segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\ & escrutinioValido(escrutinio) \end{aligned}$$

Ya tenemos todas las partes de $wp(C, I)$. Podemos pasar a demostrar que $I \wedge B_C \longrightarrow wp(C, I)$.

***
***
***

Recapitulando:

$$wp(C, I) \equiv wp(C_1, wp(C_2, I))$$

$$wp(C, I) \equiv def(B_F) \wedge ((B_F \wedge wp(F_1, wp(C_2, I))) \vee (\neg B_F \wedge wp(F_2, wp(C_2, I))))$$

$wp(C, I) \equiv$

$$\equiv def(B_F) \wedge ((B_F \wedge wp(F_2, wp(C_2, I))) \vee (\neg B_F \wedge def(B_Z) \wedge ((B_Z \wedge wp(Z_1, wp(C_2, I))) \vee (\neg B_Z \wedge wp(Z_2, wp(C_2, I))))))$$

Queremos demostrar que:

$$I \wedge B_C \longrightarrow wp(C, I)$$

Esto es equivalente a probar:

$$\begin{gathered} I \wedge B_C \longrightarrow def(B_F) \\ \wedge \\ I \wedge B_C \longrightarrow (B_F \wedge wp(F_1, wp(C_2, I))) \vee (\neg B_F \wedge wp(F_2, wp(C_2, I))) \end{gathered}$$

Primero demostramos que $I \wedge B_C \longrightarrow def(B_F)$:

$$\begin{aligned} I \wedge B_C \equiv\ & 0 \le primero,\ segundo < |escrutinio| - 1 \wedge \\ & 1 \le i \le |escrutinio| - 1 \wedge \\ & (\forall k : \mathbb{Z})\ (0 \le k < i \wedge k \ne primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\ & (\forall j : \mathbb{Z})\ (0 \le j < i \wedge j \ne primero \wedge j \ne segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\ & i < |escrutinio| - 1 \wedge \\ & escrutinioValido(escrutinio) \\ & \Longrightarrow \\ def(B_F) \equiv\ & 0 \le i < |escrutinio| \wedge \\ & 0 \le primero < |escrutinio| \end{aligned}$$

Por partes:

- $$1 \le i \le |escrutinio| - 1 \longrightarrow 0 \le i < |escrutinio|$$

- $$0 \le primero,\ segundo < |escrutinio| - 1 \longrightarrow 0 \le primero < |escrutinio|$$

Ahora veamos la segunda implicación. Demostrarla es equivalente a demostrar:

$$\begin{gathered} I \wedge B_C \longrightarrow B_F \wedge wp(F_1, wp(C_2, I)) \\ \vee \\ I \wedge B_C \longrightarrow \neg B_F \wedge wp(F_2, wp(C_2, I)) \end{gathered}$$

Existen dos posibilidades: O se cumple B_F o se cumple $\neg B_F$. Podemos separar en casos asumiéndolas como verdaderas respectivamente:

$$B_F \equiv escrutinio[i] > escrutinio[primero]$$

Empezamos asumiendo B_F y demostramos:

$$I \land B_C \land B_F \longrightarrow B_F \land wp(F_1, wp(C_2, I))$$

$$\begin{aligned}
I \land B_C \land B_F \ \equiv\ & 0 \ \leq\ primero,\ segundo \ <\ |escrutinio| - 1\land \\
& 1 \leq i \leq |escrutinio| - 1\land \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i \land k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero])\land \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i \land j \neq primero \land j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo])\land \\
& i < |escrutinio| - 1 \\
& escrutinio[i] > escrutinio[primero]\land \\
& escrutinioValido(escrutinio) \\
& \Longrightarrow \\
B_F \land wp(F_1, wp(C_2, I)) \ \equiv\ & escrutinio[i] > escrutinio[primero]\land \\
& 0 \ \leq\ i,\ primero \ <\ |escrutinio| - 1\land \\
& 0 \leq i \leq |escrutinio| - 2\land \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \land k \neq i \longrightarrow escrutinio[k] < escrutinio[i])\land \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \land j \neq i \land j \neq primero \longrightarrow escrutinio[j] < escrutinio[primero]))\land \\
& escrutinioValido(escrutinio)
\end{aligned}$$

Vamos por partes:

- $$escrutinio[i] > escrutinio[primero] \quad\Longrightarrow\quad escrutinio[i] > escrutinio[primero]\ \checkmark$$

- $$\begin{aligned}
& 0 \ \leq\ primero,\ segundo \ <\ |escrutinio| - 1 \ \land \\
& 1 \leq i \leq |escrutinio| - 1 \ \land \\
& i < |escrutinio| - 1 \\
& \Longrightarrow \\
& 0 \ \leq\ i,\ primero \ <\ |escrutinio| - 1\ \checkmark
\end{aligned}$$

- $$1 \leq i \leq |escrutinio| - 1 \ \land\ i < |escrutinio| - 1 \quad\Longrightarrow\quad 0 \ \leq\ i \ \leq\ |escrutinio| - 2\ \checkmark$$

- $$\begin{aligned}
& (\forall k : \mathbb{Z})\ (0 \leq k < i \ \land\ k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \ \land \\
& escrutinio[i] > escrutinio[primero] \\
& \Longrightarrow \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \land k \neq i \longrightarrow escrutinio[k] < escrutinio[i])\ \checkmark \\
& (por\ transitividad)
\end{aligned}$$

- $$\begin{aligned}
& (\forall k : \mathbb{Z})\ (0 \leq k < i \ \land\ k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \\
& \Longrightarrow \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \land j \neq i \land j \neq primero \longrightarrow escrutinio[j] < escrutinio[primero]))\ \checkmark
\end{aligned}$$

  (Si consideramos que en la segunda expresión $j \neq i$, entonces la expresión es equivalente a la primera)

Queda así demostrado

$$I \land B_C \land B_F \longrightarrow B_F \land wp(F_1, wp(C_2, I))$$

Ahora continuamos asumiendo $\neg B_F$ y demostramos:

$$I \wedge B_C \wedge \neg B_F \longrightarrow \neg B_F \wedge wp(F_2, wp(C_2, I))$$

Recordemos que:

$$wp(F_2, wp(C_2, I)) \equiv def(B_Z) \wedge ((B_Z \wedge wp(Z_1, wp(C_2, I))) \vee (\neg B_Z \wedge wp(Z_2, wp(C_2, I))))$$

Por lo tanto, la implicación que queremos demostrar es equivalente a:

$$\begin{array}{c} I \wedge B_C \wedge \neg B_F \longrightarrow def(B_Z) \\ \wedge \\ I \wedge B_C \wedge \neg B_F \longrightarrow (B_Z \wedge wp(Z_1, wp(C_2, I))) \vee (\neg B_Z \wedge wp(Z_2, wp(C_2, I))) \end{array}$$

Primero demostramos que $I \wedge B_C \wedge \neg B_F \longrightarrow def(B_Z)$:

$$\begin{aligned} I \wedge B_C \wedge \neg B_F \ \equiv\ & 0 \ \leq\ primero,\ segundo\ <\ |escrutinio| - 1\wedge \\ & 1 \leq i \leq |escrutinio| - 1\wedge \\ & (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero])\wedge \\ & (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo])\wedge \\ & i < |escrutinio| - 1 \\ & escrutinio[i] \leq escrutinio[primero]\wedge \\ & escrutinioValido(escrutinio) \\ & \Longrightarrow \\ def(B_Z)\ \equiv\ & 0 \leq i < |escrutinio|\wedge \\ & 0 \leq segundo < |escrutinio| \end{aligned}$$

Vamos por partes:

- $$1 \leq i \leq |escrutinio| - 1 \ \longrightarrow\ 0 \leq i < |escrutinio| \ ✓$$

- $$0\ \leq\ primero,\ segundo\ <\ |escrutinio| - 1 \ \longrightarrow\ 0 \leq segundo < |escrutinio| \ ✓$$

Queda así demostrada la implicación.

