DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo practico 1:

Especificacion y WP

18 de septiembre de 2023 — Algoritmos y Estructuras de Datos

Barbies

Integrante	LU	Correo electrónico
Gonzalez Dardik, Micaela	1143/22	micagonzdark@gmail.com
Navarro, Solana	906/22	solanan3@gmail.com
Hambra, Jacobo Brian	845/22	brianhambra10@gmail.com
Suarez, Ines	890/22	ine.suarez22@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. Especificacion

AUXILIARES:

`aux porcentaje` (in list : $seq\langle\mathbb{Z}\rangle$, n : $\mathbb{Z}$) : $\mathbb{Z}$ $= (n * 100)/sumaElementos(list)$;

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

PREDICADOS AUXILIARES:

`pred escrutinioValido` (in escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$masDeUnPartido(escrutinio) \wedge sinRepetidos(escrutinio) \wedge todosPositivos(escrutinio) \wedge pasaAlMenosUno(escrutinio)$
}
`pred masDeUnPartido` (in escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$|escrutinio| > 2$
}
`pred sinRepetidos` (in l: $seq\langle\mathbb{Z}\rangle$) {
$(\forall i, j : \mathbb{Z})((0 \leq i < |l| \wedge 0 \leq j < |l| \wedge i \neq j) \Rightarrow l[i] \neq l[j])$
}
`pred todosPositivos` (in escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$(\forall x \in escrutinio)(x \geq 0)$
}
`pred esMayor` (in x: $\mathbb{Z}$, y: $\mathbb{Z}$) {
$x \geq y$
}
`pred esMaximo` (in elem: $\mathbb{Z}$, list: $seq\langle\mathbb{Z}\rangle$) {
$(\forall i : \mathbb{Z})(0 \leq i < |list|) \implies (esMayor(elem, list[i]))$
}
`pred dHondtValida` (in matriz: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$esMatriz(matriz) \wedge todosPositivosMatriz(matriz) \wedge masDeUnPartidoMatriz(matriz) \wedge distintaCantDeVotos(matriz)$
}
`pred esMatriz` (in matriz: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})(0 \leq i < |matriz|) \implies (|matriz[i]| > 0) \wedge (\forall j : \mathbb{Z})(0 \leq j < |matriz|) \implies (|matriz[i]| = |matriz[j]|$
}
`pred todosPositivosMatriz` (in matriz: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(\forall x \in matriz)(x \geq 0)$
}
`pred masDeUnPartidoMatriz` (in matriz: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$|matriz| > 1$
}
`pred distintaCantDeVotos` (in matriz: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) {
$(\forall i : \mathbb{Z})(0 \leq i < |matriz| - 1) \implies (matriz[i][0] \neq matriz[i+1][0])$
}
`pred noEsUltimo` (in elem: $\mathbb{Z}$, list: $seq\langle\mathbb{Z}\rangle$) {
$elem \neq list[-1]$
}
`pred sumasIguales` (in s1: $seq\langle\mathbb{Z}\rangle$, s2: $seq\langle\mathbb{Z}\rangle$) {
$sumaElementos(s1) = sumaElementos(s2)$
}
`pred pasaElUmbral` (in list: $seq\langle\mathbb{Z}\rangle$, n: $\mathbb{Z}$) {
$porcentaje(list, n) \geq 3$
}
`pred pasaAlMenosUno` (in list: $seq\langle\mathbb{Z}\rangle$) {
$(\exists x \in list \wedge pasaElUmbral(list, x))$
}
`pred esDeLosNmayores` (in cantbancas: $\mathbb{Z}$, mat: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, valor: $\mathbb{Z}$, escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$(\exists A = \{n_1, .., n_r\})(0 \leq r < cantbancas)(valor \in A)(\#A = cantbancas) \wedge ((\forall n \in mat \wedge n \notin A))$
$\implies (\forall p \in A) \implies (n < p) \wedge (\forall i : \mathbb{Z})(0 \leq i < |dhondt|)(\forall j : \mathbb{Z})(0 \leq j < |dhondt[i]|)(dhondt[i][j] \in A)$
$\implies (pasaElUmbral(escrutinio, dhondt[i][0]))$
}

1.1. hayBallotage: verifica si hay ballotage en la eleccion presidencial.

proc `hayBallotage` (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : Bool
requiere $\{escrutinioValido(escrutinio)\}$
asegura $\{res = false \Leftrightarrow (((\exists x \in escrutinio)(noEsUltimo(x, escrutinio)) \land (esMaximo(x, escrutinio) \land porcentaje(escrutinio, x) \geq 45)) \lor ((\exists y \in escrutinio) (noEsUltimo(y, escrutinio)) \land (x \neq y) \land (\forall z \in escrutinio)(noEsUltimo(z, escrutinio) \land (x \neq z) \implies esMayor(y, z) \land porcentaje(escrutinio, x) \geq 40 \land porcentaje(escrutinio, x) - porcentaje(escrutinio, y) > 10))\}$

1.2. hayFraude: verifica que los votos validos de los tres tipos de cargos electivos sumen lo mismo.

proc `hayFraude` (in escrutiniopresidencial : $seq\langle\mathbb{Z}\rangle$, escrutiniosenadores : $seq\langle\mathbb{Z}\rangle$, escrutiniodiputados : $seq\langle\mathbb{Z}\rangle$) : Bool
requiere $\{escrutinioValido(escrutiniopresidencial) \land escrutinioValido(escrutiniosenadores) \land escrutinioValido(escrutiniodiputados)\}$
requiere $\{|escrutiniopresidencial| = |escrutiniosenadores| \land |escrutiniosenadores| = |escrutiniodiputados|\}$
asegura $\{res = False \Leftrightarrow sumasIguales(escrutiniopresidencial, escrutiniosenadores) \land sumasIguales(escrutiniosenadores, escrutniodiputados)\}$

1.3. obtenerSenadoresEnProvincia: obtiene los id de los partidos (primero y segundo) para la eleccion de senadores en una provincia. El id es el ındice de las listas escrutinios.

proc `obtenerSenadoresEnProvincia` (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z} \times \mathbb{Z}$
requiere $\{escrutinioValido(escrutinio)\}$
asegura $\{(\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \land (esMaximo(escrutinio[i], escrutinio) \land_L (res_0 = i)\}$
asegura $\{((\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \land esMaximo(escrutinio[i], escrutinio)) \land (\exists j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \land i \neq j \land (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \land k \neq i) \implies (esMayor(escrutinio[j], escrutinio[k])) \land_L (res_1 = j)\}$

1.4. calcularDHondtEnProvincia: calcula los cocientes segun el metodo d'Hondt para diputados en una provincia

proc `calcularDHondtEnProvincia` (in cantbancas : $\mathbb{Z}$, escrutinio : $seq\langle\mathbb{Z}\rangle$) : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$
requiere $\{escrutinioValido(escrutinio)\}$
asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |escrutinio| - 1) \implies (cantbancas = |res[i]|)\}$
asegura $\{|escrutinio| - 1 = |res|\}$
asegura $\{(\forall i : \mathbb{Z})(0 \leq i < cantbancas) \implies (\forall j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \land pasaElUmbral(escrutinio[j])) \implies (res[j][i] = escrutinio[j]/i + 1)\}$

1.5. obtenerDiputadosEnProvincia: calcula la cantidad de bancas de diputados obtenidas por cada partido en una provincia.

proc `obtenerDiputadosEnProvincia` (in cantbancas : $\mathbb{Z}$, dHondt : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$, escrutinio : $seq\langle\mathbb{Z}\rangle$) : $seq\langle\mathbb{Z}\rangle$
requiere $\{dHondtValida(dHondt)\}$
requiere $\{escrutinioValido(escrutinio)\}$
asegura $\{|res| = |dHondt|\}$
asegura $\{(\sum_{i=0}^{|list|-1} res[i] = cantbancas\}$
asegura $\{(\forall i : \mathbb{Z})(0 \leq i < |dHondt|) \implies res[i] = (\sum_{j=0}^{|dHondt[i]|-1}(if(esDeLosNmayores(cantbancas, dHondt, dHondt[i][j], escrutinio)\}$

