DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1: Especificación y WP

Elecciones nacionales

21 de marzo de 2024 — Algoritmos y Estructuras de Datos

Grupo 42

Integrante	LU	Correo electrónico
Bossi, Tomás	50/17	tomasbossi97@gmail.com
Dominguez, Rocio Julieta	798/22	rociodominguezcpm@gmail.com
Stabile, Delfina	819/22	delfistabile18@gmail.com
Ziger, Bruno Martín	218/23	ziger.bruno@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. Especificación

Aclaraciones sobre las especificaciones

- En todos los puntos, asumimos que no pueden haber empates entre partidos, ni tampoco entre cualquiera de los partidos y los votos en blanco, para todos los escrutinios. Esto lo implementamos con el predicado sinRepetidos, en los casos donde el funcionamiento del programa depende de esta condición.
- En la especificacion del problema hayBallotage, excluimos en los predicados la ultima posición mediante la función `subseq` para excluir los votos en blancos de los máximos (1.1).
- Decidimos aplicar el umbral del 3 % en el ejercicio 4 de la siguiente manera: para aquellos partidos cuyo porcentaje de votos no supere el umbral del 3 %, su fila en la matriz D'Hondt será una lista de ceros ("aplanando" los resultados pequeños) (1.4). Entonces el caso donde los coeficientes son 0 se excluye del requiere de que la matriz tenga todos los cocientes distintos. Con esto, somos consistentes y evitamos el problema de los votos en blanco en el ejercicio 5 (1.5).

1.1. hayBallotage

proc hayBallotage (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : Bool

requiere $\{|\text{escrutinio}| \geq 3 \land \texttt{sinRepetidos}(\text{escrutinio}) \land \texttt{sinNegativos}(\text{escrutinio})\}$

asegura $\{\text{res} = \text{false} \leftrightarrow \texttt{elMaximoTieneMasDeCuarentaYCinco}(\text{escrutinio}) \lor$
$(\texttt{elMaximoTieneMasDeCuarenta}(\text{escrutinio}) \land \texttt{elMaximoTieneDiferenciaMasDeDiez}(\text{escrutinio}))\}$

aux suma (in l : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ = $\sum_{i=0}^{|l|-1} l[i]$;

aux porcentajeDelTotal (in l : $seq\langle\mathbb{Z}\rangle$, in i : $\mathbb{Z}$) : $\mathbb{Z}$ = $100\frac{l[i]}{\texttt{suma}(l)}$;

pred sinNegativos (l: $seq\langle\mathbb{Z}\rangle$) {
  $(\forall i : \mathbb{Z})\ (0 \leq i < |l| \longrightarrow_L l[i] \geq 0)$
}

pred sinRepetidos (l: $seq\langle\mathbb{Z}\rangle$) {
  $(\forall i, j : \mathbb{Z})\ (0 \leq i < j < |l| \longrightarrow_L l[i] \neq l[j])$
}

pred esIdDelMaximo (l : $seq\langle\mathbb{Z}\rangle$, i : $\mathbb{Z}$) {
  $0 \leq i < |l| \land_L (\forall j : \mathbb{Z})\ (0 \leq j < |l| \longrightarrow_L l[j] \leq l[i])$
}

pred esIdDelSegundoMaximo (l : $seq\langle\mathbb{Z}\rangle$, i : $\mathbb{Z}$) {
  $0 \leq i < |l| \land_L (\neg\texttt{esIdDelMaximo}(l, i) \land (\forall j : \mathbb{Z})\ (0 \leq j < |l| \land \neg\texttt{esIdDelMaximo}(l, j) \longrightarrow_L l[j] \leq l[i]))$
}

pred elMaximoTieneMasDeCuarentaYCinco (l : $seq\langle\mathbb{Z}\rangle$) {
  $(\forall i : \mathbb{Z})\ (\texttt{esIdDelMaximo}(\texttt{subseq}(l, 0, |l| - 1), i) \longrightarrow_L \texttt{porcentajeDelTotal}(l, i) > 45)$
}

pred elMaximoTieneMasDeCuarenta (l : $seq\langle\mathbb{Z}\rangle$) {
  $(\forall i : \mathbb{Z})\ (\texttt{esIdDelMaximo}(\texttt{subseq}(l, 0, |l| - 1), i) \longrightarrow_L \texttt{porcentajeDelTotal}(l, i) > 40)$
}

pred elMaximoTieneDiferenciaMasDeDiez (l : $seq\langle\mathbb{Z}\rangle$) {
  $(\forall i, j : \mathbb{Z})$ (
    $\texttt{esIdDelMaximo}(\texttt{subseq}(l, 0, |l| - 1), i) \land \texttt{esIdDelSegundoMaximo}(\texttt{subseq}(l, 0, |l| - 1), j) \longrightarrow_L$
    $\texttt{porcentajeDelTotal}(l, i) > \texttt{porcentajeDelTotal}(l, j) + 10$
)
}

1.2. 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 $\{|\text{escrutinio\_presidencial}| = |\text{escrutinio\_senadores}| = |\text{escrutinio\_diputados}| > 0\}$

asegura $\{\text{res} = \text{false} \leftrightarrow (\texttt{suma}(\text{escrutinio\_presidencial}) = \texttt{suma}(\text{escrutinio\_senadores}) = \texttt{suma}(\text{escrutinio\_diputados}))\}$

aux suma (in l : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ = $\sum_{i=0}^{|l|-1} l[i]$;

1.3. obtenerSenadoresEnProvincia

$\texttt{proc obtenerSenadoresEnProvincia}$ (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}\times\mathbb{Z}$

$\quad\texttt{requiere}$ $\{|\text{escrutinio}| \geq 3 \land \texttt{sinRepetidos}(\text{escrutinio}) \land \texttt{sinNegativos}(\text{escrutinio})\}$

$\quad\texttt{asegura}$ $\{\texttt{esIdDelMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), res_0)\}$

$\quad\texttt{asegura}$ $\{\texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), res_1)\}$

$\quad\texttt{pred sinRepetidos}$ $(l : seq\langle\mathbb{Z}\rangle)$ {

$\quad\quad(\forall i, j : \mathbb{Z})\ (0 \leq i < j < |l| \longrightarrow_L l[i] \neq l[j])$

$\quad$}

$\quad\texttt{pred sinNegativos}$ $(l\text{: } seq\langle\mathbb{Z}\rangle)$ {

$\quad\quad(\forall i : \mathbb{Z})\ (0 \leq i < |l| \longrightarrow_L l[i] \geq 0)$