Ahora demostramos la segunda implicación, que es equivalente a demostrar:

$$\begin{array}{c} I \wedge B_C \wedge \neg B_F \longrightarrow B_Z \wedge wp(Z_1, wp(C_2, I)) \\ \vee \\ I \wedge B_C \wedge \neg B_F \longrightarrow \neg B_Z \wedge wp(Z_2, wp(C_2, I)) \end{array}$$

Al igual que antes, existen dos posibilidades: O se cumple B_Z o se cumple $\neg B_Z$. Podemos separar en casos asumiéndolas como verdaderas respectivamente:

Empezamos asumiendo B_Z y demostramos:

$$I \wedge B_C \wedge \neg B_F \wedge B_Z \longrightarrow B_Z \wedge wp(Z_1, wp(C_2, I))$$

$$\begin{aligned}
I \wedge B_C \wedge \neg B_F \wedge B_Z \;\equiv\; & 0 \;\leq\; primero,\; segundo \;<\; |escrutinio| - 1 \wedge \\
& 1 \leq i \leq |escrutinio| - 1 \wedge \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& i < |escrutinio| - 1 \\
& escrutinio[i] \leq escrutinio[primero] \wedge \\
& escrutinio[i] > escrutinio[segundo] \wedge \\
& escrutinioValido(escrutinio) \\
& \;\Longrightarrow \\
B_Z \wedge wp(Z_1, wp(C_2, I)) \;\equiv\; & escrutinio[i] > escrutinio[segundo] \wedge \\
& 0 \;\leq\; primero,\; i \;<|escrutinio| - 1 \wedge \\
& 0 \leq i \leq |escrutinio| - 2 \wedge \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq i \longrightarrow escrutinio[j] < escrutinio[i]))) \wedge \\
& escrutinioValido(escrutinio)
\end{aligned}$$

Vamos por partes:

- $$escrutinio[i] > escrutinio[segundo] \;\longrightarrow\; escrutinio[i] > escrutinio[segundo]$$

- $$\begin{aligned}
& 0 \;\leq\; primero,\; segundo \;<|escrutinio| - 1 \;\wedge \\
& 1 \leq i \leq |escrutinio| - 1 \;\wedge \\
& i < |escrutinio| - 1 \\
& \;\Longrightarrow \\
& 0 \;\leq\; primero,\; i \;<|escrutinio| - 1 \checkmark
\end{aligned}$$

- $$1 \leq i \leq |escrutinio| - 1 \;\wedge\; i < |escrutinio| - 1 \;\longrightarrow\; 0 \;\leq\; i \;\leq |escrutinio| - 2 \checkmark$$

- $$\begin{aligned}
& (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero])) \;\wedge \\
& escrutinio[i] \leq escrutinio[primero] \;\wedge\; escrutinioValido(escrutinio) \\
& \;\Longrightarrow \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \checkmark
\end{aligned}$$

  (La primera fórmula nos dice que para 0 ≤ k < i se cumple que escrutinio[k] < escrutinio[primero]. Luego faltaría analizar qué pasa si k=i. La segunda fórmula indica que escrutinio[i] ≤ escrutinio[primero], por lo que quedaría demostrar que no es posible que escrutinio[i] = escrutinio[primero] . Como sabemos que por escrutinioVálido no hay repetidos en la lista, debe ocurrir que si escrutinio[i] = escrutinio[primero] entonces i=primero. Por último, tenemos en el consecuente que k ≠ primero, por lo cual k ≠ i. De esta manera, podemos afirmar que la única opción es que escrutinio[i] < escrutinio[primero]. Luego, la implicación es válida.)

- $$\begin{aligned}
& (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& escrutinio[i] > escrutinio[segundo] \\
& \;\Longrightarrow \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq i \longrightarrow escrutinio[j] < escrutinio[i])) \checkmark
\end{aligned}$$

(Como en el consecuente tenemos que 0 ≤ j <i+1 pero también que j ≠ i, esto es lo mismo que decir que 0≤j<i. La primera fórmula nos dice que para 0 ≤ j < i (y j ≠ primero y j ≠ segundo) se cumple que escrutinio[j] < escrutinio[segundo]. Como escrutinio[i] > escrutinio[segundo], tenemos que escrutinio[j] < escrutinio[segundo]< escrutinio[i]. Luego, por transitividad, es posible afirmar escrutinio[j] < escrutinio[i].)

Queda así demostrada la implicación.

Ahora asumimos $\neg B_Z$ y demostramos:

$$I \wedge B_C \wedge \neg B_F \wedge \neg B_Z \longrightarrow \neg B_Z \wedge wp(Z_2, wp(C_2, I))$$

$$\begin{aligned}
I \wedge B_C \wedge \neg B_F \wedge \neg B_Z \;\equiv\;& 0 \;\leq\; primero,\; segundo \;<\; |escrutinio| - 1 \wedge \\
& 1 \leq i \leq |escrutinio| - 1 \wedge \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& i < |escrutinio| - 1 \\
& escrutinio[i] \leq escrutinio[primero] \wedge \\
& escrutinio[i] \leq escrutinio[segundo] \wedge \\
& escrutinioValido(escrutinio) \\
& \Longrightarrow \\
\neg B_Z \wedge wp(Z_2, wp(C_2, I)) \;\equiv\;& escrutinio[i] \leq escrutinio[segundo] \wedge \\
& 0 \;\leq\; primero,\; segundo \;< |escrutinio| - 1 \wedge \\
& 0 \leq i \leq |escrutinio| - 2 \wedge \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& escrutinioValido(escrutinio)
\end{aligned}$$

Vamos por partes:

- $$escrutinio[i] \leq escrutinio[segundo] \;\longrightarrow\; escrutinio[i] \leq escrutinio[segundo] \;\checkmark$$

- $$0 \;\leq\; primero,\; segundo \;<\; |escrutinio| - 1 \;\longrightarrow\; 0 \;\leq\; primero,\; segundo \;<\; |escrutinio| - 1 \;\checkmark$$

- $$1 \leq i \leq |escrutinio| - 1 \;\wedge\; i < |escrutinio| - 1 \;\longrightarrow\; 0 \;\leq\; i \;\leq |escrutinio| - 2 \;\checkmark$$

- $$\begin{aligned}
& (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero])) \wedge \\
& escrutinio[i] \leq escrutinio[primero] \wedge \\
& escrutinioValido(escrutinio) \\
& \Longrightarrow \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i + 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \;\checkmark
\end{aligned}$$

  (Similar al del caso anterior)

- $$\begin{aligned}
& (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& escrutinio[i] \leq escrutinio[segundo] \wedge \\
& escrutinioValido(escrutinio) \\
& \Longrightarrow \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i + 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo])
\end{aligned}$$

(La primera fórmula nos dice que para $0 \leq$ j $<$ i se cumple que escrutinio[j] $<$ escrutinio[segundo]. Luego falta demostrar el caso donde j=i en el consecuente. La segunda fórmula indica que escrutinio[i] $\leq$ escrutinio[segundo], solo queda probar que escrutinio[i] $\neq$ escrutinio[segundo]. Como en el consecuente tenemos que j$\neq$ segundo y además por escrutinioValido sabemos que no hay repetidos en la lista, podemos afirmar que la única opción es que escrutinio[j] $<$ escrutinio[segundo]. Luego, la implicación es válida.)