1.6. validarListasDiputadosEnProvincia: verifica que la listas de diputados de cada partido en una provincia contenga exactamente la misma cantidad de candidatos que bancas en disputa en esa provincia, y que ademas se cumpla la alternancia de generos.

proc `validarListasDiputadosEnProvincia` (in cantbancas : $\mathbb{Z}$, listas : $seq\langle seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle\rangle$) : Bool
requiere $\{|listas| > 1\}$
asegura $\{res = false \Leftrightarrow ((\forall i : \mathbb{Z})(0 \leq i < |listas|) \implies (|listas[i]| \neq cantbancas) \land (\forall j : \mathbb{Z})(0 \leq j < cantbancas - 1) \implies (listas[i][j]_1 = listas[i][j + 1]_1))\}$

2. Implementacion

2.1. hayBallotage: verifica si hay ballotage en la eleccion presidencial.

```
result := True;

maximo := 0;
segundoMaximo := 0;
i := 0;
while (i < escrutinio.size()-1) do
   maximoGuardado :=maximo;
      if (escrutinio[i] >= maximo) then
         maximo := escrutinio[i];
      else
        if (escrutinio[i] >= segundoMaximo) then
         segundoMaximo := escrutinio[i];
        else
         skip;
        endif
      endif
      if (maximoGuardado <= segundoMaximo) then
        segundoMaximo := maximoGuardado;
      else
         skip;
      endif
      i := i+1;
endwhile

j := 0;
sumaElementos := 0;
while (j < escrutinio.size()) do
  sumaElementos := sumaElementos + escrutinio[j];
  j := j + 1;
endwhile

porcentajePrimerMaximo := (maximo*100)/sumaElementos;
porcentajeSegundoMaximo := (segundoMaximo*100)/sumaElementos;

if(porcentajePrimerMaximo > 45) then
  result := False;
else
  skip;
endif

if(porcentajePrimerMaximo > 40 && (porcentajePrimerMaximo - porcentajeSegundoMaximo) > 10) then
  result := False;
else
  skip;
endif
```

2.2. hayFraude: verifica que los votos validos de los tres tipos de cargos electivos sumen lo mismo.

```
sumapresi := 0;
i := 0;

while (i < escrutiniopresidencial.size()) do
  sumapresi := sumapresi + escrutiniopresidencial[i];
  i := i + 1;
endwhile;

```

```
sumasena := 0;
j := 0;

while (j < escrutiniosenadores.size()) do
  sumasena := sumasena + escrutiniosenadores[j];
  j := j + 1;
endwhile;

sumadipu := 0;
k := 0;

while (k < escrutiniodiputados.size()) do
  sumadipu := sumadipu + escrutiniodiputados[k];
  k := k + 1;
endwhile;

if (sumapresi == sumasena) && (sumasena == sumadipu) then
      result := False;
else
      result := True;
endif
```

2.3. obtenerSenadoresEnProvincia: obtiene los id de los partidos (primero y segundo) para la eleccion de senadores en una provincia. El id es el ındice de las listas escrutinios.

```
max := 0;
seg :=0;
i:=0;
while (i< escrutinio.size()-1) do
   if (escrutinio[i]>escrutinio[max]) then
       max=i;
   else
       skip;
   endif
   i:= i+1;
endwhile
if (max==0) then
    seg:=1;
 else
    seg:=0;
 endif
 j:=0
 while (j< escrutinio.size()-1) do
   if (j!=max && escrutinio[j]>escrutinio[seg]) then
       seg:=j;
   else
       skip;
   endif
   j:= j+1;
 endwhile
 res[0] := max;
 res[1] := seg;
```

2.4. validarListasDiputadosEnProvincia: verifica que la listas de diputados de cada partido en una provincia contenga exactamente la misma cantidad de candidatos que bancas en disputa en esa provincia, y que ademas se cumpla la alternancia de generos.

```
result := True;
i := 0;
```

```
while (i < listas.size()) do
   if (listas[i].size() == cantbancas) then
      j := 0;
      while (j < listas[i].size()-1) do
         if(listas[i][j][1] == listas[i][j+1][1]) then
            result := False;
            j := j + 1;
          else
            j := j + 1;
          endif
      i := i + 1;
      endwhile
   else
       result := False;
       i := i + 1;
   endif
endwhile
```

3. Demostraciones de correctitud mediante el metodo de weakest precondition (WP)

Para demostrar la correctitud de un programa completo debemos demostrar estas 3 implicaciones:

1. $Pre \implies wp(Codigoprevioalciclo, P_C)$
2. $P_C \implies wp(Ciclo, Q_C)$
3. $Q_C \implies wp(Codigoposterioralciclo, Post)$

Esta $P_C \implies wp(Ciclo, Q_C)$ la hacemos con el teorema del invariante (porque no se puede calcular esa wp en general):

1. $P_C \implies I$
2. $I \wedge BSI$
3. $I \wedge \neg B \implies Q_C$
4. $I \wedge B \wedge v_0 = fvSfv < v_0$
5. $I \wedge fv \leq 0 \implies \neg B$

3.1. hayFraude: verifica que los votos validos de los tres tipos de cargos electivos sumen lo mismo.

Aclaracion: Como los tres ciclos son iguales y no tienen relacion el uno con el otro solo hicimos la demostracion con uno solo ya que es analogo para los otros 2

Datos:
$Pre \equiv (escrutinioValido(escrutiniopresidencial) \wedge escrutinioValido(escrutiniosenadores) \wedge$
$escrutinioValido(escrutiniodiputados) \wedge |escrutiniopresidencial| = |escrutiniosenadores|$
$\wedge |escrutiniosenadores| = |escrutiniodiputados|)$

Cuando hablemos de estos, los vamos a llamar como: escrutiniosvalidos y todosMismaLongitud

- $escrutinioValido(escrutiniopresidencial) \wedge escrutinioValido(escrutiniosenadores) \wedge$ $escrutinioValido(escrutiniodiputados) \equiv escrutiniosValidos$
- $|escrutiniopresidencial| = |escrutiniosenadores| \wedge$ $|escrutiniosenadores| = |escrutiniodiputados| \equiv todosMismaLongitud$

$Codigoprevioalciclo \equiv (result := True); (sumapresi := 0); (sumasena := 0); (sumadipu := 0); (i := 0)$

$P_C \equiv (escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (result := True) \wedge (sumapresi := 0)$
$\wedge (sumasena := 0) \wedge (sumadipu := 0) \wedge (i := 0)$

$Post \equiv result := False \Leftrightarrow (sumapresi == sumasena)\&\&(sumasena == sumadipu)$

Codigo posterior al ciclo $\equiv if(sumapresi == sumasena)\&\&(sumasena == sumadipu)then(result := False)$
else (result:= True) endif
$Q_C \equiv (i = |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{|escrutiniopresidencial|-1} escrutiniopresidencial[j])$

$$Q_{C2} \equiv (i = |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{|escrutiniopresidencial|-1} escrutiniopresidencial[j]) \wedge$$

(i = $|escrutiniosenadores|) \wedge_L (sumasena = \sum_{j=0}^{|escrutiniosenadores|-1} escrutiniosenadores[j]) \wedge$

(i = $|escrutiniodiputados|) \wedge_L (sumadipu = \sum_{j=0}^{|escrutiniodiputados|-1} escrutiniodiputados[j])$

Hacemos este Q_{C2} que representa la conjuncion de los tres Q_C que resultarian de los ciclos para la ultima implicacion
$I \equiv 0 \leq i \leq |escrutiniopresidencial| \wedge_L sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j]$
$B \equiv i < |escrutiniopresidencial|$
$fv = |escrutiniopresidencial| - i$

Resolvemos:

1. $Pre \implies wp(Codigoprevioalciclo, P_C)$

   Primero calculamos la wp, utilizamos el axioma 3 varias veces, ya que hay varias lineas de codigo:

   $wp((result := True); (sumapresi := 0); (sumasena := 0); (sumadipu := 0); (i := 0), P_C)$

   $\equiv wp(res := True; sumapresi := 0; sumasena := 0; sumadipu := 0; wp(i := 0, P_C))$

   $\equiv wp(res := True; sumapresi := 0; sumasena := 0, wp(sumadipu := 0, wp(i := 0, P_C)))$