$\quad$}

$\quad\texttt{pred esIdDelMaximo}$ $(l : seq\langle\mathbb{Z}\rangle, i : \mathbb{Z})$ {

$\quad\quad 0 \leq i < |l| \land_L (\forall j : \mathbb{Z})\ (0 \leq j < |l| \longrightarrow_L l[j] \leq l[i])$

$\quad$}

$\quad\texttt{pred esIdDelSegundoMaximo}$ $(l : seq\langle\mathbb{Z}\rangle, i : \mathbb{Z})$ {

$\quad\quad 0 \leq i < |l| \land \neg\texttt{esIdDelMaximo}(l, i) \land (\forall j : \mathbb{Z})\ ((0 \leq j < |l| \land \neg\texttt{esIdDelMaximo}(l, j)) \longrightarrow_L l[j] \leq l[i])$

$\quad$}

1.4. calcularDHondtEnProvincia

$\texttt{proc calcularDHondtEnProvincia}$ (in cant_bancas : $\mathbb{Z}$, in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$

$\quad\texttt{requiere}$ $\{|\text{escrutinio}| \geq 2 \land \texttt{sinNegativos}(\text{escrutinio})\}$

$\quad\texttt{requiere}$ $\{(\forall i, j, k, l : \mathbb{Z})\ (0 \leq i, k < |\text{escrutinio}| - 1 \land 0 \leq j, l < \text{cant\_bancas} \longrightarrow_L (i = k \land j = l) \lor \frac{\text{escrutinio}[i]}{j+1} \neq \frac{\text{escrutinio}[k]}{l+1})\}$

$\quad\texttt{asegura}$ $\{|\text{res}| = |\text{escrutinio}| - 1\}$

$\quad\texttt{asegura}$ $\{(\forall i, j : \mathbb{Z})\ (0 \leq i < |\text{escrutinio} - 1| \longrightarrow_L |\text{res}[i]| = \text{cant\_bancas})\}$

$\quad\texttt{asegura}$ $\{(\forall i, j, k, l : \mathbb{Z})\ (0 \leq i, k < |\text{escrutinio}| - 1 \land 0 \leq j, l < \text{cant\_bancas} \longrightarrow_L (i = k \land j = l) \lor \text{res}[i][j] = 0 \lor \text{res}[i][j] \neq \text{res}[k][l])\}$

$\quad\texttt{asegura}$ $\{(\forall i, j : \mathbb{Z})\ (0 \leq i < |\text{escrutinio}| - 1 \land 0 \leq j < \text{cant\_bancas} \longrightarrow_L \text{res}[i][j] = \texttt{DHont}(\text{escrutinio}, \text{escrutinio}[i], j))\}$

$\quad\texttt{aux DHondt}$ (in escrutinio : $seq\langle\mathbb{Z}\rangle$, in votos : $\mathbb{Z}$, in cociente : $\mathbb{Z}$) : $\mathbb{R} = \text{if } 100\frac{\text{votos}}{\texttt{suma}(\text{escrutinio})} > 3 \text{ then } \frac{\text{votos}}{\text{cociente}+1} \text{ else } 0 \text{ fi}$;

$\quad\texttt{pred sinNegativos}$ $(l : seq\langle\mathbb{Z}\rangle)$ {

$\quad\quad(\forall i : \mathbb{Z})\ (0 \leq i < |l| \longrightarrow_L l[i] \geq 0)$

$\quad$}

1.5. obtenerDiputadosEnProvincia

$\texttt{proc obtenerDiputadosEnProvincia}$ (in cant_bancas : $\mathbb{Z}$, in escrutinio : $seq\langle\mathbb{Z}\rangle$, in dHondt : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) : $seq\langle\mathbb{Z}\rangle$

$\quad\texttt{requiere}$ $\{|\text{dHondt}| > 0 \land \text{cant\_bancas} > 0 \land \texttt{sinNegativos}(\text{dHondt}) \land \texttt{noTodosCeros}(\text{dHondt})\}$

$\quad\texttt{requiere}$ $\{(\forall i, j, k, l : \mathbb{Z})\ (0 \leq i, k < |\text{dHondt}| \land 0 \leq j, l < \text{cant\_bancas} \longrightarrow_L (i = k \land j = l) \lor \text{dHondt}[i][j] = 0 \lor \text{dHondt}[i][j] \neq \text{dHondt}[k][l])\}$

$\quad\texttt{asegura}$ $\{|\text{res}| = |\text{dHondt}|\}$

$\quad\texttt{asegura}$ $\{\texttt{suma}(\text{res}) = \text{cant\_bancas}\}$

$\quad\texttt{asegura}$ $\{(\forall i : \mathbb{Z})\ (\text{res}[i] = \texttt{cuantosSuperan}(\text{dHondt}, i, \text{cant\_bancas}))\}$

$\quad\texttt{aux suma}$ (in $l : seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z} = \sum_{i=0}^{|l|-1} l[i]$;

$\quad\texttt{aux posicion}$ (in $M : seq\langle seq\langle\mathbb{Z}\rangle\rangle$, in $t : \mathbb{Z}$) : $\mathbb{Z} = \sum_{i=0}^{|M|-1}\sum_{j=0}^{|M[i]|-1} \text{if } M[i][j] \geq t \text{ then } 1 \text{ else } 0 \text{ fi}$;

$\quad\texttt{aux cuantosSuperan}$ (in $M : seq\langle seq\langle\mathbb{Z}\rangle\rangle$, in $i : \mathbb{Z}$, in cant_bancas : $\mathbb{Z}$) : $\mathbb{Z} = \sum_{j=0}^{|M[i]|-1} \text{if } \texttt{posicion}(M, M[i][j]) \leq \text{cant\_bancas} \text{ then } 1 \text{ else } 0 \text{ fi}$;

$\quad\texttt{pred sinNegativos}$ $(M : seq\langle seq\langle\mathbb{Z}\rangle\rangle)$ {

$\quad\quad(\forall i, j : \mathbb{Z})\ ((0 \leq i < |M| \land_L 0 \leq j < |M[i]|) \longrightarrow_L M[i][j] \geq 0)$

$\quad$}

$\quad\texttt{pred noTodosCeros}$ $(M : seq\langle seq\langle\mathbb{Z}\rangle\rangle)$ {

$\quad\quad(\exists i, j : \mathbb{Z})\ (0 \leq i < |M| \land_L 0 \leq j < |M[i]| \land_L M[i][j] > 0)$

$\quad$}

1.6. 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` $\{cant\_bancas > 0 \land |listas| > 0\}$