Queda así demostrada la implicación.

Finalmente, habiendo probado todas estas implicaciones, queda demostrado:

$$I \wedge B_C \longrightarrow wp(C, I)$$

Por lo tanto, el invariante sigue valiendo después de una iteración del ciclo.

$$\{I \wedge B_C\}\ C\ \{I\}$$

$\underline{I \wedge \neg B_C \longrightarrow Q_C}$

Queremos probar que si se cumple el invariante y no la guarda del ciclo, entonces se cumple la postcondición del ciclo.

$$\begin{aligned}
I \wedge \neg B_C\ \equiv\ & 0 \leq primero,\ segundo < |escrutinio| - 1 \wedge \\
& 1 \leq i \leq |escrutinio| - 1 \wedge \\
& (\forall k : \mathbb{Z})\ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& i \geq |escrutinio| - 1 \wedge \\
& escrutinioValido(escrutinio) \\
& \Longrightarrow \\
Q_C\ \equiv\ & i = |escrutinio| - 1 \wedge \\
& 0 \leq primero, segundo < |escrutinio| - 1 \wedge \\
& escrutinio[primero] > escrutinio[segundo] \wedge \\
& (\forall j : \mathbb{Z})\ (0 \leq j < |escrutinio| - 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo])
\end{aligned}$$

Separo por partes:

- $$i \leq |escrutinio| - 1 \wedge i \geq |escrutinio| - 1 \longrightarrow i = |escrutinio| - 1 \checkmark$$

- $$0 \leq primero, segundo < |escrutinio| - 1 \longrightarrow 0 \leq primero, segundo < |escrutinio| - 1 \checkmark$$

- $$\begin{aligned}
& (\forall k : \mathbb{Z})\ (0 \leq k < |escrutinio| - 1 \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z})\ (0 \leq j < |escrutinio| - 1 \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \longrightarrow \\
& escrutinio[primero] > escrutinio[segundo] \checkmark
\end{aligned}$$

- Podemos deducir del antecedente:

  $$(1 \leq i \leq |escrutinio| - 1) \wedge (i \geq |escrutinio| - 1) \longrightarrow i = |escrutinio| - 1$$

  Reemplazamos i por $|escrutinio| - 1$ en el siguiente predicado del antecedente:

  $$(\forall j : \mathbb{Z})\ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo])$$

  Y podemos ver que es equivalente al predicado que nos quedaba por demostrar del consecuente.

Queda así probado que si se cumple el invariante y no la guarda del ciclo, entonces se cumple la postcondición del ciclo.

$$I \wedge \neg B_C \longrightarrow Q_C$$

Ahora que ya probamos la correctitud parcial del ciclo, pasamos a demostrar que termina, mediante el teorema de terminación. Como función variante proponemos:

$$fv \equiv (|escrutinio| - 1) - i$$

$\underline{\{I \wedge B_C \wedge fv = V_0\}\ C\ \{fv < V_0\}}$

Esto es equivalente a probar:

$$(I \wedge B_C \wedge fv = V_0) \longrightarrow wp(C,\ fv < V_0)$$

Empezamos encontrando $wp(C, fv < V_0)$:

$$wp(C, fv < V_0) \equiv wp(C_1, wp(C_2, fv < V_0))$$

Empezamos encontrando $wp(C_2, fv < V_0)$.

C_2

```
i := i + 1
```

Por el axioma 1, de asignación, sabemos que:

$$wp(C_2, fv < V_0) \equiv (fv < V_0)^i_{i+1}$$

Reemplazamos correspondientemente:

$$wp(C_2, fv < V_0) \equiv (|escrutinio| - 1) - (i + 1) < V_0$$

Simplificamos la expresión:

$$wp(C_2, fv < V_0) \equiv |escrutinio| - 2 - i < V_0$$

Ahora que ya hallamos $wp(C_2, fv < V_0)$, podemos calcular $wp(C_1, wp(C_2, fv < V_0)$.

C_1

```
if (escrutinio[i] > escrutinio[primero]) then
    F1
else
    F2
endif
```

Como C_1 es un if, por el axioma 4 sabemos que:

$$wp(C_1, wp(C_2, fv < V_0) \ \equiv \ def(B_F) \wedge ((B_F \wedge wp(F_1, wp(C_2, fv < V_0))) \vee (\neg B_F \wedge wp(F_2, wp(C_2, fv < V_0))))$$

Entonces necesitamos calcular $wp(F_1, wp(C_2, fv < V_0))$ y $wp(F_2, wp(C_2, fv < V_0))$.

Empezamos calculando $wp(F_1, wp(C_2, fv < V_0))$:

F_1

```
primero := i
segundo := primero
```

Aplicando el axioma 1, de asignación, y como en $wp(C_2, fv < V_0)$ no están presentes ni *segundo*, ni *primero*, entonces:

$$wp(F_1, wp(C_2, fv < V_0)) \equiv wp(C_2, fv < V_0)$$

Ahora calculamos $wp(F_2, wp(C_2, fv < V_0))$.

F_2

```
if (escrutinio[i] > escrutinio[segundo]) then
    segundo := i
else
    skip
endif
```

F_2 es un if. Llamamos B_Z a su guarda, y Z_1 y Z_2 a sus casos. Por el axioma 4 sabemos que:

$$wp(F_2, wp(C_2, fv < V_0)) \equiv def(B_Z) \land ((B_Z \land wp(Z_1, wp(C_2, fv < V_0))) \lor (\neg B_Z \land wp(Z_2, wp(C_2, fv < V_0))))$$

Necesitamos calcular $wp(Z_1, wp(C_2, fv < V_0))$ y $wp(Z_2, wp(C_2, fv < V_0))$.