   $\equiv wp(res := True, wp(sumapresi := 0, wp(sumasena := 0, wp(sumadipu := 0, wp(i := 0, P_C)))))$

   Utilizamos el Axioma 1, y en el segundo termino reemplazamos directamente lo que correspondia:

   Se repite casi la misma resolucion en todos los casos:

   - $E1 \equiv wp(i := 0, P_C) \equiv def(0) \wedge_L (escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (result := True) \wedge (sumapresi := 0) \wedge (sumasena := 0) \wedge (sumadipu := 0) \wedge (0 = 0))$
   - $E2 \equiv wp(sumadipu := 0, E1) \equiv def(0) \wedge_L (escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (result := True) \wedge (sumapresi := 0) \wedge (sumasena := 0) \wedge (0 := 0)$
   - $E3 \equiv wp(sumasena := 0, E2) \equiv def(0) \wedge_L (escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (result := True) \wedge (sumapresi := 0) \wedge (0 := 0)$
   - $E4 \equiv wp(sumapresi := 0, E3) \equiv def(0) \wedge_L (escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (result := True) \wedge (0 := 0)$
   - $E5 \equiv wp(result := True, E4) \equiv def(True) \wedge_L (escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (True := True)$

   En conclusion, $Pre \implies wp(codigoprevioalciclo, P_C) \Leftrightarrow (escrutiniosValidos) \wedge (todosMismaLongitud) \implies (escrutiniosValidos) \wedge (todosMismaLongitud)$

   Son lo mismo, se cumple.

2. $P_c \implies wp(Ciclo, Q_c)$

   a) $P_C \implies I$

      $P_C \implies I \equiv (escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (res = True) \wedge (sumapresi = 0) \wedge (sumasena = 0) \wedge (sumadipu = 0) \wedge (i = 0) \implies (0 \leq i \leq |escrutiniopresidencial| \wedge_L sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j])$

      Para probar esta implicacion hay que ver el caso cuando P_C es True ya que si es False la implicacion seria True de cualquier forma

      Entonces asumimos $P_C = True$ y reemplazamos en los terminos correspondientes de la invariante

- $(escrutiniosValidos) \wedge (todosMismaLongitud) \wedge (res = True) \wedge (sumapresi = 0) \wedge (sumasena = 0) \wedge (sumadipu = 0) \wedge (i = 0) \implies (0 \leq 0 \leq |escrutiniopresidencial| \wedge_L 0 = \sum_{j=0}^{0-1} escrutiniopresidencial[j])$
  $\implies (0 \leq 0 \leq |escrutiniopresidencial|) \wedge_L (0 = 0)$

La implicacion es True

b) $\{I \wedge B\}S\{I\}$

Para probar que la tripla es valida hay que demostrar que $\{I \wedge B\} \implies wp(S, I)$

$wp(S, I) \equiv wp(sumapresi := sumapresi + escrutiniopresidencial[i]; i := i+1; 0 \leq i \leq |escrutiniopresidencial| \wedge_L sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j]) \equiv wp(sumapresi := sumapresi + escrutiniopresidencial[i]; wp(i := i+1; 0 \leq i \leq |escrutiniopresidencial| \wedge_L sumapresi = \sum_{j=0}^{i-1}))$

- $E1 \equiv wp(i := i+1, I) \equiv def(i+1) \wedge_L (0 \leq i+1 \leq |escrutiniopresidencial|) \wedge$
  $(sumapresi = \sum_{j=0}^{i} escrutiniopresidencial[j]) \equiv$
  $(0 \leq i+1 < |escrutiniopresidencial|) \wedge_L (0 \leq i+1 \leq |escrutiniopresidencial|) \wedge$
  $(sumapresi = \sum_{j=0}^{i} escrutiniopresidencial[j]) \equiv$
  $(0 \leq i < |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{i} escrutiniopresidencial[j])$
- $E2 \equiv wp(sumapresi := sumapresi + escrutiniopresidencial[i], E1) \equiv$
  $def(sumapresi + escrutiniopresidencial[i]) \wedge_L (0 \leq i < |escrutiniopresidencial| \wedge$
  $sumapresi + escrutiniopresidencial[i] = \sum_{j=0}^{i} escrutiniopresidencial[j]) \equiv$
  $0 \leq i < |escrutiniopresidencial| \wedge_L (0 \leq i < |escrutiniopresidencial|$
  $\wedge\, sumapresi = \sum_{j=0}^{i} escrutiniopresidencial[j] - escrutiniopresidencial[i] \equiv$
  $0 \leq i < |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j])$
- $(0 \leq i \leq |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j])$
  $\wedge (i < |escrutiniopresidencial|) \implies (0 \leq i < |escrutiniopresidencial|)$
  $\wedge (sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j])$
  $(0 \leq i < |escrutiniopresidencial|) \implies (0 \leq i < |escrutiniopresidencial|)$
  $(sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j]) \implies (sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j])$

La implicacion es True

c) $I \wedge \neg B \implies Q_C$

$I \wedge \neg B \equiv (0 \leq i \leq |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j]) \wedge (\neg(i < |escrutiniopresidencial|))$

$I \wedge \neg B \implies Q_C \equiv (0 \leq i \leq |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j]) \wedge$
$(i >= |escrutiniopresidencial|) \implies (i = |escrutiniopresidencial|) \wedge_L$
$(sumapresi = \sum_{j=0}^{|escrutiniopresidencial|-1} escrutiniopresidencial[j])$

Lo separamos por partes:

$(0 \leq i \leq |escrutiniopresidencial|) \wedge (i >= |escrutiniopresidencial|) \implies (i = |escrutiniopresidencial|)$

Como $(i = |escrutiniopresidencial|)$ **:**

$(sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j]) \implies (sumapresi = \sum_{j=0}^{|escrutiniopresidencial|-1} escrutiniopresidencial[j])$

La implicacion es True

d) $\{I \wedge B \wedge v_0 = fv\}S\{fv < v_0\}$

Para probar que la tripla es valida hay que demostrar que $\{I \wedge B \wedge v_0 = fv\} \implies wp(S, fv < v_0)$

$\{I \wedge B \wedge v_0 = fv\} \equiv ((0 \leq i \leq |escrutiniopresidencial|) \wedge_L sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j] \wedge (i < |escrutiniopresidencial|) \wedge v_0 = |escrutiniopresidencial| - i$

Buscamos wp(S, fv $<$ **v**$_0$**)** :

$wp(S, fv < v_0) \equiv wp((sumapresi := sumapresi + escrutiniopresidencial[i]); wp((i := i+1), fv < v_0)$

$E1 \equiv wp(i := i+1, |escrutiniopresidencial| - i < v_0) \equiv def(i+1) \wedge_L |escrutiniopresidencial| - (i+1) < v_0$
$E2 \equiv wp(sumapresi := sumapresi + escrutiniopresidencial[i], E1) \equiv def(sumapresi + escrutiniopresidencial[i]) \wedge_L |escrutiniopresidencial| - (i+1) < v_0$

$wp(S, fv < v_0) \equiv (0 \leq i < |escrutinio|) \wedge_L |escrutiniopresidencial| - (i+1) < v_0$

Juntamos todo:

$((0 \leq i \leq |escrutiniopresidencial|) \wedge_L sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j] \wedge (i < |escrutiniopresidencial|) \wedge$ $v_0 = |escrutiniopresidencial| - i \implies \mathbf{(0} \leq i < |escrutinio|) \wedge_L |escrutiniopresidencial| - (i+1) < v_0$

Separo por partes:
$(0 \leq i \leq |escrutiniopresidencial|) \wedge (i < |escrutiniopresidencial|) \implies (0 \leq i < |escrutinio|)$
$v_0 = |escrutiniopresidencial| - i \implies |escrutiniopresidencial| - (i+1) < v_0$
La implicacion es True

e) $I \wedge fv \leq 0 \implies \neg B$
$I \wedge fv \leq 0 \implies \neg B \equiv \mathbf{(0} \leq i \leq |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{i-1} escrutiniopresidencial[j]) \wedge$ $|escrutiniopresidencial| - i \leq 0 \implies (\neg(i < |escrutiniopresidencial|))$

Como $(\neg(i < |escrutiniopresidencial|)) \equiv (i \geq |escrutiniopresidencial|)$ **y**