`requiere` $\{(\forall i : \mathbb{Z})\ (0 \leq i < |listas| \longrightarrow_L (\texttt{tieneDnisValidos}(listas[i]) \land \texttt{tieneGenerosValidos}(listas[i])))\}$

`asegura` $\{res = true \leftrightarrow (\forall i : \mathbb{Z})\ (0 \leq i < |listas| \longrightarrow_L |listas[i]| = cant\_bancas \land \texttt{alternaGeneros}(lista[i]))\}$

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

$(\forall i : \mathbb{Z})\ (0 \leq i < |lista| \longrightarrow_L (lista[i]_0 > 0))$

}

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

$(\forall i : \mathbb{Z})\ (0 \leq i < |lista| \longrightarrow_L (lista[i]_1 = 1 \lor lista[i]_1 = 2))$

}

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

$(\forall i, j : \mathbb{Z})\ (0 \leq i, j < |lista| \longrightarrow_L lista[i]_2 = lista[j]_2 \leftrightarrow i\,\%2 = j\,\%2)$

}

2. Implementaciones

2.1. hayBallotage

```
i := 0;
total_votos := 0;
maximo_sin_blanco := 0;
segundo_sin_blanco := 0;

while (i < |escrutinio|) do
    suma := suma + escrutinio[i]
    if (i < |escrutinio| - 1 && escrutinio[i] > maximo_sin_blanco) then
        segundo_sin_blanco := maximo_sin_blanco
        maximo_sin_blanco := escrutinio[i]
    else
        if (escrutinio[i] > segundo_sin_blanco) then
            segundo_sin_blanco := escrutinio[i]
        else
            skip
        endif;
    endif;
    i := i + 1
endwhile;

porcentaje_maximo := 100 * maximo_sin_blanco / total_votos;
porcentaje_segundo := 100 * segundo_sin_blanco / total_votos;

if (porcentaje_maximo > 45 || (porcentaje_maximo > 40 && porcentaje_maximo - porcentaje_segundo > 10)) then
    res := false
else
    res := true
endif;
```

2.2. hayFraude

```
i := 0;
suma_presidencial := 0;
suma_senadores := 0;
suma_diputados := 0;

while (i < |escrutinio_presidencial|) do
    suma_presidencial := suma_presidencial + escrutinio_presidencial[i]
    suma_senadores := suma_senadores + escrutinio_senadores[i]
    suma_diputados := suma_diputados + escrutinio_diputados[i]
    i := i + 1
endwhile;

if (suma_presidencial == suma_senadores && suma_senadores == suma_diputados) then
    res := false
else
    res := true
endif;
```

2.3. obtenerSenadoresEnProvincia

```
if (escrutinio[0] > escrutinio[1]) then
    indice_maximo = 0
    indice_segundo = 1
else
    indice_maximo = 1
    indice_segundo = 0
endif;
i := 2;
while (i < |escrutinio| - 1) do
    if (escrutinio[i] > escrutinio[indice_maximo]) then
        indice_segundo := indice_maximo
        indice_maximo := i
    else
        if (escrutinio[i] > escrutinio[indice_segundo]) then
            indice_segundo := i
        else
            skip
        endif;
    endif;
    i := i + 1
endwhile;

res := (indice_maximo, indice_segundo);
```

2.4. validarListasDiputadosEnProvincia

```
res := true;
i := 0;

while (i < |listas|) do
    lista := listas[i]

    if (|lista| != cant_bancas) then
        res := false
    else
        skip
    endif;

    j := 0
    while (j < |lista| - 1) do
        if (lista[j][1] == lista[j+1][1]) then
            res := false
        else
            skip
        endif;
        j := j + 1;
    endwhile;

    i := i + 1;
endwhile;
```

3. Demostraciones formales de correctitud

Aclaraciones sobre las demostraciones

- Usamos la notación $Q^{b_1,b_2,\ldots,b_k}_{a_1,a_2,\ldots,a_k}$ indicando que estamos reemplazando muchas variables libres de Q secuencialmente, desde $a_i := b_1$ hasta $a_k := b_k$.

3.1. hayFraude

Nombremos los predicados que vamos a utilizar para el método de la weakest precondition:

- El auxiliar para sumar los elementos de una lista: $\texttt{suma}(l) = \sum_{i=0}^{|l|-1} l[i]$
- Precondición: $P = \{|\text{escrutinio\_presidencial}| = |\text{escrutinio\_senadores}| = |\text{escrutinio\_diputados}| > 0\}$
- Postcondición: $Q = \{\text{res} = \text{false} \leftrightarrow A\}$, donde $A = \{\texttt{suma}(\text{escrutinio\_presidencial}) = \texttt{suma}(\text{escrutinio\_senadores}) = \texttt{suma}(\text{escrutinio\_diputados})\}$
- Condición del ciclo: $B = \{i < |\text{escrutinio\_presidencial}|\}$
- Condición del if: $C = \{\text{suma\_presidencial} = \text{suma\_senadores} = \text{suma\_diputados}\}$
- Invariantes: $I_1 = \{0 \leq i \leq |\text{escrutinio\_presidencial}| \wedge_L \text{suma\_presidencial} = \sum_{k=0}^{i-1} \text{escrutinio\_presidencial}[k]\}$
- $I_2 = \{0 \leq i \leq |\text{escrutinio\_diputados}| \wedge_L \text{suma\_diputados} = \sum_{k=0}^{i-1} \text{escrutinio\_diputados}[k]\}$
- $I_3 = \{0 \leq i \leq |\text{escrutinio\_senadores}| \wedge_L \text{suma\_senadores} = \sum_{k=0}^{i-1} \text{escrutinio\_senadores}[k]\}$

Para demostrar la correctitud del ciclo, proponemos:

- $P_c = \{i = 0 \wedge \text{suma\_presidencial} = 0 \wedge \text{suma\_diputados} = 0 \wedge \text{suma\_senadores} = 0 \wedge P\}$
  Por ser razonable que P_c sea la conjunción entre P y el estado resultante de las instrucciones previas al ciclo.
- $I = \{P \wedge I_1 \wedge I_2 \wedge I_3\}$
  Porque refleja lo que desde la intuición comprendemos que debe ocurrir al principio, durante (luego de cada iteración) y al final de la ejecución del ciclo.

- $Q_c = \{\text{suma\_presidencial} = \texttt{suma}(\text{escrutinio\_presidencial}) \wedge \text{suma\_senadores} = \texttt{suma}(\text{escrutinio\_senadores}) \wedge \text{suma\_diputados} = \texttt{suma}(\text{escrutinio\_diputados})\}$.

