DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

TP: elecciones presidenciales

18 de septiembre de 2023 — Algoritmos y estructura de datos II

The Cooks

Integrante	LU	Correo electrónico
Melli, Tomas Felipe	371/22	tomas.melli1@gmail.com
Seltzer, Ramiro	715/22	ramiroseltzer@gmail.com
Sassot, Maria	38/23	sassotmaria@gmail.com
Valencia, Juan Segundo	705/22	reamnobis@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

1.1. hayBallotage

`proc` `hayBallotage` (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : Bool
`requiere` $\{|escrutinio| \geq 3 \land sonDistintos(escrutinio)\}$
`asegura` $\{(\forall i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \longrightarrow_L ((escrutinio[i]/total(escrutinio) \leq 0,45) \land_L$
$((\forall j : \mathbb{Z})(0 \leq j < |escrutinio| - 1 \land esMaximo(j, escrutinio)) \longrightarrow_L (escrutinio[j]/total(escrutinio) \leq 0,45)$
$\land\ difMenorA10(escrutinio, j))))\}$

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` $\{SonDistintos(escrutinio\_diputados) \land_L SonDistintos(escrutinio\_presidencial)$
$\land_L SonDistintos(escrutinio\_senadores) \land_L |escrutinio\_presidencial| > 2 \land_L |escrutinio\_senadores| > 2 \land_L$
$|escrutinio\_diputados| > 2\}$
`asegura` $\{res = True \iff (sumaVotos(escrutinio\_presidencial) \neq sumaVotos(escrutinio\_senadores))$
$\lor (sumaVotos(escrutinio\_senadores) \neq sumaVotos(escrutinio\_diputados)$
$\lor (sumaVotos(escrutinio\_presidencial) \neq sumaVotos(escrutinio\_diputados))\}$

1.3. obtenerSenadoresEnProvincias

`proc` `obtenerSenadoresEnProvincias` (in escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$x$\mathbb{Z}$
`requiere` $\{|escrutinio| \geq 3 \land sonDistintos(escrutinio)\}$
`asegura` $\{(\forall j : \mathbb{Z})((0 \leq j < |escrutinio| - 1 \land j \neq res_0 \land j \neq res_1) \longrightarrow_L (escrutinio[j] < escrutinio[res_0] \land$
$escrutinio[j] < escrutinio[res_1])) \land escrutinio[res_0] > escrutinio[res_1]\}$

1.4. calcularDHontEnProvincia

`proc` `calcularDHondtEnProvincia` (in cant_bancas : $\mathbb{Z}$, in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $seq\langle seq\langle\mathbb{Z}\rangle\rangle$
`requiere` $\{|escrutinio| > 2 \land cant\_bancas \geq 1\}$
`asegura` $\{(|res| = cant\_bancas \land_L |res| = |escrutinio| - 1) \land ((\forall i : \mathbb{Z})(0 \leq i < |escrutinio| - 1) \longrightarrow_L$
$((\forall j : \mathbb{Z})(0 \leq j < cant\_bancas) \longrightarrow_L res[i][j] = divisón(escrutinio[i], j + 1)))\}$

1.5. obtenerDiputadosEnProvincias

`proc` `obtenerDiputadosEnProvincias` (in cant_bancas: $\mathbb{Z}$, in escrutinio : $seq\langle\mathbb{Z}\rangle$, in D´Hondt :$seq\langle seq\langle\mathbb{Z}\rangle\rangle$) : $seq\langle\mathbb{Z}\rangle$
`requiere` $\{cant\_bancas \leq \#dHontCumplenUmbrals(DHont, total(escrutinio)) \land_L |DHont| = cant\_bancas \land_L$
$coincideDhontEscrutinio(DHondt, escrutinio) \land_L sonDistintos(escrutinio) \land_L |escrutinio| > 2\}$
`asegura` $\{|escrutinio| - 1 = |res| \land_L (\forall i : \mathbb{Z})(0 \leq i < |res| \longrightarrow_L$
$res[i] = bancasAsignadas(i, escrutinio, 1, total(escrutinio), cant\_bancas)\}$