Podemos notar que ambas son equivalentes a $wp(C_2, fv < V_0))$, por axiomas 1 y 2. Y por lo tanto:

$$wp(F_2, wp(C_2, fv < V_0)) \equiv def(B_Z) \land ((B_Z \land wp(C_2, fv < V_0)) \lor (\neg B_Z \land wp(C_2, fv < V_0)))$$

Por reglas de equivalencia lógica podemos simplificar la expresión como:

$$wp(F_2, wp(C_2, fv < V_0)) \equiv def(B_Z) \land (B_Z \lor \neg B_Z) \land wp(C_2, fv < V_0)$$

Notamos que el segundo predicado es una tautología, y por lo tanto podemos obviarlo y simplicar la expresión como:

$$wp(F_2, wp(C_2, fv < V_0)) \equiv def(B_Z) \land wp(C_2, fv < V_0)$$

Teniendo presentes todas estas cosas, podemos volver a expresar $wp(C,\ fv < V_0)$ de la siguiente manera:

$$wp(C,\ fv < V_0)\ \equiv\ def(B_F) \land ((B_F \land wp(C_2, fv < V_0)) \lor (\neg B_F \land def(B_Z) \land wp(C_2, fv < V_0)))$$

Por reglas de equivalencia lógica, podemos simplificar la expresión como:

$$wp(C,\ fv < V_0)\ \equiv\ def(B_F) \land wp(C_2, fv < V_0) \land (B_F \lor (\neg B_F \land def(B_Z)))$$

Y esto podemos simplificarlo más como:

$$wp(C,\ fv < V_0)\ \equiv\ def(B_F) \land wp(C_2, fv < V_0) \land (B_F \lor def(B_Z))$$

Por lo tanto, demostrar la implicación $I \land B_C \land fv = V_0 \longrightarrow wp(C,\ fv < V_0)$ es equivalente a demostrar:

$$\begin{gathered} I \land B_C \land fv = V_0 \longrightarrow def(B_F) \\ \land \\ I \land B_C \land fv = V_0 \longrightarrow wp(C_2, fv < V_0) \\ \land \\ I \land B_C \land fv = V_0 \longrightarrow (B_F \lor def(B_Z)) \end{gathered}$$

Demostramos la primera, $I \land B_C \land fv = V_0 \longrightarrow def(B_F)$:

$$\begin{aligned} I \land B_C \land fv = V_0 \equiv\ & 0\ \leq\ primero,\ segundo\ <\ |escrutinio| - 1 \land \\ & 1 \leq i \leq |escrutinio| - 1 \land \\ & (\forall k : \mathbb{Z})\ (0 \leq k < i \land k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \land \\ & (\forall j : \mathbb{Z})\ (0 \leq j < i \land j \neq primero \land j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \land \\ & i < |escrutinio| - 1 \land \\ & (|escrutinio| - 1) - i = V_0 \land \\ & escrutinioValido(escrutinio) \\ & \Longrightarrow \\ def(B_F)\ \equiv\ & 0 \leq i < |escrutinio| \land \\ & 0 \leq primero < |escrutinio| \end{aligned}$$

Por partes:

- $$1 \leq i \leq |escrutinio| - 1 \longrightarrow 0 \leq i < |escrutinio|$$

- $$0 \ \leq \ primero, \ segundo \ < \ |escrutinio| - 1 \longrightarrow 0 \leq primero < |escrutinio|$$

Ahora, demostramos la segunda, $I \wedge B_C \wedge fv = V_0 \longrightarrow wp(C_2, fv < V_0)$:

$$\begin{aligned}
I \wedge B_C \wedge fv = V_0 \equiv \ & 0 \ \leq \ primero, \ segundo \ < \ |escrutinio| - 1 \wedge \\
& 1 \leq i \leq |escrutinio| - 1 \wedge \\
& (\forall k : \mathbb{Z}) \ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z}) \ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& i < |escrutinio| - 1 \wedge \\
& (|escrutinio| - 1) - i = V_0 \wedge \\
& escrutinioValido(escrutinio) \\
& \ \Longrightarrow \\
wp(C_2, fv < V_0) \equiv \ & |escrutinio| - 2 - i < V_0
\end{aligned}$$

- $$(|escrutinio| - 1) - i = V_0 \longrightarrow |escrutinio| - 2 - i < V_0$$

  Reemplazando V_0 del lado derecho podemos ver fácilmente que se cumple:

  $$|escrutinio| - 2 - i < |escrutinio| - 1 - i$$

Por último, demostramos la tercera, $I \wedge B_C \wedge fv = V_0 \longrightarrow (B_F \vee def(B_Z))$:

$$\begin{aligned}
I \wedge B_C \wedge fv = V_0 \equiv \ & 0 \ \leq \ primero, \ segundo \ < \ |escrutinio| - 1 \wedge \\
& 1 \leq i \leq |escrutinio| - 1 \wedge \\
& (\forall k : \mathbb{Z}) \ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z}) \ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& i < |escrutinio| - 1 \wedge \\
& (|escrutinio| - 1) - i = V_0 \wedge \\
& escrutinioValido(escrutinio) \\
& \ \Longrightarrow \\
B_F \vee def(B_Z) \equiv \ & (escrutinio[i] > escrutinio[primero]) \vee \\
& (0 \leq i < |escrutinio| \wedge 0 \leq segundo < |escrutinio|)
\end{aligned}$$

- Podemos demostrar que se implica $def(B_Z)$ y por lo tanto se implica la disyunción referida:

  $$1 \leq i \leq |escrutinio| - 1 \longrightarrow 0 \leq i < |escrutinio|$$

  $$0 \ \leq \ primero, \ segundo \ < \ |escrutinio| - 1 \longrightarrow 0 \leq segundo < |escrutinio|$$

Queda así demostrado que

$$\{I \wedge B_C \wedge fv = V_0\} \ C \ \{fv < V_0\}$$

$(I \wedge fv \leq 0) \longrightarrow \neg B$

Como último paso, tenemos que demostrar que $fv \leq 0$ implica que el ciclo se acaba.

$$\begin{aligned}
I \wedge fv \leq 0 \ \equiv \ & 0 \leq primero, segundo < |escrutinio| - 1 \wedge \\
& 1 \leq i \leq |escrutinio| - 1 \wedge \\
& (\forall k : \mathbb{Z}) \ (0 \leq k < i \wedge k \neq primero \longrightarrow escrutinio[k] < escrutinio[primero]) \wedge \\
& (\forall j : \mathbb{Z}) \ (0 \leq j < i \wedge j \neq primero \wedge j \neq segundo \longrightarrow escrutinio[j] < escrutinio[segundo]) \wedge \\
& (|escrutinio| - 1) - i \leq 0 \wedge \\
& escrutinioValido(escrutinio) \\
& \ \Longrightarrow \\
\neg B \equiv \ & i \geq |escrutinio| - 1
\end{aligned}$$

Podemos notar que:

$$|escrutinio| - 1 - i \leq 0 \ \equiv \ |escrutinio| - 1 \leq i$$

Que al estar en conjunción queda:

$$i \geq escrutinio - 1 \wedge i \leq |escrutinio - 1| \longrightarrow i = |escrutinio| - 1$$

Y vemos que:

$$i = |escrutinio| - 1 \longrightarrow i \geq |escrutinio| - 1 \checkmark$$

Queda así demostrado que cuando la función variante vale 0 o menos, el ciclo acaba.

$$(I \wedge fv \leq 0) \longrightarrow \neg B$$

Habiendo demostrado que se cumplen las tres condiciones del teorema del invariante y las dos condiciones del teorema de terminación, queda demostrada la correctitud del ciclo.

$$\{P_C\}\ C\ \{Q_C\}$$

4.1.3. Final del programa

Finalmente, probamos la correctitud del final del programa: Vamos a probar $Q_C \longrightarrow wp(S_2, Q)$

Primero calculamos $wp(S_2,\ Q)$:

S_2

```
res := (primero, segundo)
```

Aplicando el axioma 1, de asignación, sabemos que:

$$wp(S_2,\ Q) \equiv (Q^{res_0}_{primero})^{res_1}_{segundo}$$

Hacemos los reemplazos correspondientes:

$$\begin{aligned} Q \ \equiv \ & 0 \leq res_0, res_1 < |escrutinio| - 1 \wedge \\ & escrutinio[res0] > escrutinio[reS_1] \wedge \\ & (\forall i : \mathbb{Z})\ (0 \leq i < |escrutinio| - 1 \wedge i \neq res0 \wedge i \neq reS_1 \longrightarrow escrutinio[i] < escrutinio[reS_1]) \end{aligned}$$

$$\begin{aligned} wp(S_2, Q) \ \equiv \ & 0 \leq primero, segundo < |escrutinio| - 1 \wedge \\ & escrutinio[primero] > escrutinio[segundo] \wedge \\ & (\forall i : \mathbb{Z})\ (0 \leq i < |escrutinio| - 1 \wedge i \neq primero \wedge i \neq segundo \longrightarrow escrutinio[i] < escrutinio[segundo]) \end{aligned}$$

Ahora podemos ver la implicación $Q_C \longrightarrow wp(S_2, Q)$:

$$\begin{aligned} Q_C \ \equiv \ & i = |escrutinio| - 1 \wedge \\ & 0 \leq primero, segundo < |escrutinio| - 1 \wedge \\ & escrutinio[primero] > escrutinio[segundo] \wedge \\ & (\forall i : \mathbb{Z})\ (0 \leq i < |escrutinio| - 1 \wedge i \neq primero \wedge i \neq segundo \longrightarrow escrutinio[i] < escrutinio[segundo]) \\ & \Longrightarrow \\ wp(S_2, Q) \ \equiv \ & 0 \leq primero, segundo < |escrutinio| - 1 \wedge \\ & escrutinio[primero] > escrutinio[segundo] \wedge \\ & (\forall i : \mathbb{Z})\ (0 \leq i < |escrutinio| - 1 \wedge i \neq primero \wedge i \neq segundo \longrightarrow escrutinio[i] < escrutinio[segundo]) \end{aligned}$$

Se puede ver a simple vista que es correcta la implicación.

Habiendo probado la correctitud de las tres partes del programa por separado, se puede concluir que nuestro programa es correcto con respecto a la especificación.

4.2. hayFraude

Recordemos la especificación:

`proc` `hayFraude` (in escrutinio_presidencial : $seq\langle\mathbb{Z}\rangle$, in escrutinio_senadores : $seq\langle\mathbb{Z}\rangle$, in escrutinio_diputados : $seq\langle\mathbb{Z}\rangle$) : Bool

`requiere` $\{|escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados|\}$

`requiere` $\{|escrutinio\_senadores| > 2\}$

`requiere` $\{escrutinioValido(escrutinio\_presidencial)\}$

`requiere` $\{escrutinioValido(escrutinio\_senadores)\}$

`requiere` $\{escrutinioValido(escrutinio\_diputados)\}$

`asegura` $\{res = true \iff \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores))\wedge$ $(totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados)))\}$

`aux` `totalEscrutinio` (escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ =

$$\sum_{i=0}^{|escrutinio|-1} escrutinio[i]\,;$$

El programa tiene tres partes:

S_1

```
totalEscP := 0
totalEscS := 0
totalEscD := 0
i := 0
```

C

```
while (i < escrutinioPresidencial.size()) do
    totalEscP := totalEscP + escrutinio_presidencial[i]
    totalEscS := totalEscS + escrutinio_senadores[i]
    totalEscD := totalEscD + escrutinio_diputados[i]
    i := i + 1
endwhile
```

S_2

```
res := false
if (totalEscP ≠ totalEscS ∨ totalEscP ≠ totalEscD) then
    res := true
else
    skip
endif
return res
```

Para probar la correctitud del programa, debemos probar la siguiente tripla de Hoare:

$$\{P\}\ S_1; C; S_2\ \{Q\}$$

Para ello, basta con probar que valen las siguientes tres cosas:

$$\{P\}\ S_1\ \{P_C\}$$
$$\{P_C\}\ C\ \{Q_C\}$$
$$\{Q_C\}\ S_1\ \{Q\}$$

4.2.1. Principio del programa

Probamos la correctitud del prinicpio de nuestro programa, representado por la tripla de Hoare:

$$\{P\}\ S_1\ \{P_C\}$$

Esto es equivalente a probar

$$P\ \longrightarrow\ wp(S_1,\ P_C)$$

P : condiciones del "requiere" de la especificación

$$P \equiv |escrutinio\_senadores| > 2 \wedge |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados|$$

S_1: código previo al ciclo

```
totalEscP := 0
totalEscS := 0
tootalEscD := 0
i := 0
```

P_C: Precondiciones del ciclo

$$\begin{aligned} P_C \ \equiv\ & i = 0 \wedge |escrutinio\_presidencial| > 2 \wedge \\ & |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados| \wedge \\ & totalEscP = 0 \wedge totalEscD = 0 \wedge totalEscS = 0 \end{aligned}$$

Empezamos calculando $wp(S_1, P_C)$, reemplazando en P_C con los valores de S_1:

$$\begin{aligned} wp(S_1, P_C) \ \equiv\ & 0 = 0 \wedge |escrutinio\_presidencial| > 2 \wedge \\ & |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados| \wedge \\ & 0 = 0 \wedge 0 = 0 \wedge 0 = 0 \end{aligned}$$

Simplificamos.

$$\begin{aligned} wp(S_1, P_C) \ \equiv\ & |escrutinio\_presidencial| > 2 \wedge \\ & |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados| \end{aligned}$$

Ahora que ya hallamos $wp(S_1, P_C)$, podemos probar que $P \longrightarrow wp(S_1, P_C)$:

$$\begin{aligned} P \ \equiv\ & |escrutinio\_senadores| > 2 \wedge \\ & |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados| \end{aligned}$$

Podemos notar a simple viste que $P \equiv wp(S_1, P_C)$, y por lo tanto, $P \longrightarrow wp(S_1, P_C)$.

Queda así demostrada la correctitud de la primera parte de nuestro programa.

$$\{P\}\ S_1\ \{P_C\}$$

4.2.2. Correctitud del Ciclo

$$\{P_C\}\ C\ \{Q_C\}$$

Para probar la correctitud del ciclo vamos a hacer uso del teorema del invariante y del teorema de terminación. Primero vamos con el invariante. Para deducir un posible invariante, vemos $P_C,\ B\ y\ Q_C$

$$\begin{aligned}
P_C \ &\equiv \ i = 0 \land |escrutinio_presidencial| > 2 \land \\
&\quad |escrutinio_presidencial| = |escrutinio_senadores| = |escrutinio_diputados| \land \\
&\quad totalEscP = 0 \land totalEscD = 0 \land totalEscS = 0
\end{aligned}$$

$$B \ \equiv \ i < |escrutinio\_presidencial|$$

$$\begin{aligned}
Q_C \ &\equiv \ i = |escrutinio_presidencial| = |escrutinio_senadores| = |escrutinio_diputados| \land \\
&\quad \sum_{j=0}^{|escrutinio_presidencial|-1} escrutinio_presidencial[j] = totalEscP \land \\
&\quad \sum_{j=0}^{|escrutinio_presidencial|-1} escrutinio_senadores[j] = totalEscS \land \\
&\quad \sum_{j=0}^{|escrutinio_presidencial|-1} escrutinio_diputados[j] = totalEscD
\end{aligned}$$