  Porque esto garantiza que las variables suma_presidencial, suma_senadores y suma_diputados contienen la información que deberían.

- $f_v = |\text{escrutinio\_presidencial}| - i$
  Porque fv decrece en cada iteración del ciclo y se hace 0 luego de la última, cuando $i = |\text{escrutinio\_presicencial}|$.

Primero vamos a demostrar la correctitud del if que le sigue al ciclo, al cual llamamos S_{if}. Para ello, calculamos la wp de S_{if} con la postcondición Q. Recordar que llamamos C a la condición de este if.

Aplicando los axiomas vistos en clase sobre la wp:

$$\text{wp}(\textsf{if } C \textsf{ then } \text{res} := \text{false} \textsf{ else } \text{res} := \text{true } \textsf{fi}, Q) \equiv \text{def}(C) \wedge_L ((C \wedge \text{wp}(res := \text{false}, Q)) \vee (\neg C \wedge \text{wp}(res := \text{true}, Q)))$$

Y repetimos lo mismo en las wp de cada asignación por separado:

$$\text{wp}(\text{res} := \text{false}, Q) \equiv \text{def}(\text{false}) \wedge Q^{\text{res}}_{\text{false}} \equiv \{\text{false} = \text{false} \leftrightarrow A\} \equiv A$$

$$\text{wp}(\text{res} := \text{true}, Q) \equiv \text{def}(\text{true}) \wedge Q^{\text{res}}_{\text{true}} \equiv \{\text{true} = \text{false} \leftrightarrow A\} \equiv \neg A$$

Combinando:

$$\text{wp}(\textsf{if } C \textsf{ then } \text{res} := \text{false} \textsf{ else } \text{res} := \text{true } \textsf{fi}, Q) \equiv \{(C \wedge A) \vee (\neg C \wedge \neg A)\} \equiv \{C \leftrightarrow A\}$$

Para demostrar la correctitud de $\{Q_c\}S_{\text{if}}\{Q\}$ basta ver que $Q_c \longrightarrow \text{wp}(S_{\text{if}}, Q) = \{C \leftrightarrow A\}$. Pero Q_c implica que $\text{suma\_presidencial} = \texttt{suma}(\text{escrutinio\_presidencial})$, etc., y reemplazando estas igualdades en C se obtiene A y viceversa, con lo cual queda probado que $Q_c \longrightarrow \{C \leftrightarrow A\}$.

Para probar la correctitud parcial de $\{P_c\}S_c\{Q_c\}$, por el Teorema del Inviarante, basta demostrar que:

1. $P_c \longrightarrow I$
2. $\{I \wedge B\}S_c\{I\}$
3. $I \wedge \neg B \longrightarrow Q_c$

Demostremos cada ítem individualmente

1. Notar que
   $$P_c \longrightarrow i = 0 \longrightarrow 0 \leq i \leq |\text{escrutinio\_presidencial}|$$
   pues $P_c \longrightarrow P$ y por P, vale $|\text{escrutinio\_presidencial}| > 0$. Por otro lado,
   $$P_c \longrightarrow i = 0 \wedge \text{suma\_presidencial} = 0 \longrightarrow \sum_{k=0}^{i-1} \text{escrutinio\_presidencial}[i] = 0$$
   dado que es la suma vacía. Estas dos cosas implican I_1. Ídem para $P_c \longrightarrow I_2$ y $P_c \longrightarrow I_3$. Además, $P_c \longrightarrow P$. Uniendo todo, $P_c \longrightarrow I$.

2. Queremos ver que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$, para lo cual calcularemos esta última wp.
   $$\text{wp}(S_c, I) \equiv \text{wp}(S1_c; S2_c; S3_c; S4_c, I) \equiv \text{wp}(S1_c, \text{wp}(S2_c, \text{wp}(S3_c, \text{wp}(S4_c, I))))$$
   donde $S1_c$, $S2_c$, $S3_c$ y $S4_c$ son las instrucciones dentro del ciclo.
   $$\text{wp}(S4_c, I) \equiv \text{def}(i) \wedge_L I^i_{i+1} \equiv I^i_{i+1} \equiv Q_4$$
   $$\text{wp}(S3_c, Q_4) \equiv \text{def}(sd + ed[i]) \wedge_L Q4^{sd}_{sd+ed[i]} \equiv 0 \leq i < |ed| \wedge_L Q4^{sd}_{sd+ed[i]} \equiv Q_3$$
   $$\text{wp}(S2_c, Q_3) \equiv \text{def}(ss + es[i]) \wedge_L Q3^{ss}_{ss+es[i]} \equiv 0 \leq i < |es| \wedge_L Q3^{ss}_{ss+es[i]} \equiv Q_2$$
   $$\text{wp}(S1_c, Q_2) \equiv \text{def}(sp + ep[i]) \wedge_L Q2^{sp}_{sp+ep[i]} \equiv 0 \leq i < |ep| \wedge_L Q2^{sp}_{sp+ep[i]} \equiv Q_1$$
   donde sd = suma_diputados, ed = escrutinio_diputados, ss = suma_senadores, es = escrutinio_senadores, sp = suma_presidencial y ep = escrutinio_presidencial. Es decir, la wp es equivalente a las tres condiciones de los rangos y a I reemplazando i por $i+1$ y agregando el nuevo término a cada sumatoria. Es decir:
   $$\text{wp}(S_c, I) \equiv \{0 \leq i < |ep|, |ed|, |es| \wedge_L |ep| = |ed| = |es| \wedge$$
   $$(0 \leq i + 1 \leq |\text{ep}| \wedge_L \text{sp+ep[i]} = \sum_{k=0}^{i} \text{ep}[k]) \wedge$$

$$(0 \leq i+1 \leq |\text{ed}| \wedge_L \text{sd+ed[i]} = \sum_{k=0}^{i} \text{ed}[k]) \wedge$$

$$(0 \leq i+1 \leq |\text{es}| \wedge_L \text{ss+es[i]} = \sum_{k=0}^{i} \text{es}[k]\}$$