1.6. validarListasDiputadosEnProvincias

`proc` `validarListasDiputadosEnProvincias` (in cant_bancas:$\mathbb{Z}$,in escrutinio: $seq\langle\mathbb{Z}\rangle$,in D´Hondt: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) : Bool
`requiere` $\{cantbancas \geq 1\}$
`asegura` $\{|listas| > 1 \land_L (\forall i : \mathbb{Z})(0 \leq i < |listas| \longrightarrow_L (|listas[i]| == cantbancas) \land$
$((\#mujeres(listas[i]) == \#hombres(lista[i]) \lor (\#mujeres(listas[i]) + 1 == \#hombres(lista[i]) \lor$
$(\#mujeres(listas[i]) - 1 == \#hombres(lista[i]))))\}$

2. Implementaciones SmallLang

2.1. hayBallotage

```
-- total --
i := 0;
total := 0;
while(i < escrutinio.size() - 1 ) do
    total = total + escrutinio[i];
    i = i +1 ;
endwhile

-- Indicemaximo --
id_1 := 0;
j := 1 ;
while ( j < s.size() - 1) do
    if ( escrutinio[j] > escrutinio[id_1] ) do
        id_1 := j;
    else
        skip;
    end if;
j := j+1
endwhile


-- IndiceegundoMax --
escrutinio[id_1] := 0
id_2 := 0;
j := 1;
while ( j < s.size() - 1 ) do
    if ( escrutinio[j] > escrutinio[id_2] ) do
        id_2 := j;
    else
        skip;
    end if;
j := j+1
endwhile

-- diferenciaMenor10 --
diferenciaMenor10 := False
if (((escrutinio[id_1] - escrutinio[id_2]) div total) < 0.1 ) then
    diferenciaMenor10 := True;
else
    diferenciaMenor10 := False;

-- main --
if (escrutinio[id_1] div total > 0.45) then
    return false;
else
    if (escrutinio[id_1] div total > 0.40 && !(diferenciaMenor10))
        return false;
    else
        return true;
```

2.2. hayFraude

```
— suma_presidencial —
i := 0;
suma_presidencial := 0;
while(i < escrutinio_presidencial.size()) do
    suma_presidencial = suma_presidencial + escrutinio_presidencial[i];
    i = i +1 ;
endwhile

— suma_senadores —
i := 0;
suma_senadores := 0;
while(i < escrutinio_senadores.size()) do
    suma_senadores = suma_senadores + escrutinio_senadores[i];
    i = i +1 ;
endwhile

— suma_diputados —
i := 0;
suma_diputados := 0;
while(i < escrutinio_diputados.size()) do
    suma_diputados = suma_diputados + escrutinio_diputados[i];
    i = i +1 ;
endwhile

— main —
if (suma_presidencial != suma_senadores || suma_senadores != suma_diputados || suma_presidencial !=
    suma_diputados)
    res := True;
else
    res := False;
return res;
```

2.3. obtenerSenadoresEnProvincia

```
-- suma_presidencial --
i := 0;
suma_presidencial := 0;
while(i < escrutinio_presidencial.size() ) do
    suma_presidencial = suma_presidencial + escrutinio_presidencial[i];
    i = i + 1 ;
endwhile

-- suma_senadores --
i := 0;
suma_senadores := 0;
while(i < escrutinio_senadores.size() ) do
    suma_senadores = suma_senadores + escrutinio_senadores[i];
    i = i + 1 ;
endwhile

-- suma_diputados --
i := 0;
suma_diputados := 0;
while(i < escrutinio_diputados.size() ) do
    suma_diputados = suma_diputados + escrutinio_diputados[i];
    i = i + 1 ;
endwhile

-- IndiceMaximo --
id_1 := 0 ;
j := 1 ;
while ( j < s.size() - 1) do
    if ( escrutinio[j] > escrutinio[id_1] )
        id_1 := j;
    else
        skip;
    endif;
j := j+1
endwhile

id_2 := 0;
escrutinio [1d_1]:= 0
j := 1 ;
while ( j < s.size() - 1) do
    if ( escrutinio[j] > escrutinio[id_2] )
        id_2 := j;
    else
        skip;
    endif;
j := j+1
endwhile

return id_1 x id_2
```