Nuestro posible invariante es el siguiente:

$$\begin{aligned}
I \ \equiv \ 0 \ \leq \ i \ \leq \ &|escrutinio_presidencial| \ \land \\
totalEscP \ &= \ \sum_{j=0}^{i-1} escrutinio_presidencial[j] \ \land \\
totalEscS \ &= \ \sum_{j=0}^{i-1} escrutinio_senadores[j] \ \land \\
totalEscD \ &= \ \sum_{j=0}^{i-1} escrutinio_diputados[j]
\end{aligned}$$

Procedemos a probar que cumple las tres condiciones del teorema:

1. $P_C \longrightarrow I$
2. $\{I \ \land \ B\} \ S \ \{I\}$
3. $I \ \land \ \neg \ B \ \longrightarrow \ Q_C$

$\underline{P_C \longrightarrow I}$

Queremos probar que la precondición del ciclo implica el invariante.

$$\begin{aligned}
P_C \ &\equiv \ i = 0 \ \land |escrutinio_presidencial| > 2 \land \\
&\quad |escrutinio_presidencial| = |escrutinio_senadores| = |escrutinio_diputados| \land \\
&\quad totalEscP = 0 \land totalEscD = 0 \land totalEscS = 0 \\
&\quad \Longrightarrow \\
I \ &\equiv \ 0 \ \leq \ i \ \leq \ |escrutinio_presidencial| \ \land \\
&\quad totalEscP \ = \ \sum_{j=0}^{i-1} escrutinio_presidencial[j] \ \land \\
&\quad totalEscS \ = \ \sum_{j=0}^{i-1} escrutinio_senadores[j] \ \land \\
&\quad totalEscD \ = \ \sum_{j=0}^{i-1} escrutinio_diputados[j]
\end{aligned}$$

Vamos por partes:

- $$i \ = \ 0 \ \longrightarrow \ 0 \ \leq \ i \ \leq \ |escrutinio\_presidencial|$$

- $$i \ = \ 0 \ \wedge \ totalEscP \ = \ 0 \ \longrightarrow \ totalEscP \ = \ \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \ \equiv \ \sum_{j=0}^{0-1} escrutinio\_presidencial[j] \ = \ 0$$

- $$i \ = \ 0 \ \wedge \ totalEscS \ = \ 0 \ \longrightarrow \ totalEscS \ = \ \sum_{j=0}^{i-1} escrutinio\_senadores[j] \ \equiv \ \sum_{j=0}^{0-1} escrutinio\_senadores[j] \ = \ 0$$

- $$i \ = \ 0 \ \wedge \ totalEscD \ = \ 0 \ \longrightarrow \ totalEscD \ = \ \sum_{j=0}^{i-1} escrutinio\_diputados[j] \ \equiv \ \sum_{j=0}^{0-1} escrutinio\_diputados[j] \ = \ 0$$

Queda probado que $P_C \ \longrightarrow \ I$ se cumple.

$\{I \ \wedge \ B\} \ C \ \{I\}$

Queremos probar que el invariante permanece válido tras una iteración del ciclo.
Esto es equivalente a probar

$$I \wedge B \longrightarrow wp(C, \ I)$$

Calculamos $wp(C, I)$

Con C:

$C_1 \ \equiv$ totalEscP := totalEscP + escrutinio_presidencial[i]
$C_2 \ \equiv$ totalEscS := totalEscS + escrutinio_senadores[i]
$C_3 \ \equiv$ totalEscD := totalEscD + escrutinio_diputados[i]
$C_4 \ \equiv$ i := i + 1

$$wp(C, I) \ \equiv \ wp(C_1; C_2; C_3; C_4, \ I) \ \equiv \ wp(C_1; wp(C_2; wp(C_3; wp(C_4, \ I)))) \equiv$$
$$(((I^{i}_{i+1})^{totalEscD}_{totalEscD+escrutinio\_diputados[i]})^{totalEscS}_{totalEscS+escrutinio\_senadores[i]})^{totalEscP}_{totalEscP+escrutinio\_presidencial[i]}$$

Aplicando los axiomas vistos en clase, reemplazamos los valores correspondientes.

$$\begin{aligned} I \ \equiv\ & 0 \ \leq\ i \ \leq\ |escrutinio\_presidencial| \ \wedge \\ & totalEscP \ = \ \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \ \wedge \\ & totalEscS \ = \ \sum_{j=0}^{i-1} escrutinio\_senadores[j] \ \wedge \\ & totalEscD \ = \ \sum_{j=0}^{i-1} escrutinio\_diputados[j] \end{aligned}$$

$$\begin{aligned} wp(C, I) \ \equiv\ & 0 \ \leq\ i \ +1 \ \leq\ |escrutinio\_presidencial| \ \wedge \\ & (totalEscP + escrutinio\_presidencial[i]) = \sum_{j=0}^{i+1-1} escrutinio\_presidencial[j] \wedge \\ & (totalEscS + escrutinio\_senadores[i]) = \sum_{j=0}^{i+1-1} escrutinio\_senadores[j] \wedge \\ & (totalEscD + escrutinio\_diputados[i]) = \sum_{j=0}^{i+1-1} escrutinio\_diputados[j] \end{aligned}$$

Podemos simplificar la expresión de $wp(C, I)$ teniendo en cuenta:

- $$0 \leq i + 1 \leq |escrutinio\_presidencial| \equiv -1 \leq i < |escrutinio\_presidencial|$$

- $$\begin{aligned} (totalEscP + escrutinio\_presidencial[i]) = \sum_{j=0}^{i+1-1} escrutinio\_presidencial[j] \equiv & \\ \equiv totalEscP = (\sum_{j=0}^{i} escrutinio\_presidencial[j]) - escrutinio\_presidencial[i] \equiv & \\ \equiv totalEscP = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] & \end{aligned}$$

- $$\begin{aligned} (totalEscS + escrutinio\_senadores[i]) = \sum_{j=0}^{i+1-1} escrutinio\_senadores[j] \equiv & \\ totalEscS = (\sum_{j=0}^{i} escrutinio\_senadores[j]) - escrutinio\_senadores[i] \equiv & \\ totalEscS = \sum_{j=0}^{i-1} escrutinio\_senadores[j] & \end{aligned}$$

- $$\begin{aligned} (totalEscD + escrutinio\_diputados[i]) = \sum_{j=0}^{i+1-1} escrutinio\_diputados[j] \equiv & \\ totalEscD = (\sum_{j=0}^{i} escrutinio\_diputados[j]) - escrutinio\_diputados[i] \equiv & \\ totalEscD = \sum_{j=0}^{i-1} escrutinio\_diputados[j] & \end{aligned}$$

La expresión simplificada de $wp(C,\ I)$ quedaría de la siguiente forma:

$$\begin{aligned} wp(C,I) \equiv\ & -1 \leq i < |escrutinio\_presidencial| \wedge \\ & totalEscP = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge \\ & totalEscS = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge \\ & totalEscD = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \end{aligned}$$

Ahora que ya hallamos $wp(C,I)$ procedemos a demostrar $I \wedge B \longrightarrow wp(C,I)$

Vemos primero $I \wedge B$:

$$\begin{aligned} I \wedge B \equiv\ & 0 \leq i \leq |escrutinio\_presidencial| \wedge \\ & totalEscP = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge \\ & totalEscS = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge \\ & totalEscD = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge \\ & i < |escrutinio\_presidencial| \end{aligned}$$

Podemos simplificar la expresión de $I \wedge B$ viendo que:

$$(0 \leq i \leq |escrutinio\_presidencial| \wedge i < |escrutinio\_presidencial|) \equiv 0 \leq i < |escrutinio\_presidencial|$$

$$\begin{aligned} I \wedge B \equiv\ & 0 \leq i < |escrutinio\_presidencial| \wedge \\ & totalEscP = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge \\ & totalEscS = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge \\ & totalEscD = \sum_{j=0}^{i} escrutinio\_diputados[j] \end{aligned}$$

Ahora, podemos comprobar $I \wedge B \longrightarrow wp(C,I)$. Todos los predicados se comprueban por equivalencia excepto:

$$-1 \leq i < |escrutinio\_presidencial| \longrightarrow 0 \leq i < |escrutinio\_presidencial|$$

Queda demostrado que $I \wedge B \longrightarrow wp(C,\ I)$.