$(0 \leq i \leq |escrutiniopresidencial|) \wedge |escrutiniopresidencial| - i \leq 0 \implies (i \geq |escrutiniopresidencial|)$
La implicacion es True

---

3. $Q_C \implies wp(Codigoposterioralciclo, Post)$

Para esto usamos el Q_{C2} que teniamos arriba ya que necesitamos la info que resulto de los 3 ciclos

Primero calculamos la wp, utilizamos el axioma 3:

$wp(Codigoposterioralciclo, Post) \equiv wp(if(sumapresi == sumasena)\&\&(sumasena == sumadipu)then(result := False)else(result := True)endif, result := False \Leftrightarrow (sumapresi = sumasena)\&\&(sumasena = sumadipu))$

Por el axioma 4 sabemos que la wp es

$def((sumapresi = sumasena)\&\&(sumasena = sumadipu)) \wedge_L ((sumapresi = sumasena)\&\&(sumasena = sumadipu) \wedge wp(result := False, result := False \Leftrightarrow (sumapresi = sumasena)\&\&(sumasena = sumadipu))) \vee (\neg((sumapresi = sumasena)\&\&(sumasena = sumadipu)) \wedge wp(result := True, result := False \Leftrightarrow (sumapresi = sumasena)\&\&(sumasena = sumadipu)))$

- $def((sumapresi = sumasena)\&\&(sumasena = sumadipu)) = True$
- $wp(result := False, result := False \Leftrightarrow (sumapresi = sumasena)\&\&(sumasena = sumadipu))$
  - $False := False \Leftrightarrow (sumapresi = sumasena)\&\&(sumasena = sumadipu)$
- $wp(result := True, result := False \Leftrightarrow (sumapresi = sumasena)\&\&(sumasena = sumadipu))$
  - $True := False \Leftrightarrow (sumapresi = sumasena)\&\&(sumasena = sumadipu)$

Juntamos todo:

$wp(Codigoposterioralciclo, Post) \equiv True \wedge_L ((sumapresi = sumasena)\&\&(sumasena = sumadipu) \wedge True) \vee ((\neg((sumapresi = sumasena)\&\&(sumasena = sumadipu)) \wedge False))$

Demostramos la implicacion:

$((i = |escrutiniopresidencial|) \wedge_L (sumapresi = \sum_{j=0}^{|escrutiniopresidencial|-1} escrutiniopresidencial[j])) \wedge$

$((i = |escrutiniosenadores|) \wedge_L (sumasena = \sum_{j=0}^{|escrutiniosenadores|-1} escrutiniosenadores[j])) \wedge ((i = |escrutiniodiputados|) \wedge_L (sumadipu = \sum_{j=0}^{|escrutiniodiputados|-1} escrutiniodiputados[j])) \implies ((sumapresi = sumasena)\&\&(sumasena = sumadipu)) \vee (False)$

La implicacion es True

3.2. obtenerSenadoresEnProvincia: obtiene los id de los partidos (primero y segundo) para la eleccion de senadores en una provincia. El id es el ındice de las listas escrutinios.

Predicados Auxiliares para la demostracion de la correctitud:

`pred esMaximo` (in max: $\mathbb{Z}$, lista: $seq\langle\mathbb{Z}\rangle$) {
$(\forall i : \mathbb{Z})(0 \leq i < |escrutinio| - 1) \implies escrutinio[i] < escrutinio[max])$
}
`pred esSegMaximo` (in seg: $\mathbb{Z}$, in max: $\mathbb{Z}$, lista: $seq\langle\mathbb{Z}\rangle$) {

$(\forall i:\mathbb{Z})(0 \leq i < |escrutinio| - 1) \wedge (i \neq max) \implies_L (escrutinio[i] < escrutinio[seg])$
}

Datos:
$Pre \equiv escrutinioValido(escrutinio)$
$P_{C1} \equiv (i := 0) \wedge (max := 0) \wedge (escrutinioValido(escrutinio))$
$Q_{C1} \equiv (i := |escrutinio| - 1) \wedge (esMaximo(max, lista))$
$I_1 \equiv (0 \leq i \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k:\mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies$
$(escrutinio[max] > escrutinio[k])$
$B_1 \equiv (i < |escrutinio| - 1)$
$P_{C2} \equiv (j = 0) \wedge (0 \leq seg \leq 1) \wedge (seg \neq max) \wedge (esMaximo(max, escrutinio)) \wedge_L (escrutinio[max] > escrutinio[seg])$
$Q_{C2} \equiv (j = |escrutinio| - 1) \wedge (max \neq seg) \wedge (esMaximo(max, escrutinio)) \wedge_L (esSegundoMaximo(max, seg, escrutinio))$
$I_2 \equiv (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge (esMaximo(max, escrutinio)) \wedge$
$(\forall k:\mathbb{Z})(0 \leq k \leq j) \wedge (k \neq max) \wedge (k \neq seg) \implies (escrutinio[seg] > escrutinio[k])$
$B_2 \equiv (j < |escrutinio| - 1)$
$S1 \equiv max := 0; seg := 0; i := 0;$
$S2 \equiv While1$
$S3 \equiv if(max == 0) then seg := 1 else seg := 0; j := 0)$
$S4 \equiv While2$
$S5 \equiv res[0] := max; res[1] := seg;$

Para demostrar la correctitud de un programa completo vamos a demostrar todas las implicaciones de los bloques que componen al programa completo :

1. $Pre \implies wp(S1, P_C 1)$

   $Pre \implies wp(S1, P_C 1) \equiv escrutinioValido \implies wp(max := 0; seg := 0; i := 0, i := 0 \wedge max := 0 \wedge escrutinioValido)$

   Utilizamos el Axioma 3

   $wp(max := 0; seg := 0; i := 0, i := 0 \wedge max := 0 \wedge escrutinioValido) \equiv wp(max := 0; seg := 0,$
   $wp(i := 0, i := 0 \wedge max := 0 \wedge escrutinioValido))$

   - $F1 \equiv wp(i := 0, i := 0 \wedge max := 0 \wedge escrutinioValido) \equiv def(0) \wedge_L 0 := 0 \wedge max := 0 \wedge escrutinioValido \equiv True \wedge_L True \wedge max := 0 \wedge escrutinioValido \equiv max := 0 \wedge escrutinioValido$

   Volvemos a la wp grande:

   $wp(max := 0; seg := 0, F1) \equiv wp(max := 0, wp(seg := 0; F1))$

   - $F2 \equiv wp(seg := 0, F1) \equiv def(0) \wedge_L max := 0 \wedge escrutinioValido \equiv True \wedge_L max := 0 \wedge escrutinioValido \equiv max := 0 \wedge escrutinioValido$

   Volvemos a la wp grande:

   $wp(max := 0, F2) \equiv def(0) \wedge_L 0 = 0 \wedge escrutinioValido \equiv True \wedge_L True \wedge escrutinioValido \equiv escrutinioValido$

   $\implies wp(max := 0; seg := 0; i := 0, i := 0 \wedge max := 0 \wedge escrutinioValido) \equiv escrutinioValido$

   Volviendo a nuestra implicacion inicial:

   - $escrutinioValido \implies escrutinioValido$

   La implicacion es True

   $P_{C_1} \implies wp(Ciclo1, Q_{C_1})$
   Vamos a utilizar el teorema del invariante (porque no se puede calcular esa wp en general):

   ---

2. $P_{C_1} \implies I_1$
   $\equiv ((i := 0 \wedge max := 0 \wedge escrutinioValido) \implies [((0 \leq i \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge$
   $((\forall k:\mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))))]$

Para resolver la implicacion vale revisar solamente el caso donde el termino izquierdo sea Verdadero.