2.4. validarListasDiputadosEnProvincias

```
–– verificador ––
i := 0;
j := 0;
verifica := true;
while (i < listas.size()) do
    if (listas[i].size() != cant_bancas) then
        verifica := false;
    else
        skip;
endwhile

–– hombre_mujer ––
i := 0;
j := 0;
h := 0;
m := 0;
while (i < listas.size()) do
    while(j < listas[i].size()) do
        if (listas[i][j][1] = 1)
            h := h + 1;
        else
            m =: m+ 1;
        j := j + 1;
    endwhiles
    i := i + 1;
endwhile


–– main ––
if (verifica = true and ((h = m) || (h = m− 1) || (h = m+ 1)))
    res := True;
else
    res := False;
return res;
```

3. Demostración correctitud

3.1. Correctitud hayFraude

Demostración 1
Comenzamos viendo la correctitud del bloque de código encargado de la asignación de variables. Tenemos que ver que valga:
Pre $\implies wp(i = 0; sumaPresidencial = 0, Pc) \equiv Pre \implies wp(i = 0, wp(sumaPresidencial = 0, i = 0 \wedge sumaPresidencial = 0))$
$\equiv Pre \implies wp(i = 0, def(sumaPresidencial) \wedge_L i = 0 \wedge 0 = 0)$
$\equiv Pre \Longrightarrow wp(i = 0, True \wedge_L i = 0 \wedge True)$
$\equiv Pre \Longrightarrow wp(i = 0, i = 0)$
$\equiv Pre \Longrightarrow def(i) \wedge_L 0 = 0$
$\equiv Pre \Longrightarrow true \iff True$

Queda probado la correctitud de la asignación inicial de las variables.

Demostramos la correctitud del ciclo y su terminación.

- $P_c \equiv \{i = 0 \wedge sumaPresidencial = 0\}$
- $Q_c \equiv \{sumaPresidencial = \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_presidencial[j]\}$
- $I \equiv \{0 \leq i \leq |escrutinio\_presidencial| \wedge_L sumaPresidencial = \sum_{j=0}^{i-1} escrutinio\_presidencial[j]\}$
- $B \equiv \{i < |escrutinio\_presidencial|\}$
- $f_v \equiv \{|escrutinio\_presidencial| - i\}$

Tenemos que ver que valga:

1. $P_c \implies I$
2. $\{I \wedge B\}S\{I\}$
3. $(I \wedge \neg B) \implies Q_c$
4. $\{I \wedge B \wedge fv = v_0\}S\{fv < v_0\}$
5. $(I \wedge fv \leq 0) \implies \neg B$

Demostracion 1
$P_c \implies I$
$\{i = 0 \wedge sumaPresidencial = 0\} \implies \{0 \leq i \leq |escrutinioPresidencial| \wedge_L sumaPresidencial = \sum_{j=0}^{i-1} escrutinioPresidencial[j]\}$

1. $i = 0 \implies 0 \leq i \leq |escrutinioPresidencial| \iff 0 \leq 0 \leq |escrutinioPresidencial|$
2. $i = 0 \wedge sumaPresidencial = 0 \implies sumaPresidencial = \sum_{j=0}^{i-1} escrutinioPresidencial[j] \iff$
   $0 = \sum_{j=0}^{-1} escrutinioPresidencial[j]$ vale por sumatoria en rango vacio

Demostracion 2
$\{I \wedge B\}S\{I\} \iff (I \wedge B) \implies WP(S, I)$
$\equiv (0 \leq i \leq |escrutinio\_presidencial| \wedge_L sumaPresidencial = \sum_{j=0}^{i-1} escrutinioPresidencial[j] \implies WP(S, I)$
$\equiv (0 \leq i \leq |escrutinio\_presidencial| \wedge_L sumaPresidencial = \sum_{j=0}^{i-1} escrutinioPresidencial[j] \implies$
$(0 \leq i \leq |escrutinio\_presidencial| \wedge_L sumaPresidencial = \sum_{j=0}^{i-1} escrutinioPresidencial[j]$

Por $(I \wedge B)$ sabemos que $0 \leq i < |escrutinio\_presidencial|$ y que $sumaPresidencial + escrutinioPresidencial[i] = \sum_{j=0}^{i} escrutinio[j] \iff sumaPresidencial = \sum_{j=0}^{i-1} escrutinioPresidencial[j]$. Queda probada la implicancia.