$I \wedge \neg B \longrightarrow Q_C$

Queremos probar que si se cumple el invariante y no la guarda, entonces se cumple la postcondición del ciclo.

$$Q_C \equiv i = |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados| \wedge$$
$$\sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_presidencial[j] = totalEscP \wedge$$
$$\sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_senadores[j] = totalEscS \wedge$$
$$\sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_diputados[j] = totalEscD$$

$$I \wedge \neg B \equiv 0 \leq i \leq |escrutinio\_presidencial| \wedge$$
$$totalEscP = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge$$
$$totalEscS = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge$$
$$totalEscD = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge$$
$$\neg(i < |escrutinio\_presidencial|)$$

Podemos simplificar $I \wedge \neg B$ teniendo en cuenta:

- $$\neg(i < |escrutinio\_presidencial|) \equiv i \geq |escrutinio\_presidencial|$$
- $$0 \leq i \leq |escrutinio\_presidencial| \wedge i \geq |escrutinio\_presidencial| \equiv i = |escrutinio\_presidencial|$$

Simplificamos y queda:

$$I \wedge \neg B \equiv i = |escrutinio\_presidencial| \wedge$$
$$totalEscP = \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_presidencial[j] \wedge$$
$$totalEscS = \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_senadores[j] \wedge$$
$$totalEscD = \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_diputados[j] \wedge$$
$$i \geq |escrutinio\_presidencial|$$

Podemos ver que $I \wedge \neg B \equiv Q_C$, y por lo tanto, $I \wedge \neg B \longrightarrow Q_C$. Con esto demostramos la correctitud parcial del ciclo. ✓ Ahora falta probar que el ciclo termina.

Teorema de Terminación

Para probar que el ciclo termina, hacemos uso del teorema de terminación y proponemos como función variante:

$$fv \equiv |escrutinio\_presidencial| - i$$

Tenemos que probar que cumple con las siguientes dos condiciones:

$$\{I \wedge B \wedge fv = V_0\} \; C \; \{fv < V_0\}$$

$$I \wedge fv \leq 0 \longrightarrow \neg B$$

Empezamos probando que cumple con la primera.

$\{I \ \wedge \ B \ \wedge \ fv = V_0\} \ C \ \{fv < V_0\}$

Queremos probar que el valor de la función variante disminuye tras una iteración del ciclo. Esto es equivalente a probar:

$$I \ \wedge \ B \ \wedge \ fv = V_0 \ \longrightarrow \ wp(C, fv < V_0)$$

Calculamos $wp(C, \ fv < V_0) \equiv wp(C, \ |escrutinio\_presidencial| - i < V_0)$.

Con C:

$C_1 \ \equiv$ totalEscP := totalEscP + escrutinio_presidencial[i]
$C_2 \ \equiv$ totalEscS := totalEscS + escrutinio_senadores[i]
$C_3 \ \equiv$ totalEscD := totalEscD + escrutinio_diputados[i]
$C_4 \ \equiv$ i := i + 1

En este momento solo nos importa C_4, ya que las otras modifican variables que no están presentes.
Reemplazamos i por $i + 1$:

$$wp(C, \ fv < V_0) \ \equiv \ |escrutinio\_presidencial| - (i + 1) < V_0 \ \equiv \ |escrutinio\_presidencial| - i - 1 < V_0$$

Ahora que ya hallamos $wp(C, \ fv < V_0)$, podemos demostrar $I \wedge B \wedge fv = V_0 \longrightarrow wp(C, fv < V_0)$

$$\begin{aligned} I \wedge B \wedge fv = V_0 \ \equiv \ & 0 \leq i \leq |escrutinio\_presidencial| \wedge \\ & totalEscP = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge \\ & totalEscS = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge \\ & totalEscD = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge \\ & i < |escrutinio\_presidencial| \wedge \\ & V_0 = |escrutinio\_presidencial| - i \end{aligned}$$

Observamos que:

$$|escrutinio\_presidencial| - i - 1 < |escrutinio\_presidencial| - i$$

Teniendo en cuenta que:

$$|escrutinio\_presidencial| - i = V_0$$

Se comprueba:

$$I \wedge B \wedge fv = V_0 \longrightarrow |escrutinio\_presidencial| - i - 1 < V_0$$

Así, queda demostrado que el valor de la función variante disminuye tras una iteración del ciclo.

$$\{I \ \wedge \ B \ \wedge \ fv = V_0\} \ C \ \{fv < V_0\}$$

$I \wedge fv \leq 0 \longrightarrow \neg B$

Queremos probar que si se cumple el invariante y la función variante tiene un valor menor o igual a cero, entonces no se cumple la guarda y por ende, el ciclo termina.

$$\neg B \equiv \neg(i < |escrutinio\_presidencial|) \equiv i \geq |escrutinio\_presidencial|$$

$$
\begin{aligned}
I \wedge fv \leq 0 \;\equiv\; & 0 \leq i \leq |escrutinio\_presidencial| \wedge \\
& totalEscP = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge \\
& totalEscS = \sum_{j=0}^{i-1} escrutinio\_senadores[j] \wedge \\
& totalEscD = \sum_{j=0}^{i-1} escrutinio\_diputados[j] \wedge \\
& |escrutinio\_presidencial| - i \leq 0
\end{aligned}
$$

Podemos notar que:

$|escrutinio\_presidencial| - i \leq 0 \equiv |escrutinio\_presidencial| \leq i$

De esta manera queda demostrada la segunda condición del teorema de terminación.

Habiendo probado las tres condiciones del teorema del invariante y las dos condiciones del teorema de terminación, queda probada la correctitud del ciclo.

$$\{P_C\}\ C\ \{Q_C\}$$

4.2.3. Final del programa

Ahora probamos la correctitud de la última parte de nuestro programa, representado por la tripla de Hoare:

$$\{Q_C\}\ S_2\ \{Q\}$$

Demostrar que es válida es equivalente a demostrar:

$$Q_C \;\longrightarrow\; wp(S_2,\ Q)$$

Empezamos calculando $wp(S_2,\ Q)$

$$
\begin{aligned}
Q \;\equiv\; & res = true \iff \\
& \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores)) \wedge \\
& (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados))
\end{aligned}
$$

Recordemos que:

`aux totalEscrutinio` (escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ =
$\sum\limits_{i=0}^{|escrutinio|-1} escrutinio[i]$ **;**

S_2:

```
res := false

if (totalEscP ≠ totalEscS ∨ totalEscP ≠ totalEscD) then
    res := true
else
    skip
endif

return res
```

S_2 es una secuencia de instrucciones. Llamamos K_1 a la asignación y K_2 al if. Podemos obviar el return para la demostración. Aplicando el axioma 3, sabemos que:

$$wp(S_2,\ Q)\ \equiv\ wp(K_1,\ wp(K_2,\ Q)$$

Calculamos $wp(K_2, Q)$:

Como K_2 es un if, llamamos B a su guarda, y G_1 y G_2 a sus dos ramas.