Notar que $0 \leq i < |ep|, |ed|, |es| \longrightarrow (0 \leq i+1 \leq |\text{ep}|) \wedge (0 \leq i+1 \leq |\text{ed}|) \wedge (0 \leq i+1 \leq |\text{es}|)$, por lo que podemos quitar estas condiciones de la wp, dado que son redundantes. Por otro lado, notar que

$$sp + ep[i] = \sum_{k=0}^{i} ep[k] \iff sp = \sum_{k=0}^{i-1} ep[k].$$

Operando similarmente con las otras variables, podemos obtener que

$$\text{wp}(S_c, I) \equiv \{0 \leq i < |ep| = |ed| = |es| \wedge_L \text{sp} = \sum_{k=0}^{i-1} \text{ep}[k] \wedge \text{sd} = \sum_{k=0}^{i-1} \text{ed}[k]) \wedge \text{ss} = \sum_{k=0}^{i-1} \text{es}[k]\}$$

Solo queda ver que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$. Sabemos que

$$I \longrightarrow 0 \leq i \leq |ep| = |ed| = |es| \wedge \text{sp} = \sum_{k=0}^{i-1} \text{ep}[k] \wedge \text{sd} = \sum_{k=0}^{i-1} \text{ed}[k] \wedge \text{ss} = \sum_{k=0}^{i-1} \text{es}[k]$$

Por otro lado, $B = \{i < |ep|\}$, entonces $I \wedge B$ implican $\{0 \leq i < |ep| = |ed| = |es|\}$. Queda demostrado, entonces, que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ y por lo tanto se cumple $\{I \wedge B\}S_c\{I\}$.

3. Puede verse que $I \wedge \neg B \longrightarrow \{i = |ep| = |ed| = |es|\}$, lo que a su vez implica:

   - $\text{sp} = \sum_{k=0}^{i-1} \text{ep}[k] = \texttt{suma}(\text{ep})$
   - $\text{sd} = \sum_{k=0}^{i-1} \text{ed}[k] = \texttt{suma}(\text{ed})$
   - $\text{ss} = \sum_{k=0}^{i-1} \text{es}[k] = \texttt{suma}(\text{es})$

   Como Q_c es la conjunción de estos tres predicados, queda claro que $I \wedge \neg B \longrightarrow Q_c$.

Ahora, tenemos que probar la terminación del ciclo. Para eso, por el Teorema de Terminación de Ciclos, basta con demostrar:

1. $\{I \wedge B \wedge f_v = v_0\}S_c\{f_v < v_0\}$
2. $I \wedge (f_v \leq 0) \longrightarrow \neg B$

Veamos que esto es cierto:

1. Queremos ver que $\{I \wedge B \wedge f_v = v_0\} \longrightarrow \text{wp}(S_c, f_v < v_0)$, para lo cual calcularemos esta última wp.

   $$\text{wp}(S_c, f_v < v_0) \equiv \text{def}(i+1) \wedge \text{wp}(S1_c; S2_c; S3_c; S4_c, f_v < v_0) \equiv \text{wp}(S1_c, \text{wp}(S2_c, \text{wp}(S3_c, \text{wp}(S4_c, f_v < v_0)))$$

   En particular, podemos calcular

   $$\text{wp}(S4_c, f_v < v_0))) \equiv \text{wp}(i := i+1, |ep| - i < v_0))) \equiv \{|ep| - i - 1 < v_0\} \equiv \{f_v < v_0 + 1\}$$

   Como las primeras tres líneas no alteran ninguna variable libre de la postcondición, sus wp son equivalentes a sus postcondiciones (las wp de las líneas siguientes). Por lo tanto, escribimos:

   $$\text{wp}(S_c, f_v < v_0) \equiv 0 \leq i < |ep|, |ed|, |es| \wedge_L f_v < v_0 + 1$$

   dado que $\text{def}(sp + ep[i]) \leftrightarrow 0 \leq i < |ep|$ y similarmente para las otras variables.

   Queda entonces ver que $\{I \wedge B \wedge f_v = v_0\} \longrightarrow \text{wp}(S_c, f_v < v_0)$:

   - $f_v = v_0$ implica $f_v < v_0 + 1$.
   - $I \wedge B$ implica $0 \leq i < |ep|, |ed|, |es|$.

   Entonces queda demostrado que $\{I \wedge B \wedge f_v = v_0\} \longrightarrow \text{wp}(S_c, f_v < v_0)$ y por lo tanto se cumple la tripla de Hoare $\{I \wedge B \wedge f_v = v_0\}S_c\{f_v < v_0\}$.

2. Dado que $f_v \leq 0 \leftrightarrow |ep| \leq i$, se ve que $I \wedge f_v \leq 0$ implica $\neg(i < |ep|) \equiv \neg B$.

Habiendo demostrado la correctitud parcial y la terminación del ciclo, queda demostrada la correcitud total con la precondición de ciclo P_c, el invariante I y la función variante f_v propuestos.

Solo queda ver que $P \longrightarrow \text{wp}(S1; S2; S3; S4, P_c)$, donde S1 a S4 son las primeras cuatro instrucciones del programa. Para ello, podemos calcular esta última wp. Como todas las opraciones en las instrucciones S1 a S4 están bien definidas,

$$\text{wp}(S1; S2; S3; S4, P_c) \equiv \{0 = 0 \land 0 = 0 \land 0 = 0 \land 0 = 0 \land P\} \equiv \{\text{true} \land P\} \equiv \{P\}$$

dado que reemplazamos todas las variables de P_c por 0.

Entonces queda demostrado que $P \longrightarrow \text{wp}(S1; S2; S3; S4, P_c)$, pues $P \longrightarrow P$, y queda demostrada la correctitud del programa completo respecto a su especificación luego de aplicar el Corolario de la Monotonía, habiendo sido demostrada la correctitud de las distintas partes del programa (antes, en y después del ciclo).

3.2. obtenerSenadoresEnProvincia

Nombremos los predicados y auxiliares que vamos a usar en la demostración de correctitud del código de este programa:

- $\texttt{sinRepetidos}(l) = \{(\forall i, j : \mathbb{Z})\ (0 \leq i < j < |l| \longrightarrow_L l[i] \neq l[j])\}$
- $\texttt{sinNegativos}(l) = \{(\forall i : \mathbb{Z})\ (0 \leq i < |\text{l}| \longrightarrow_L \text{l}[i] \geq 0)\}$
- $\texttt{esIdDelMaximo}(l, i) = \{0 \leq i < |l| \land_L (\forall j : \mathbb{Z})\ (0 \leq j < |l| \longrightarrow_L l[j] \leq l[i])\}$
- $\texttt{esIdDelSegundoMaximo}(l, i) = \{0 \leq i < |l| \land \neg\texttt{esIdDelMaximo}(l, i) \land$
  $(\forall j : \mathbb{Z})\ ((0 \leq j < |l| \land \neg\texttt{esIdDelMaximo}(l, j)) \longrightarrow_L l[j] \leq l[i])\}$
- La precondición de la especificación: $P = \{|\text{escrutinio}| \geq 3 \land \texttt{sinRepetidos}(\text{escrutinio}) \land \texttt{sinNegativos}(\text{escrutinio})\}$
- La postcondición de la especificación: $Q = \{\texttt{esIdDelMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), res_0) \land$
  $\texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), res_1)\}$
- La condición del primer if: $A = \{\text{escrutinio}[0] > \text{escrutinio}[1]\}$.
- La condición del ciclo: $B = \{i < |\text{escrutinio}| - 1\}$.
- La condición del primer if del ciclo: $C = \{\text{escrutinio}[i] > \text{escrutinio}[\text{indice\_maximo}]\}$.
- La condición del segundo if del ciclo: $D = \{\text{escrutinio}[i] > \text{escrutinio}[\text{indice\_segundo}]\}$.