Demostracion 3
$(I \wedge \neg B) \implies Q_c$
$\{i = |escrutinio\_presidencial| \wedge sumaPresidencial = \sum_{j=0}^{i-1} escrutinio\_presidencial[j]\} \implies$
$\{sumaPresidencial = \sum_{j=0}^{|escrutinio\_presidencial|-1} escrutinio\_presidencial[j]\}$

Queda demostrada la implicancia si se reemplaza i en la sumatoria

En esta instancia podemos decir que el ciclo es parcialmente correcto.

Nos interesará evaluar la terminación del ciclo a partir de la axiomatización y la función invariante

Demostracion 4

$(I \wedge B \wedge fv = v_0) \implies wp(S, fv < v_0)$

$\{0 \leq i < |escrutinio\_presidencial| \wedge_L sumaPresidencial = \sum_{j=0}^{i-1} escrutinio\_presidencial[j] \wedge |escrutinio\_presidencial| - i = Vo\} \implies$

$0 \leq i < |escrutinio\_presidencial| \wedge |escrutinio\_presidencial| - i - 1 < V_0$

$0 \leq i < |escrutinio\_presidencial| \implies 0 \leq i < |escrutinio\_presidencial|$ y por otro lado tenemos que $|escrutinio\_presidencial| - i = v_0 \implies |escrutinio\_presidencial| - i - 1 < |escrutinio\_presidencial| - i \iff -1 < 0$

Demostracion 5

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

$\{0 \leq i < |escrutinio\_presidencial| \wedge_L sumaPresidencial = \sum_{j=0}^{i-1} escrutinio[j] \wedge |escrutinio\_presidencial| - i \leq 0\} \implies \{i \geq |escrutinio\_presidencial|\}$

como $|escrutinio\_presidencial| - i \leq 0 \iff |escrutinio\_presidencial| \leq i$ queda solamente ver que $\{i \geq |escrutinio\_presidencial|\}$ $\{i \geq |escrutinio\_presidencial|\}$ **trivial**.

Queda demostrado entonces que el ciclo es correcto y finaliza.

Habiendo hecho esto, la demostración para los ciclos restantes y sus asignaciones es análoga con diferencia de los nombres de las variables. Solamente nos quedan ver los siguientes puntos.

Veamos cuando $Q_{c_3} \implies \mathbf{WP}(S, Post)$ siendo Q_{c_3} la postcondición del último ciclo y Post la postcondición de mi especificación. Para mayor comodidad decimos que B = $(suma\_presidencial \neq suma\_senadores \vee suma\_senadores \neq suma\_diputados \vee suma\_presidencial \neq suma\_diputados)$.

$$Q_{c_3} \implies \mathbf{WP}(S, Post) \iff \{sumaDiputados = \sum_{j=0}^{|escrutinio\_diputados|-1} escrutinio\_diputados[j]\} \implies \mathbf{WP}(S, Post)$$

$$\begin{aligned}\mathbf{WP}(S, Post) &\equiv \mathbf{WP}(if(suma\_presidencial \neq suma\_senadores \vee suma\_senadores \neq suma\_diputados \vee \\ &\quad suma\_presidencial \neq suma\_diputados)\ then\ res := True\ else\ res := False, Post) \\ &\equiv def(suma\_presidencial \vee suma\_senadores \vee suma\_diputados) \wedge_L ((B \wedge \mathbf{WP}(S_1, Post)) \vee (\neg B \wedge \mathbf{WP}(S_2, Post))) \\ &\equiv True \wedge_L ((B \wedge (def(S_1) \wedge_L Post^{res}_{res=True})) \vee (\neg B \wedge (def(S_1) \wedge_L Post^{res}_{res=False})) \\ &\equiv ((B \wedge (True \iff (...))) \vee (\neg B \wedge (False \iff (...)))) \overset{distribuyendo}{\equiv} True \vee False \equiv True\end{aligned}$$

Como tenemos que la $\mathbf{WP}(S, Post) \equiv$ True y queremos probar que $Q_{c_3} \implies \mathbf{WP}(S, Post)$ entonces sabemos que $Q_{c_3} \implies True \iff True$ vale para toda proposición lógica como se quería probar.