$\equiv (i := 0 \wedge max := 0 \wedge escrutinioValido) \implies ((0 \leq 0 \leq |escrutinio| - 1) \wedge (0 \leq 0 < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq 0) \wedge (k \neq 0) \implies (escrutinio[0] > escrutinio[k]))$

$\equiv (i := 0 \wedge max := 0 \wedge escrutinioValido) \implies ((0 \leq |escrutinio| - 1) \wedge (0 < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq 0) \wedge (k \neq 0) \implies (escrutinio[0] > escrutinio[k]))$

Como son varios terminos unidos por $\wedge$, vale revisar la implicacion de cada uno de ellos:

$i := 0 \wedge max := 0 \wedge escrutinioValido \implies (0 \leq |escrutinio| - 1) \wedge (0 < |escrutinio| - 1)$

$escrutinioValido \implies (|escrutinio| > 2) \implies (1 \leq |escrutinio|) \wedge (1 < |escrutinio|)$

Esta primer parte es True. Ahora la otra:

$(i := 0 \wedge max := 0) \implies ((\forall k : \mathbb{Z})(k = 0) \wedge (k \neq 0) \implies (escrutinio[max] > escrutinio[k]))$

Como la implicacion de para todo k no esta en el rango (k tiene que ser distinto de cero y no lo es), la implicacion es True siempre.

Por ende, la implicacion principal que buscabamos demuestra ser True.

---

3. $\{I_1 \wedge B_1\}S2\{I_1\}$

Para probar que la tripla es valida hay que demostrar que: $I_1 \wedge B_1 \implies wp(S2, I_1)$

$I_1 \equiv (0 \leq i \leq |escrutinio|-1) \wedge (0 \leq max < |escrutinio|-1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))$

$B_1 \equiv (i < |escrutinio| - 1)$

$wp(S2, I_1) \equiv wp(if(escrutinio[i] > escrutinio[max])then : max := i; else : skip; endif; i := i + 1, I_1)$

$\equiv wp(ifescrutinio[i] > escrutinio[max]thenmax := ielseskipendif, wp(i := i + 1, I_1))$

Axioma 1:

$wp(i := i + 1, I_1)$

$\equiv def(i+1) \wedge (0 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i+1) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))$

$\equiv (1 \leq i+1 \leq |escrutinio|) \wedge (0 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i+1) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))$

Los primeros dos terminos pueden ser agrupados, ya que tienen info similar que sirve para acotar:

$(1 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i+1) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k])) \equiv G1$

Volvemos al wp principal:

$wp(ifescrutinio[i]¿escrutinio[max]thenmax := ielseskipendif, G1)$

Utilizamos el axioma 4:

$def(escrutinio[i] > escrutinio[max]) \wedge_L (((escrutinio[i] > escrutinio[max]) \wedge wp(max := i, G1)) \vee$

$(\neg(escrutinio[i] > escrutinio[max]) \wedge wp(skip, G1))$

$\equiv (1 \leq i+1 \leq |escrutinio|) \wedge_L (0 \leq max < |escrutinio| - 1) \wedge_L ((escrutinio[i] > escrutinio[max]) \wedge def(i) \wedge (1 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq i < |escrutinio| - 1) \wedge [(\forall k : Z)(0 \leq k \leq i+1) \wedge (k \neq i) \implies (escrutinio[i] > escrutinio[k])) \vee ((escrutinio[i][max]) \wedge (1 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge [(\forall k : Z)(0 \leq k \leq i+1) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))))$

$I_1 \wedge B_1 \implies wp(\dots)$ $((0 \leq i \leq |escrutinio|-1) \wedge (0 \leq max < |escrutinio|-1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))) \wedge (i < |escrutinio| - 1) \implies wp(..)$

Juntamos los dos terminos del rango de i:

$((0 \leq i < |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))) \implies (1 \leq i+1 \leq |escrutinio|) \wedge_L (0 \leq max < |escrutinio| - 1) \wedge_L ((escrutinio[i] > escrutinio[max]) \wedge def(i) \wedge (1 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq i < |escrutinio| - 1) \wedge (\forall k : Z)(0 \leq k \leq i+1) \wedge (k! = i) \implies (escrutinio[i] > escrutinio[k])) \vee ((escrutinio[i][max]) \wedge (1 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge ((\forall k : \mathbb{Z})$ $(0 \leq k \leq i+1) \wedge (k \neq max)| > (escrutinio[max] > escrutinio[k])))))$

Los convierto en lo mas parecido posible a lo que tenemos en el lado izquierdo de la implicacion, que es True. Primero probamos la primer parte que tiene que ser True obligatoriamente para que la implicacion sea verdadera. Y luego vemos cual de las dos expresiones del .ºr.es True, ya que asi se demuestra la implicacion.

Primero la parte True:

$((0 \leq i < |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))) \implies (1 \leq i+1 \leq |escrutinio|) \wedge_L (0 \leq max < |escrutinio| - 1)$

Modificamos el primer termino de la derecha (el del rango de i), para que quede a nuestra conveniencia:

$(1 \leq i+1 \leq |escrutinio|) \equiv (0 \leq i \leq |escrutinio| - 1)$

$((0 \leq i < |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))) \implies (1 \leq i+1 \leq |escrutinio|) \wedge_L (0 \leq max < |escrutinio| - 1)$

Es True.

Ahora chequeamos el primer termino del Or:

$((0 \leq i < |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))) \implies ((escrutinio[i] > escrutinio[max]) \wedge def(i) \wedge (1 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq i < |escrutinio| - 1) \wedge [(\forall k : \mathbb{Z})(0 \leq k \leq i+1) \wedge (k \neq i) \implies (escrutinio[i] > escrutinio[k]))]$

Ya podemos asumir que es falso solo viendo el ultimo termino. "$escrutinio[i] > escrutinio[max]$" **Falso! Si vemos la parte izquierda de la implicacion dice que escrutinio[max] es el maximo. Esto esta dicho en el para todo k en Z...**

Parece que el primero no va. Hay que probar por el segundo termino:

$(\ (0 \leq i < |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k])))$

$\implies$

((escrutinio[i]$\leq escrutinio[max]) \wedge (1 \leq i+1 \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge [(\forall k : Z)(0 \leq k < i+1) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k]))]))$

- $(0 \leq max < |escrutinio| - 1)$ es True. Aparece completamente igual del lado izquierdo
- Luego el termino de i lo podemos convertir: (1-1 $\leq i+1-1 \leq |escrutinio| - 1 - 1) = (0 \leq i \leq |escrutinio| - 2) = (0 \leq i < |escrutinio| - 1). Ylisto((escrutinio[i] \leq escrutinio[max]). Estaenlaparteizquierda$

- Y el para todo es igual.

---

4. $I_1 \wedge \neg B_1 \implies Q_C$
   - $I_1 \wedge \neg B_1 \equiv (0 \leq i \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k])) \wedge (i \geq |escrutinio| - 1)$
   - $Q_C \equiv (i = |escrutinio| - 1) \wedge (esMaximo(max, lista))$

$I_1 \wedge \neg B_1 \implies Q_C \equiv (0 \leq i \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k])) \wedge (i \geq |escrutinio| - 1) \implies (i = |escrutinio| - 1) \wedge (esMaximo(max, lista))$

Miramos parte por parte:

- $(0 \leq i \leq |escrutinio| - 1) \wedge (i \geq |escrutinio| - 1) \implies i = |escrutinio| - 1$
- $((0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max) \implies (escrutinio[max] > escrutinio[k])) \implies (esMaximo(max, lista))$

Ambas implicaciones se cumplen

---

5. $\{I_1 \wedge B_1 \wedge v_0 = fv\}S\{fv < v_0\}$

Para probar que la tripla es valida hay que demostrar que: $\{I \wedge B \wedge v_0 = fv\} \implies wp(S2, fv < v_0)$

- $\{I_1 \wedge B_1 \wedge v_0 = fv\} \equiv \{(0 \leq i \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (((\forall k : \mathbb{Z})(0 \leq k \leq i) \wedge (k \neq max)) \implies (escrutinio[max] > escrutinio[k])) \wedge (i < |escrutinio| - 1) \wedge (v_0 = |escrutinio| - 1 - i)\}$

Resolvemos la Wp

$wp(S2, fv{<}v_0) = wp(if(escrutinio[i] > escrutinio[max])then(max := i)else(skip)endif; i := i + 1,$
$|escrutinio| - 1 - i < v_0) =$

$wp(if(escrutinio[i] > escrutinio[max])then(max := i)else(skip)endif; wp(i := i + 1, |escrutinio| - 1 - i < v_0))$