Para demostrar la correctitud del ciclo, proponemos la siguiente pre y postcondición, invariante y función variante:

- $P_c = \{P \land_L i = 2 \land (A \land \text{indice\_maximo} = 0 \land \text{indice\_segundo} = 1) \lor (\neg A \land \text{indice\_maximo} = 1 \land \text{indice\_segundo} = 0)\}$.

  Por ser razonable que P_c sea la conjunción entre P y el estado resultante de las instrucciones previas al ciclo.

- $I = \{2 \leq i < |\text{escrutinio}| \land 0 \leq \text{indice\_maximo}, \text{indice\_segundo} < |\text{escrutinio}| - 1 \land_L$
  $\texttt{esIdDelMaximo}(\texttt{subseq}(\text{escrutinio}, 0, i), \text{indice\_maximo}) \land \texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{escrutinio}, 0, i), \text{indice\_segundo})\}$.

  Porque refleja lo que desde la intuición comprendemos que debe ocurrir al principio, durante (luego de cada iteración) y al final de la ejecución del ciclo.

- $Q_c = \{\texttt{esIdDelMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), \text{indice\_maximo}) \land$
  $\texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), \text{indice\_segundo})\}$
  Porque esto implica que al finalizar el ciclo las variables indice_maximo e indice_segundo van a contener a lo que queremos luego guardar y devolver en la variable res.

- $f_v = |\text{escrutinio}| - 1 - i$
  Porque f_v decrece en cada iteración del ciclo y se hace 0 luego de la última, cuando $i = |\text{escrutinio}| - 1$.

Comenzamos calculando la wp de la última instrucción del código:

$$\text{wp}(\text{res} := (\text{indice\_maximo}, \text{indice\_segundo}), Q) \equiv Q^{\text{res}_0, \text{res}_1}_{\text{indice\_maximo}, \text{indice\_segundo}} \equiv$$

$$\{\texttt{esIdDelMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), \text{indice\_maximo}) \land$$

$$\texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{escrutinio}, 0, |\text{escrutinio}| - 1), \text{indice\_segundo})\} \equiv Q_c$$

Para ver que la instrucción que le sigue al ciclo es correcta queremos ver que

$$Q_c \longrightarrow \text{wp}(\text{res} := (\text{indice\_maximo}, \text{indice\_segundo}), Q)$$

pero como vimos esa wp es equivalente a Q_c, y $Q_c \longrightarrow Q_c$ es tautológico. Entonces queda probada la correctitud de la porción del programa posterior al ciclo.

Para probar la correctitud parcial del ciclo, por el Teorema del Invariante basta con demostrar:

1. $P_c \longrightarrow I$
2. $\{I \wedge B\} S_c \{I\}$
3. $I \wedge \neg B \longrightarrow Q_c$

Procedamos:

1. Dado que P_c implica que
   - $|\text{escrutinio}| \geq 3 \wedge i = 2 \longrightarrow 2 \leq i < |\text{escrutinio}|$.
   - $|\text{escrutinio}| \geq 3 \wedge \texttt{sinRepetidos}(\text{escrutinio})$ implica la excistencia de un primer y segundo máximo en la subsecuencia $\texttt{subseq}(\text{escrutinio}, 0, 2)$, según los predicados.
   - $A \wedge \text{indice\_maximo} = 0 \wedge \text{indice\_segundo} = 1$ implica que

     $$\texttt{esIdDelMaximo}(\texttt{subseq}(\text{escrutinio}, 0, 2), 0) \wedge \texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{escrutinio}, 0, 2), 1)$$

     y que los índices de los máximos están entre 0 y $|\text{escrutinio}|$.
   - De forma análoga, si vale $\neg A \wedge \text{indice\_maximo} = 1 \wedge \text{indice\_segundo} = 0$, junto con la hipótesis de $\texttt{sinRepetidos}$, vale que

     $$\texttt{esIdDelMaximo}(\texttt{subseq}(\text{escrutinio}, 0, 2), 1) \wedge \texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{escrutinio}, 0, 2), 0)$$

     y que los índices son válidos.

   Entonces es cierto que $P_c \longrightarrow I$, como se quería probar.

2. Queremos ver que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$, para lo cual calcularemos esta última wp:

   $$\text{wp}(i := i + 1, I) \equiv I^i_{i+1}$$

   $$\text{wp}(S_c, I) \equiv \text{wp}(\textsf{if } C \textsf{ then } S_{11-12} \textsf{ else } S_{14-18} \textsf{ fi}, I^i_{i+1}) \equiv$$
   $$\{0 \leq i, \text{i\_max} < |\text{escrutinio}| \wedge_L (C \wedge \text{wp}(S_{11-12}, I^i_{i+1})) \vee (\neg C \wedge \text{wp}(S_{14-18}, I^i_{i+1})))\}$$

   donde S_{i-j} representa a la secuencia de instrucciones del programa de la línea i a la línea j inclusive, e $\text{i\_max} = \text{indice\_maximo}$ y $\text{i\_seg} = \text{indice\_segundo}$.