3.2. Correctitud obtenerSenadoresEnProvincia

3.2.1. PRE $\implies$ $wp(asignación, Pre_{sumaPresidencial})$

$|escrutinio| \geq 3 \wedge sonDistintos(escrutinio) \implies wp(i := 0,\ wp(sumaPresidencial := 0, \{i := 0, sumaPresidencial := 0\}))$
$|escrutinio| \geq 3 \wedge sonDistintos(escrutinio) \implies True$

3.2.2. Q sumaPresidencial $\implies$ $wp(asignación, Pre_{sumaSenadores})$

$sumapresidencial = \sum_{j=0}^{|escrutinioPresidencial|-1} escrutinioPresidencial[j] \implies wp(i := 0, sumaSenadores := 0, i := 0 \wedge sumaSenadores := 0)$
$sumaPresidencial = \sum_{j=0}^{|escrutinioPresidencial|-1} escrutinioPresidencial[j] \implies True$

3.2.3. Q sumaSenadores $\implies$ $wp(asignación, Pre_{sumaDiputados})$

$sumaSenadores = \sum_{j=0}^{|escrutinioSenadores|-1} escrutinioSenadores[j] \implies wp(i := 0, sumaDiputados := 0, i := 0 \wedge sumaDiputados := 0)$
$sumaSenadores = \sum_{j=0}^{|escrutinioSenadores|-1} escrutinioSenadores[j] \implies True$

3.2.4. Q sumaDiputados $\implies$ $wp(asignación, Pre_{indiceMáximo})$

$sumaDiputados = \sum_{j=0}^{|escrutinioDiputados|-1} escrutinioDiputados[j] \implies wp(j := 1, id = 0, j := 1 \wedge id = 0)$
$sumaDiputados = \sum_{j=0}^{|escrutinioDiputados|-1} escrutinioDiputados[j] \implies True$

3.2.5. IndiceMaximo

Veamos la correctitud del bloque de código encargado de hallar el índice del maximo elemento de un escrutinio dado

- $P_c \equiv \{id = 0 \wedge j = 1 \wedge |escrutinio| \geq 3\}$
- $Q_c \equiv \{0 \leq id < |escrutinio| - 1 \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id]) \wedge j = |escrutinio| - 1$
- $I \equiv \{0 \leq id < |escrutinio| - 1 \wedge (1 \leq j \leq |escrutinio| - 1) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j \longrightarrow_L escrutinio[k] \leq escrutinio[id])$
- $B \equiv \{j < |escrutinio| - 1\}$

Tenemos que ver que valga:

1. $P_c \implies I$
2. $\{I \wedge B\}S\{I\}$
3. $(I \wedge \neg B) \implies Q_c$
4. $\{I \wedge B \wedge f_v = v_0\}S\{f_v < v_0\}$
5. $I \wedge f_v \leq 0 \implies \neg B$

Demostracion 1
$P_c \implies I$

$id = 0 \wedge j = 1 \wedge |escrutinio| \geq 3 \implies 0 \leq id < |escrutinio| - 1 \wedge 1 \leq j \leq |escrutinio| - 1 \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j \longrightarrow_L escrutinio[k] \leq escrutinio[id])$

$id = 0 \implies 0 \leq id < |escrutinio| - 1$
$id = 0 \implies 0 < |escrutinio| - 1$
$pues |escrutinio| >= 3 \equiv True$

$j = 1 \implies 1 \leq j \leq |escrutinio| - 1$
$j = 1 \implies 1 \leq |escrutinio| - 1$
$pues |escrutinio| >= 3 \equiv True$

$id = 0 \wedge j = 1 \wedge |escrutinio \geq 3 \implies (\forall k : \mathbb{Z})(0 \leq k < j \longrightarrow_L escrutinio[0] \leq escrutinio[0] \equiv True$

Demostracion 2
$\{I \wedge B\} S \{I\}$
I $\wedge B \implies wp(S, I)$

$I \wedge B \equiv 0 \leq id < |escrutinio| - 1 \wedge 1 \leq j \leq |escrutinio| - 1 \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j \longrightarrow_L escrutinio[k] \leq escrutinio[id]) \wedge (j < |escrutinio| - 1)$