$wp(i := i + 1, |escrutinio| - 1 - i < v_0)) = |escrutinio| - 1 - (i + 1) < v_0$

$$\implies H1 = |escrutinio| - 2 - i < v_0$$

$wp(if(escrutinio[i] > escrutinio[max])then(max := i)else(skip)endif; H1) \equiv$

$def(escrutinio[i] > escrutinio[max]) \wedge_L ((escrutinio[i] > escrutinio[max] \wedge wp(max := i, H1))$
$v(\neg(escrutinio[i] > escrutinio[max]]) \wedge wp(skip, H1)))def(escrutinio[i] > escrutinio[max]) = (0 \leq i < |escrutinio|) \wedge$
$(0 \leq max < |escrutinio| - 1)$

$$wp(max := i, H1) = def(i) \wedge_L |escrutinio| - 2 - i < v_0$$

$$wp(skip, H1) = H1$$

$wp(S, fv¡v_0) \equiv (0 \leq i < |escrutinio| - 1) \wedge (0 \leq max < |escrutinio|) \wedge_L$
$((escrutinio[i] > escrutinio[max] \wedge |escrutinio| - 2 - i < v_0) \vee ((escrutinio[i] \leq escrutinio[max]]) \wedge$
$|escrutinio| - 2 - i < v_0))$

Reemplazamos todo en $\{I \wedge B \wedge v_0 = fv\} \implies wp(S2, fv < v_0)$

$0 \leq i \leq |escrutinio| - 1 \wedge 0 \leq max < |escrutinio| - 1 \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i \wedge k \neq max \implies$
$escrutinio[max] > escrutinio[k]))i < |escrutinio| - 1v_0 = |escrutinio| - 1 - i \implies (0 \leq i < |escrutinio|) \wedge$
$(0 \leq max < |escrutinio| - 1) \wedge_L ((escrutinio[i] > escrutinio[max] \wedge |escrutinio| - 2 - i < v_0) \vee$
$((escrutinio[i] \leq escrutinio[max]]) \wedge |escrutinio| - 2 - i < v_0))$

Separamos por partes:

$0 \leq i \leq |escrutinio| - 1i < |escrutinio| - 1 \implies (0 \leq i < |escrutinio|)$

$(0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i \wedge k \neq max \implies escrutinio[max] > escrutinio[k])) \implies$
$(0 \leq max < |escrutinio| - 1) \wedge_L (escrutinio[i] > escrutinio[max]) \vee ((escrutinio[i] \leq escrutinio[max]])$

$v_0 = |escrutinio| - 1 - i \implies |escrutinio| - 2 - i < v_0 \vee |escrutinio| - 2 - i < v_0$

---

6. $I \wedge fv \leq 0 \implies \neg B$

   $I \wedge fv \leq 0 \equiv (0 \leq i \leq |escrutinio| - 1) \wedge (0 \leq max < |escrutinio| - 1) \wedge (\forall k : \mathbb{Z})((0 \leq k \leq i) \wedge (k \neq max) \implies escrutinio[max] > escrutinio[k])) \wedge (|escrutinio| - 1 - i \leq 0)$

   $\neg B : i >= |escrutinio| - 1$

Juntamos todo:

$0 \leq i \leq |escrutinio| - 1 \wedge 0 \leq max < |escrutinio| - 1 \wedge (\forall k : \mathbb{Z})(0 \leq k \leq i \wedge k \neq max \implies escrutinio[max] > escrutinio[k])) \wedge |escrutinio| - 1 - i \leq 0 \implies i >= |escrutinio| - 1$

$$(0 \leq i \leq |escrutinio| - 1) \wedge (|escrutinio| - 1 - i \leq 0)(i >= |escrutinio| - 1)$$

La implicacion es Verdadera

---

7. $\{Q_C1\}S3\{P_C2\}$

Para probar que la tripla es valida hay que demostrar que: $Q_C1 \implies wp(S3, P_C2)$

$Q_C1 : i = |escrutinio| - 1 \wedge esMaximo(max, escrutinio)$

$S3 : if(max = 0)then seg := 1 else seg := 0; j := 0)$

$P_C2 : j = 0 \wedge 0 <= seg <= 1 \wedge seg \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg]$

Buscamos la $wp(S3, P_C2)$ **(Con axioma 4):**

$wp(S3, PC2) = def(max = 0) \wedge_L (max = 0 \wedge wp(seg := 1, j = 0 \wedge (0 \leq seg \leq 1) \wedge (seg \neq max) \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg])) \vee (\neg(max = 0) \wedge wp(seg := 0; j := 0, j = 0 \wedge (0 \leq seg \leq 1) \wedge (seg \neq max) \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg]))$

$wp(seg := 1, j = 0 \wedge 0 \leq seg \leq 1 \wedge seg \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg])\ def(1) \wedge j = 0 \wedge 0 \leq 0 \leq 1 \wedge 0 \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[0]$

$wp(seg := 0; j := 0, j = 0 \wedge 0 <= seg <= 1 \wedge seg \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg]))$

$wp(j := 0, j = 0 \wedge 0 <= seg <= 1 \wedge seg \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg])$
$H1 = def(0) \wedge 0 = 0 \wedge 0 <= seg <= 1 \wedge seg \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg]$

$wp(seg := 0; H1) = def(0) \wedge (0 <= 0 <= 1) \wedge 0 \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[0]$

$\implies wp(S3, PC2) = def(max = 0) \wedge_L (max = 0 \wedge j = 0 \wedge 0 <= 0 <= 1 \wedge 0 \neq max \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[0])((max \neq 0) \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[0])$

$Q_C1 \implies wp(S3, PC2) \equiv$

$i = |escrutinio| - 1 \wedge esMaximo(max, escrutinio) \implies (max = 0) \wedge (j = 0) \wedge (0 <= 0 <= 1) \wedge (0 \neq max) \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[0])((max \neq 0) \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[0])\ esMaximo(max, escrutinio) \implies esMaximo(max, escrutinio) esMaximo(max, escrutinio)$

$P_{C_2} \implies wp(Ciclo2, Q_{C_2})$

Vamos a utilizar el teorema del invariante (porque no se puede calcular esa wp en general):

---

8. $P_{C_2} \implies I_2$

$\equiv (j = 0 \wedge (0 <= seg <= 1) \wedge (seg \neq max) \wedge esMaximo(max, escrutinio) \wedge_L escrutinio[max] > escrutinio[seg] \implies (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k]))$

Para resolver la implicacion vale revisar solamente el caso donde el termino izquierdo sea Verdadero.

Como son varios terminos unidos por $\wedge$**, vale revisar la implicacion de cada uno de ellos:**

$j = 0 \implies 0 \leq j \leq |escrutinio| - 1?0 \leq 0 \leq |escrutinio| - 1$

$seg \neq max \implies max \neq seg$

$esMaximo(max, escrutinio) \implies esMaximo(max, escrutinio)$

$(escrutinio[max] > escrutinio[seg] \implies esMaximo(max, escrutinio)$

(Es cierto por la definicion de esMaximo)

$0 <= seg <= 1 \implies 0 \leq seg < |escrutinio| - 1$

(sabemos que $|escrutinio| > 2$ por lo tanto lo mínimo que puede ser $|escrutinio|$ es 3, entonces $|escrutinio| \geq 3 \wedge |escrutinio| - 1 \geq 2 \implies 0 \leq seg < 2 \leq |escrutinio| - 1$ entonces $0 <= seg <= 1 \implies 0 <= seg < 2$)

Tenemos que j=0 y $(0 \leq seg \leq 1) \implies (\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k]))$
Entonces nos queda $(\forall k : \mathbb{Z})(0 \leq k \leq 0 \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k])) \implies k = 0$
$\implies (\forall k : \mathbb{Z})(0 \leq 0 \leq 0 \wedge 0 \neq max \wedge 0 \neq seg \implies escrutinio[seg] > escrutinio[0]))$

- $Sitomamosseg = 0((0 \leq 0 \leq 0) \wedge 0 \neq max \wedge (0 \neq 0)$**(False)** $\implies escrutinio[0] > escrutinio[0]))$
  Como el primer termino es False, la implicacion me queda True

- $Sitomamosseg = 1((0 \leq 0 \leq 0) \wedge (0 \neq max) \wedge (0 \neq 1) \implies escrutinio[1] > escrutinio[0]))$
  Como el primer termino es False, la implicacion me queda True