G_1

```
res := true
```

G_2

```
skip
```

Sabemos que por axioma 4:

$$wp(K_2, Q)\ \equiv\ def(B) \land ((B \land wp(G_1, Q)) \lor (\neg B \land wp(G_2, Q)))$$

Entonces necesitamos calcular $wp(G_1, Q)$ y $wp(G_2, Q)$ por separado.

Calculamos primero $wp(G_1, Q)$:

Aplicando el axioma 1, de asignación, sabemos que:

$$wp(G_1, Q)\ \equiv\ Q^{res}_{true}$$

Reemplazamos correspondientemente:

$$\begin{aligned} Q\ \equiv\ & res = true \iff \\ & \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores))\land \\ & (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados) \end{aligned}$$

$$\begin{aligned} wp(G_1, Q)\ \equiv\ & true = true \iff \\ & \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores))\land \\ & (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados)) \end{aligned}$$

Podemos notar que $true = true$ es una tautología y por lo tanto podemos obviarla. Solo necesitamos probar que el segundo predicado vale.

$$\begin{aligned} wp(G_1, Q)\ \equiv\ & \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores))\land \\ & (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados))) \end{aligned}$$

Ahora calculamos $wp(G_2, Q)$, pero como G_2 es un **skip**, entonces:

$$wp(G_2, Q)\ \equiv\ Q$$

Ya tenemos todas las partes de $wp(K_2, Q)$, y podemos simplificar la expresión como:

$$wp(K_2, Q)\ \equiv\ def(B) \land ((B \land Q^{res}_{true}) \lor (\neg B \land Q))$$

***

Ahora que tenemos calculado $wp(K_2, Q)$, podemos calcular $wp(K_1,\ wp(K_2,\ Q))$.
Recordemos que K_1 es la asignación **res := false**, por lo tanto, aplicando el axioma 1, sabemos que:

$$wp(K_1,\ wp(K_2,\ Q))\ \equiv\ (def(B) \land ((B \land Q^{res}_{true}) \lor (\neg B \land Q)))^{res}_{false}$$

Observamos que tanto $def(B)$ como $(B \land Q^{res}_{true})$ no tienen variables res que puedan ser reemplazadas con $false$. En el predicado restante, $(\neg B \land Q)$, solo Q tiene variables reemplazables. Reemplazamos respectivamente:

$$\begin{aligned} Q\ \equiv\ & res = true \iff \\ & \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores))\land \\ & (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados)) \end{aligned}$$

$$\begin{aligned} Q^{res}_{false}\ \equiv\ & false = true \iff \\ & \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores))\land \\ & (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados)) \end{aligned}$$

Veamos que como $false = true$ es una contradicción, entonces equivale a la negación del predicado a la derecha del **iff**.

$$\begin{aligned} Q^{res}_{false} \;\equiv\;& totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores) \wedge \\ & (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados)) \end{aligned}$$

La forma final de nuestro $wp(S_2,\ Q)$ es la siguiente:

$$wp(S_2,\ Q) \;\equiv\; def(B) \wedge ((B \wedge Q^{res}_{true}) \vee (\neg B \wedge Q^{res}_{false}))$$

Es decir:

$$\begin{aligned} def(B) \;\equiv\;& def(totalEscP) \wedge def(totalEscS) \wedge def(totalEscD) \\ & \wedge \\ (B \wedge Q^{res}_{true}) \;\equiv\;& (totalEscP \neq totalEscS \vee totalEscP \neq totalEscD) \wedge \\ & \neg((totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores)) \wedge \\ & (totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados))) \\ & \vee \\ (\neg B \wedge Q^{res}_{false}) \;\equiv\;& totalEscP = totalEscS \wedge totalEscP = totalEscD \wedge \\ & totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_senadores) \wedge \\ & totalEscrutinio(escrutinio\_presidencial) = totalEscrutinio(escrutinio\_diputados) \quad) \end{aligned}$$

Podemos ahora demostrar la implicación $Q_C \;\longrightarrow\; wp(S_2,\ Q)$. Pero antes, veamos Q_C

$$\begin{aligned} Q_C \;\equiv\;& i = |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados| \wedge \\ & \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_presidencial[j] = totalEscP \;\; \wedge \\ & \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_senadores[j] = totalEscS \;\; \wedge \\ & \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_diputados[j] = totalEscD \end{aligned}$$

Podemos simplificar esta expresión usando el predicado especificado anteriormente, totalEscrutinio(escrutinio):

$$\begin{aligned} Q_C \;\equiv\;& i = |escrutinio\_presidencial| = |escrutinio\_senadores| = |escrutinio\_diputados| \wedge \\ & totalEscrutinio(escrutinio\_presidencial) = totalEscP \;\; \wedge \\ & totalEscrutinio(escrutinio\_senadores) = totalEscS \;\; \wedge \\ & totalEscrutinio(escrutinio\_diputados) = totalEscD \end{aligned}$$

Ahora asumiendo Q_C como verdadera, podemos reemplazar en $wp(S_2,\ Q)$, $totalEscrutinio(escrutinio\_presidencial[j])$ por $totalEscP$ y hacer lo mismo con $totalEscS$ y $totalEscD$.

$$\begin{aligned} def(B) \;\equiv\;& def(totalEscP) \wedge def(totalEscS) \wedge def(totalEscD) \\ & \wedge \\ (B \wedge Q^{res}_{true}) \;\equiv\;& (totalEscP \neq totalEscS \vee totalEscP \neq totalEscD \\ & \neg((totalEscP = totalEscP) \wedge (totalEscP = totalEscD)) \\ & \vee \\ (\neg B \wedge Q^{res}_{false}) \;\equiv\;& totalEscP = totalEscS \wedge totalEscP = totalEscD \wedge \\ & totalEscP = totalEscP \wedge totalEscP = totalEscD) \end{aligned}$$

Sabemos que $def(totalEscP) \wedge def(totalEscS) \wedge def(totalEscD)$ se cumplen, ya que son resultados de sumatorias ya demostradas como válidas. Y podemos ver que $B \equiv Q^{res}_{true}$ y que $\neg B \equiv Q^{res}_{false}$. Podemos entonces simplificar la expresión que nos queda por probar:

$$\begin{aligned}\equiv\ & (totalEscP \neq totalEscS \lor totalEscP \neq totalEscD)\\ & \lor \\ & (totalEscP = totalEscS \land totalEscP = totalEscD)\end{aligned}$$

Vemos que es una tautología y, por lo tanto, queda demostrada la correctitud de la última parte de nuestro programa.

Habiendo demostrado la correctitud de las tres partes de nuestro programa por separado, aplicando el corolario de monotonía, queda probada la correctitud del programa completo.