   Podemos entonces calcular la wp de cada parte por separado:

   $$\text{wp}(S_{11-12}, I^i_{i+1}) \equiv \text{wp}(\text{i\_seg} := \text{i\_max}, \text{wp}(\text{i\_max} := i, I^i_{i+1})) \equiv I^{i,\text{i\_seg},\text{i\_max}}_{i+1,\text{i\_max},i}$$

   $$\text{wp}(S_{14-18}, I^i_{i+1}) \equiv \text{wp}(\textsf{if } D \textsf{ then } \text{i\_seg} := i \textsf{ else } \mathit{skip} \textsf{ fi}) \equiv$$
   $$\{0 \leq i, \text{i\_seg} < |\text{escrutinio}| \wedge_L ((D \wedge \text{wp}(\text{i\_seg} := i, I^i_{i+1})) \vee (\neg D \wedge \text{wp}(\mathit{skip}, I^i_{i+1})))\} \equiv$$
   $$\{0 \leq i, \text{i\_seg} < |\text{escrutinio}| \wedge_L ((D \wedge I^{i,\text{i\_seg}}_{i+1,i}) \vee (\neg D \wedge I^i_{i+1}))\}$$

   Entonces,

   $$\text{wp}(S_c, I) \equiv 0 \leq i, i\_max < |esc| \wedge_L (C \wedge I^{i,\text{i\_seg},\text{i\_max}}_{i+1,\text{i\_max},i}) \vee (\neg C \wedge (0 \leq i, i\_seg < |esc| \wedge_L ((D \wedge I^{i,i\_seg}_{i+1,i}) \vee (\neg D \wedge I^i_{i+1})))))$$

   Desglosando esta última igualdad:
   - $I^i_{i+1} \equiv \{2 \leq i + 1 < |\text{esc}| \wedge 0 \leq i\_max, i\_seg < |\text{esc}| - 1 \wedge_L$
     $\texttt{esIdDelMaximo}(\texttt{subseq}(\text{esc}, 0, i + 1), i\_max) \wedge$
     $\texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{esc}, 0, i + 1), i\_seg)\}$
   - $I^{i,i\_seg}_{i+1,i} \equiv \{2 \leq i + 1 < |\text{esc}| \wedge 0 \leq i\_max, i < |\text{esc}| - 1 \wedge_L$
     $\texttt{esIdDelMaximo}(\texttt{subseq}(\text{esc}, 0, i + 1), i\_max) \wedge$
     $\texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{esc}, 0, i + 1), i)\}$
   - $I^{i,\text{i\_seg},\text{i\_max}}_{i+1,\text{i\_max},i} \equiv \{2 \leq i + 1 < |\text{esc}| \wedge 0 \leq i, i\_max < |\text{esc}| - 1 \wedge_L$
     $\texttt{esIdDelMaximo}(\texttt{subseq}(\text{esc}, 0, i + 1), i) \wedge$
     $\texttt{esIdDelSegundoMaximo}(\texttt{subseq}(\text{esc}, 0, i + 1), i\_max)\}$

   Notemos que parte del invariante y B, $(2 \leq i < |\text{esc}| \wedge 0 \leq i\_max, i\_seg < |\text{esc}| - 1 \wedge i < |\text{esc}| - 1)$, implica $(0 \leq i, i\_max, i\_seg < |esc| - 1) \wedge (2 \leq i + 1 < |esc|)$, que a su vez implica $(0 \leq i, i\_max, i\_seg < |esc|)$. Para ver que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$, basta con ver que esto se cumple al analizar los tres casos posibles dentro del ciclo (o más bien, sus paralelos dentro de la wp):

- Caso $C \equiv esc[i] > esc[i\_max]$. En este caso, para que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ basta con que $\{I \wedge B\} \longrightarrow \{0 \leq i, i\_max < |esc| \wedge_L C \wedge I^{i,\text{i\_seg},\text{i\_max}}_{i+1,\text{i\_max},i}\}$ (ya que es uno de los terminos separados por una disyuncion logica del resto de los terminos). Como ya comentamos, $\{I \wedge B\} \longrightarrow \{0 \leq i, i\_max < |esc|\}$, y estamos asumiendo que estamos en el caso en que vale C. Entonces queda ver que, asumiendo C, $\{I \wedge B\} \longrightarrow \{I^{i,\text{i\_seg},\text{i\_max}}_{i+1,\text{i\_max},i}\}$. Esto es verdadero ya que $I \wedge B \wedge C$ implica que i es el índice del máximo de la subsecuencia $subseq(esc, 0, i+1)$ y que por lo tanto i es el nuevo i_max y el i_max anterior es el nuevo i_seg, que es precisamente lo que está expresado en $I^{i,\text{i\_seg},\text{i\_max}}_{i+1,\text{i\_max},i}$. Además, como ya se comentó, $I \wedge B \longrightarrow 2 \leq i+1 < |\text{esc}| \wedge 0 \leq i, i\_max < |\text{esc}| - 1$. Entonces vale que $\{I \wedge B\} \longrightarrow \{I^{i,\text{i\_seg},\text{i\_max}}_{i+1,\text{i\_max},i}\}$ y por lo tanto también que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ si vale C.
- Caso $\neg C \wedge D \equiv esc[i] \leq esc[i\_max] \wedge esc[i] > esc[i\_seg]$. Para que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ basta con que $\{I \wedge B\} \longrightarrow \{0 \leq i, i\_max < |esc| \wedge_L \neg C \wedge (0 \leq i, i\_seg < |esc| \wedge_L D \wedge I^{i,i\_seg}_{i+1,i})\}$. Ya comentamos que $\{I \wedge B\} \longrightarrow \{0 \leq i, i\_max, i\_seg < |esc|\}$, y estamos asumiendo que estamos en el caso en que vale $\neg C \wedge D$. Queda ver que, asumiendo $\neg C \wedge D$, $\{I \wedge B\} \longrightarrow \{I^{i,i\_seg}_{i+1,i}\}$. Esto es verdadero pues $I \wedge \wedge B \neg C \wedge D$ implica que i es el índice del segundo máximo de la subsecuencia $subseq(esc, 0, i+1)$ y que por lo tanto i es el nuevo i_seg y el i_max anterior no es afectado, exáctamente lo que expresa $I^{i+1,i}_{i,i\_seg}$. Además, como ya se comentó, $I \wedge B \longrightarrow 2 \leq i+1 < |\text{esc}| \wedge 0 \leq i\_max, i < |\text{esc}| - 1$. Entonces vale que $\{I \wedge B\} \longrightarrow \{I^{i,i\_seg}_{i+1,i}\}$ y por lo tanto también que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ si vale $\neg C \wedge D$.
- Caso $\neg C \wedge \neg D \equiv esc[i] \leq esc[i\_max] \wedge esc[i] \leq esc[i\_seg]$. Para que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ basta con que $\{I \wedge B\} \longrightarrow \{0 \leq i, i\_max < |esc| \wedge_L \neg C \wedge (0 \leq i, i\_seg < |esc| \wedge_L \neg D \wedge I^{i}_{i+1})\}$. Ya comentamos que $\{I \wedge B\} \longrightarrow \{0 \leq i, i\_max, i\_seg < |esc|\}$, y estamos asumiendo que estamos en el caso en que vale $\neg C \wedge \neg D$. Queda ver que, asumiendo $\neg C \wedge \neg D$, $\{I \wedge B\} \longrightarrow \{I^{i}_{i+1}\}$. Esto es verdadero considerando que $I \wedge B \wedge \neg C \wedge \neg D$ implica que no se tiene ni al máximo ni al segundo máximo de $subseq(esc, 0, i+1)$ en la posición i y que por lo tanto i_max e i_seg no son modificados en la iteración, que es exáctamente lo que expresa I^{i}_{i+1}. Por último, sabemos que $I \wedge B \longrightarrow 2 \leq i+1 < |\text{esc}| \wedge 0 \leq i\_max, i\_seg < |\text{esc}| - 1$. Entonces vale que $\{I \wedge B\} \longrightarrow \{I^{i}_{i+1}\}$ y por lo tanto también que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ si vale $\neg C \wedge \neg D$.