---

9. $\{I_2 \wedge B_2\}S4\{I_2\}$

Para demostrar que la tripla es valida tenemos que probar que: $I_2 \wedge B_2 \implies wp(S4, I_2)$

$I_2 \wedge B_2 \equiv (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio)$
$\wedge (\forall k : \mathbb{Z})(0 \leq k \leq j) \wedge (k \neq max) \wedge (k \neq seg) \implies escrutinio[seg] > escrutinio[k])) \wedge j < |escrutinio| - 1$

$Calculamoswp(S4, I_2)$

$wp(S4, I_2) \equiv wp(if(j \neq segescrutinio[j] > escrutinio[seg])then(seg := j); else(skip); endif; j := j + 1,$
$0 \leq j \leq |escrutinio| - 1 \wedge 0 \leq seg < |escrutinio| - 1 \wedge max \neq seg \wedge esMaximo(max, escrutinio) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k]))$

- $Wp(j := j+1, (0 \leq j \leq |escrutinio|-1) \wedge (0 \leq seg < |escrutinio|-1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge$
  $(\forall k : \mathbb{Z})((0 \leq k \leq j) \wedge (k \neq max) \wedge (k \neq seg) \implies escrutinio[seg] > escrutinio[k])) :$

  $def(j+1) \wedge (0 \leq j+1 \leq |escrutinio|-1) \wedge (0 \leq seg < |escrutinio|-1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge$
  $(\forall k : \mathbb{Z})(0 \leq k \leq j + 1 \wedge (k \neq max) \wedge (k \neq seg) \implies escrutinio[seg] > escrutinio[k])$

  $H1 \equiv True \wedge -1 \leq j \leq |escrutinio| - 2 \wedge 0 \leq seg < |escrutinio| - 1 \wedge max \neq seg \wedge esMaximo(max, escrutinio) \wedge$
  $(\forall k : \mathbb{Z})(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k])$

$wp(if(j \neq seg \wedge escrutinio[j] > escrutinio[seg])then(seg := j)else(skip)endif, H1)$

Por el axioma 4, sabemos que:

$def(j \neq seg \wedge escrutinio[j] > escrutinio[seg]) \wedge_L (j \neq seg \wedge escrutinio[j] > escrutinio[seg] \wedge wp(seg := j, H1)) \vee$
$(\neg(j \neq seg \wedge escrutinio[j] > escrutinio[seg])wp(skip, H1))$

- $def(j \neq seg \wedge escrutinio[j] > escrutinio[seg]) \equiv (0 \leq j \leq |escrutinio| - 1 \wedge 0 \leq seg \leq |escrutinio| - 1)$
- $wp(seg := j, H1) \equiv def(j) \wedge (-1 \leq j \leq |escrutinio|-2 \wedge 0 \leq j < |escrutinio|-1 \wedge max \neq j \wedge esMaximo(max, escrutinio) \wedge$
  $(\forall k : \mathbb{Z})(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq j \implies escrutinio[j] > escrutinio[k]))$
- $\neg(j \neq seg \wedge escrutinio[j] > escrutinio[seg]) \equiv (j = seg \vee escrutinio[j] \leq escrutinio[seg])$
- $wp(skip, H1) = H1$

wp(S4,I2) $\equiv (0 \leq j \leq |escrutinio| - 1 \wedge 0 \leq seg \leq |escrutinio| - 1) \wedge_L (j \neq segescrutinio[j] > escrutinio[seg]$
$\wedge (-1 \leq j \leq |escrutinio| - 2 \wedge 0 \leq j < |escrutinio| - 1 \wedge max \neq j \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})$
$(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq j \implies escrutinio[j] > escrutinio[k]))) \vee ((j = seg \vee escrutinio[j] \leq escrutinio[seg]) - 1$

$\leq j \leq |escrutinio| - 2 \wedge 0 \leq seg < |escrutinio| - 1 \wedge max \neq seg \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})$
$(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k]))$

Juntamos todo:

$0 \leq j \leq |escrutinio| - 1 \wedge 0 \leq seg < |escrutinio| - 1 \wedge max \neq seg \wedge esMaximo(max, escrutinio) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k])) \wedge j < |escrutinio| - 1 \implies$
$(0 \leq j \leq |escrutinio| - 1 \wedge 0 \leq seg \leq |escrutinio| - 1) \wedge_L (j \neq segescrutinio[j] > escrutinio[seg] \wedge$
$(-1 \leq j \leq |escrutinio| - 2 \wedge 0 \leq j < |escrutinio| - 1 \wedge max \neq j \wedge esMaximo(max, escrutinio) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq j \implies escrutinio[j] > escrutinio[k]))) \vee ((j = seg \vee escrutinio[j]$
$\leq escrutinio[seg]) \wedge -1 \leq j \leq |escrutinio| - 2 \wedge 0 \leq seg < |escrutinio| - 1 \wedge max \neq seg \wedge esMaximo(max, escrutinio$
$) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k])$

Separamos por partes:

$(0 \leq j \leq |escrutinio| - 1) \wedge (j < |escrutinio| - 1) \implies (0 \leq j \leq |escrutinio| - 1) \wedge (-1 \leq j \leq |escrutinio| - 2) \wedge$
$(0 \leq j < |escrutinio| - 1) \vee (-1 \leq j \leq |escrutinio| - 2)$

$0 \leq seg < |escrutinio| - 1 \wedge max \neq seg \wedge (\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg$
$\implies escrutinio[seg] > escrutinio[k])) \implies 0 \leq seg \leq |escrutinio| - 1) \wedge_L (j \neq segescrutinio[j] > escrutinio[seg] \wedge$
$(\forall k : \mathbb{Z})(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq j \implies escrutinio[j] > escrutinio[k]))) \vee ((j = seg \vee escrutinio[j] \leq escrutinio[seg]) \wedge$
$0 \leq seg < |escrutinio| - 1 \wedge max \neq seg \wedge (\forall k : \mathbb{Z})(0 \leq k \leq j + 1 \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k]))$

esMaximo(max, escrutinio) $\implies esMaximo(max, escrutinio) \vee esMaximo(max, escrutinio)$

Por lo tanto la implicacion es correcta

---

10. $I_2 \wedge \neg B_2 \implies Q_C 2$

$(I_2 \equiv (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge$
$(\forall k : \mathbb{Z})(0 \leq k < j \wedge k! = max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k]))$

$\neg B_2 \equiv (j >= |escrutinio| - 1)$

$Q_C 2 \equiv (j = |escrutinio| - 1) \wedge (max \neq seg) \wedge (esMaximo(max, escrutinio)) \wedge_L (esSegundoMaximo(max, seg, escrutinio))$

Juntamos todo:

$(0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})(0 \leq k < j) \wedge k \neq max \wedge k \neq seg \implies$
$escrutinio[seg] > escrutinio[k])) \wedge (|escrutinio| - 1 >= j >= |escrutinio| - 1) \implies (j = |escrutinio| - 1) \wedge$
$(max \neq seg) \wedge (esMaximo(max, escrutinio)) \wedge_L (esSegundoMaximo(max, seg, escrutinio))$

Como j = $|escrutinio|$ -1, cambiamos todos los j por $|escrutinio| - 1$:

$(0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1) \wedge (k \neq max)$
$\wedge (k \neq seg) \implies escrutinio[seg] > escrutinio[k])) \wedge (|escrutinio| - 1 >= j >= |escrutinio| - 1) \implies$
$(j = |escrutinio| - 1) \wedge (max \neq seg) \wedge (esMaximo(max, escrutinio)) \wedge_L (esSegundoMaximo(max, seg, escrutinio))$

$$(|escrutinio| - 1 >= j >= |escrutinio| - 1) \implies (j = |escrutinio| - 1)$$

$$(max \neq seg) \implies (max \neq seg)$$

$$esMaximo(max, escrutinio) \implies (esMaximo(max, escrutinio))$$

$((\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1) \wedge (k \neq max) \wedge (k \neq seg) \implies escrutinio[seg] > escrutinio[k]) \implies$
$(esSegundoMaximo(max, seg, escrutinio))$

True

---

11. $\{I_2 \wedge B_2 \wedge v_0 = fv\}S\{fv < v_0\}$

Para probar que la tripla es valida hay que demostrar que: $\{I_2 \wedge B_2 \wedge v_0 = fv\} \implies wp(S4, fv < v_0)$

$I_2 \wedge B_2 \wedge v_0 = fv \equiv (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k]) \wedge (j < |escrutinio| - 1) \wedge v_0 = |escrutinio| - 1 - j)$

$wp(S4, fv < v_0) \equiv wp(if(j \neq seg \wedge escrutinio[j] > escrutinio[seg])then(seg := j)else(skip)endif; j := j + 1, |escrutinio| - 1 - j < v_0)$

$H1 \equiv wp(j := j + 1, |escrutinio| - 1 - j < v_0) \equiv def(j + 1) \wedge |escrutinio| - 2 - j < v_0$

$$H1 \equiv |escrutinio| - 2 - j < v_0 \wedge (0 \leq j \leq |escrutinio| - 1)$$

Ahora volvemos a la wp principal

$wp(if(j \neq seg \wedge escrutinio[j] > escrutinio[seg])then(seg := j)else(skip)endif, H1)$

$\equiv def(j \neq seg) \wedge (escrutinio[j] > escrutinio[seg]) \wedge_L$

$((\text{j} \neq seg \wedge escrutinio[j] > escrutinio[seg]) \wedge wp(seg := j, |escrutinio| - 2 - j < v_0)) \vee ((j = segmax$
$\vee escrutinio[j] <= escrutinio[seg])wp(skip, (|escrutinio| - 2 - j < v_0) \wedge (0 \leq j \leq |escrutinio| - 1)))$

La parte derecha de la implicacion esta compuesta por un termino que debe ser True si o si(ya que esta con el conector logico "and" , y luego, dos expresiones conectadas por un "Or". Por lo que al menos una de esas dos debe ser True. Todo esto, para demostrar que la implicacion en si es True.

- $def((j \neq seg) \wedge (escrutinio[j] > escrutinio[seg])) \equiv (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg \leq |escrutinio| - 1)$

- wp(seg:=j, ($|escrutinio|$-2-j < v_0)$\wedge(0 \leq j \leq |escrutinio| - 1)) \equiv def(j) \wedge ($—$escrutinio$——$2 - j$¡$v_0) \wedge (0 \leq j \leq |escrutinio| - 1)$
- wp(skip, $|escrutinio| - 2 - j < v_0$ $\wedge(0 \leq j \leq |escrutinio| - 1)) \equiv$—escrutinio—-2-j ¡$v_0$ $\wedge(0 \leq j \leq |escrutinio| - 1)$

$\implies wp(S4, fv < v0) \equiv (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg \leq |escrutinio| - 1) \wedge_L ((j \neq seg \wedge escrutinio[j] > escrutinio[seg]) \wedge |escrutinio| - 2 - j < v_0) \vee ((j = seg \vee escrutinio[j] <= escrutinio[seg]) \wedge |escrutinio| - 2 - j < v_0)$

Juntamos Todo: $\{I_2 \wedge B_2 \wedge v_0 = fv\} \implies wp(S4, fv < v_0)$

$(0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})((0 \leq k \leq j) \wedge (k \neq max) \wedge (k \neq seg) \implies escrutinio[seg] > escrutinio[k]) \wedge (j < |escrutinio| - 1) \wedge (v0 = |escrutinio| - 1 - j)) \implies (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg \leq |escrutinio| - 1) \wedge_L ((j \neq seg \wedge escrutinio[j] > escrutinio[seg]) \wedge |escrutinio| - 2 - j < v_0) \vee ((j = seg \vee escrutinio[j] <= escrutinio[seg])|escrutinio| - 2 - j < v_0)$

$(0 \leq j \leq |escrutinio| - 1) \implies (0 \leq j \leq |escrutinio| - 1)$

v0 =$|escrutinio|$-1-j) $\implies$—escrutinio—$-2 - j < v_0 \vee$—escrutinio—-2-j ¡v_0

---

12. $I_2 \wedge (fv \leq 0) \implies \neg B_2$

$I_2 \wedge (fv \leq 0) \equiv (0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k])) \wedge (|escrutinio| - 1 - j \leq 0)$

$\neg B2 \equiv (j >= |escrutinio| - 1)$

Juntamos todo:

$(((0 \leq j \leq |escrutinio| - 1) \wedge (0 \leq seg < |escrutinio| - 1) \wedge (max \neq seg) \wedge esMaximo(max, escrutinio) \wedge (\forall k : \mathbb{Z})(0 \leq k \leq j \wedge k \neq max \wedge k \neq seg \implies escrutinio[seg] > escrutinio[k])) \wedge (|escrutinio| - 1 - j \leq 0)) \implies j >= |escrutinio| - 1$

$(0 \leq j \leq |escrutinio| - 1) \wedge (|escrutinio| - 1 - j \leq 0) \implies (j >= |escrutinio| - 1)$

Se cumple la implicacion.

---

13. $\{Q_C 2\} S5 \{Post\}$

Para probar que la tripla es valida hay que demostrar que: $Q_C \implies wp(codigoposterioralciclo, Post)$

Post: $(\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge (esMaximo(escrutinio[i], escrutinio) \wedge_L (res_0 = i) \wedge ((\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge esMaximo(escrutinio[i], escrutinio) \wedge (\exists j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \wedge i \neq j \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \wedge k \neq i) \implies (esMayor(escrutinio[j], escrutinio[k])) \wedge_L (res_1 = j)$

Codigo posterior al ciclo: $res[0] := max; res[1] := seg$

Calculamos la $wp(codigoposterioralciclo, Post)$**:**

$\implies wp(res[0] := max; res[1] := seg, Post) \equiv wp(res[0] := max, wp(res[1] := seg, Post))$

Primero $wp(res[1] := seg, POST)) \equiv$

- $H1 \equiv (0 \leq seg < |escrutinio| - 1) \wedge_L (\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge (esMaximo(escrutinio[i], escrutinio) \wedge_L (res_0 = i) \wedge ((\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge esMaximo(escrutinio[i], escrutinio) \wedge (\exists j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \wedge i \neq j \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \wedge k \neq i) \implies (esMayor(escrutinio[j], escrutinio[k])) \wedge_L (seg = j)$

Volvemos a la wp principal:

$wp(res[0] := max, H1) \equiv$

- $H2 \equiv (0 \leq max < |escrutinio| - 1) \wedge_L (0 \leq seg < |escrutinio| - 1) \wedge_L (\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge (esMaximo(escrutinio[i], escrutinio) \wedge_L (max = i) \wedge ((\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge$ $esMaximo(escrutinio[i], escrutinio) \wedge (\exists j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \wedge i \neq j \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \wedge k \neq i) \implies (esMayor(escrutinio[j], escrutinio[k])) \wedge_L (seg = j)$

Reemplazamos todo en la implicacion: $Q_C 2 \implies H_2$

$(j = |escrutinio| - 1) \wedge (max \neq seg) \wedge (esMaximo(max, escrutinio)) \wedge_L (esSegundoMaximo(max, seg, escrutinio)) \implies$
$(0 \leq max < |escrutinio| - 1) \wedge_L (0 \leq seg < |escrutinio| - 1) \wedge_L (\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1$
$\wedge (esMaximo(escrutinio[i], escrutinio) \wedge_L (max = i) \wedge$
$((\exists i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \wedge esMaximo(escrutinio[i], escrutinio) \wedge (\exists j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \wedge i \neq j \wedge$
$(\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \wedge k \neq i) \implies (esMayor(escrutinio[j], escrutinio[k])) \wedge_L (seg = j)$

Separemos por partes:

$esMaximo(max, list) \implies (0 \leq max < |escrutinio| - 1)(esSegundoMaximo(max, seg, escrutinio)) \implies$
$(0 \leq seg < |escrutinio| - 1)$

Esto es verdadero porque ya sabemos que hay un maximo:
$(\exists i : \mathbb{Z})(0 \leq i < |escrutinio| 1) \wedge (esMaximo(escrutinio[i], escrutinio) \wedge_L (max = i)$

Por esto probamos que el ciclo es correcto