Dado que siempre se cumple que $(C \vee (\neg C \wedge (D \vee \neg D))) \equiv ((C) \vee (\neg C \wedge D) \vee (\neg C \wedge \neg D))$ (es decir, ocurre uno de estos 3 casos analizados) y a que se probó que en todos los casos vale $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$, entonces es verdad que $\{I \wedge B\} \longrightarrow \text{wp}(S_c, I)$ en general, y por lo tanto tambíen que $\{I \wedge B\} S_c \{I\}$ como se quería demostrar.

3. $I \wedge \neg B \longrightarrow i = |\text{escrutinio}| - 1$, condición bajo la cual Q_c es idéntico a parte de I. Por lo tanto es inmediato que $I \wedge \neg B \longrightarrow Q_c$.

Para probar la terminación del ciclo, debemos demostrar:

1. $\{I \wedge B \wedge f_v = v_0\} S_c \{f_v < v_0\}$
2. $I \wedge f_v \leq 0 \longrightarrow \neg B$

Veamos que esto es cierto:

1. Queremos ver que $\{I \wedge B \wedge f_v = v_0\} \longrightarrow \text{wp}(S_c, f_v < v_0)$, para lo cual calcularemos esta última wp.
   $\text{wp}(i := i+1, |esc| - 1 - i < v_0) \equiv \{|esc| - 1 - i - 1 < v_0\} \equiv \{f_v < v_0 + 1\}$

   $\text{wp}(S_c, f_v < v_0) \equiv \text{wp}(\textsf{if } C \textsf{ then } S_{11-12} \textsf{ else } S_{14-18} \textsf{ fi}, f_v < v_0 + 1)$
   Como $f_v < v_0 + 1$ no depende de i_max ni de i_seg, esto resulta en:
   $\text{wp}(S_c, f_v < v0) \equiv 0 \leq i, i\_max < |esc| \wedge_L ((C \wedge f_v < v_0 + 1) \vee (\neg C \wedge (0 \leq i, i\_seg < |esc| \wedge_L ((D \wedge f_v < v_0 + 1) \vee (\neg D \wedge f_v < v_0 + 1)))))$
   Luego, dado que

   - $f_v = v_0 \longrightarrow f_v < v_0 + 1$
   - $I \wedge B \longrightarrow 0 \leq i, i\_max, i\_seg < |esc|$
   - Siempre se cumple $(C \vee (\neg C \wedge (D \vee \neg D)))$

   Se vé que $\{I \wedge B \wedge f_v = v_0\} \longrightarrow \text{wp}(S_c, f_v < v_0)$ y que por lo tanto queda probado que $\{I \wedge B \wedge f_v = v_0\} S_c \{f_v < v_0\}$.
2. Considerando que $f_v \leq 0 \leftrightarrow i \geq |esc| - 1$ se vé que $I \wedge f_v \leq 0 \longrightarrow i = |esc| - 1$, condición bajo la cual vale $\neg B$. Por lo tanto, $I \wedge f_v \leq 0 \longrightarrow \neg B$.

Demostrada la correctitud parcial y la terminación del ciclo, queda entonces demostrada su correctitud total.
Para ver que el programa entero es correcto respecto de su especificación, queda ver que $P \longrightarrow wp(S_{1-8}, P_c)$
$wp(i := 2, P_c) \equiv \{P \wedge_L 2 = 2 \wedge ((esc[0] > esc[1] \wedge i\_max = 0 \wedge i\_seg = 1) \vee (esc[1] > esc[0] \wedge i\_max = 1 \wedge i\_seg = 0))\} \equiv \{P \wedge_L ((esc[0] > esc[1] \wedge i\_max = 0 \wedge i\_seg = 1) \vee (esc[1] > esc[0] \wedge i\_max = 1 \wedge i\_seg = 0))\} \equiv Q_8$

$wp(S_{1-8}, P_c) \equiv wp(\textsf{if } esc[0] > esc[1] \textsf{ then } i\_max := 0; i\_seg := 1 \textsf{ else } i\_max := 1; i\_seg := 0 \textsf{ fi}, Q_8) \equiv$

$1 < |esc| \wedge_L ((esc[0] > esc[1] \wedge wp(i\_max := 0, wp(i\_seg := 1, Q_8))) \vee (esc[0] \leq esc[1] \wedge wp(i\_max := 1, wp(i\_seg := 0, Q_8))))$

$wp(i\_max := 0, wp(i\_seg := 1, Q_8)) \equiv \{P \wedge_L ((esc[0] > esc[1] \wedge 0 = 0 \wedge 1 = 1) \vee (esc[1] > esc[0] \wedge 0 = 1 \wedge 1 = 0))\} \equiv \{P \wedge_L esc[0] > esc[1]\}$

$wp(i\_max := 1, wp(i\_seg := 0, Q_8)) \equiv \{P \wedge_L ((esc[0] > esc[1] \wedge 1 = 0 \wedge 0 = 1) \vee (esc[1] > esc[0] \wedge 1 = 1 \wedge 0 = 0))\} \equiv \{P \wedge_L esc[0] \leq esc[1]\}$

$wp(S_{1-8}, P_c) \equiv \{1 < |esc| \wedge_L P\}$

Dado que

- $P \longrightarrow P$
- $P \longrightarrow 1 < |esc|$

Se vé que $P \longrightarrow wp(S_{1-8}, P_c)$, con lo que queda demostrada la correctitud del programa previo al ciclo. Demostrada la correctitud del programa antes, durante y después del ciclo, por monotonía queda demostrada la correctitud del programa completo.