$I \wedge B \equiv 0 \leq id \leq |escrutinio| - 1 \wedge 1 \leq j < |escrutinio| - 1 \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j \longrightarrow_L escrutinio[k] \leq escrutinio[id])$

siendo H $\equiv escrutinio[j] > escrutinio[id]$
wp(S,I) $\equiv wp(\textsf{if } H \textsf{ then } S_0 \textsf{ else } S_1 \textsf{ fi}, S_2, I)$
$wp(S, I) \equiv wp(\textsf{if } H \textsf{ then } S_0 \textsf{ else } S_1 \textsf{ fi}, wp(S_2, I))$

comenzamos viendo wp(S_2, I)
$wp(S_2, I) \equiv wp(j := j + 1, I)$
$wp(S_2, I) \equiv (0 \leq id < |escrutinio| - 1 \wedge 1 \leq j + 1 \leq |escrutinio| - 1 \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j + 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id]$

veamos
wp($\textsf{if } (escrutinio[j] > escrutinio[id]) \textsf{ then } id := j \textsf{ else } skip \textsf{ fi}$, wp($S_2, I$))

$\equiv def(escrutinio[j] > escrutinio[id]) \wedge_L ((escrutinio[j] > escrutinio[id]) \wedge wp(id := j, wp(S_2, I)) \vee ((escrutinio[j] > escrutinio[id]) \wedge wp(skip, wp(S_2, I))))$

$\equiv (0 \leq id < |escrutinio| - 1 \wedge 0 \leq j < |escrutinio| - 1) \wedge_L$
$((escrutinio[j] > escrutinio[id]) \wedge (0 \leq j < |escrutinio| - 1 \wedge 1 \leq j + 1 \leq |escrutinio| - 1 \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j + 1 \longrightarrow_L escrutinio[k] \leq escrutinio[j])))$
$\vee$
$((escrutinio[j] \leq escrutinio[id])(0 \leq id < |escrutinio| - 1) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j + 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id])))$

$\equiv (0 \leq id < |escrutinio| - 1 \wedge 0 \leq j < |escrutinio| - 1) \wedge_L$
$((escrutinio[j] > escrutinio[id]) \wedge 1 \leq j + 1 \leq |escrutinio| - 1 \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j + 1 \longrightarrow_L escrutinio[k] \leq escrutinio[j]))$
$\vee$
$((escrutinio[j] \leq escrutinio[id] \wedge 0 \leq id < |escrutinio| - 1 \wedge 1 \leq j + 1 \leq |escrutinio| - 1 \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j + 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id])))$

$\equiv (0 \leq id < |escrutinio| - 1 \wedge 0 \leq j < |escrutinio| - 1) \wedge_L$
$((escrutinio[j] > escrutinio[id]) \wedge (\forall k : \mathbb{Z})(0 \leq k < j + 1 \longrightarrow_L escrutinio[k] \leq escrutinio[j]))$
$\vee$
$((escrutinio[j] \leq escrutinio[id]) \wedge (\forall k : \mathbb{Z})(0 \leq k < j + 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id]))$

lo cual es esperable ya que si vale que escrutinio[j] es mayor que escrutinio[id], me va a guardar que escrutinio[k] es menor o igual a escrutinio[j] y entonces tenemos que escrutinio[j] es el máximo en ese momento de la ejecución del programa; caso contrario, mantenemos la guarda anterior que respondería que escrutinio[id] es el máximo actual.

Demostracion 3
(I $\wedge \neg B) \implies Q_c$
veamos qué pasa en (I $\wedge \neg B$)
(I $\wedge \neg B) \equiv (1 \leq j \leq |escrutinio| - 1) \wedge (0 \leq id < |escrutinio| - 1) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j \longrightarrow_L escrutinio[k] \leq escrutinio[id]) \wedge (j \geq |escrutinio| - 1)$
$(I \wedge \neg B) \equiv (j = |escrutinio| - 1) \wedge (0 \leq id < |escrutinio| - 1) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id])$

$(I \wedge \neg B) \equiv (j = |escrutinio| - 1) \wedge (0 \leq id < |escrutinio| - 1) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j) \longrightarrow_L escrutinio[k] \leq escrutinio[id])$

$((j = |escrutinio| - 1) \wedge (0 \leq id < |escrutinio| - 1) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1) \longrightarrow_L escrutinio[k] \leq escrutinio[id]) \implies (j = |escrutinio| - 1 \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1) \longrightarrow_L escrutinio[k] \leq escrutinio[id])$

Demostracion 4
$(I \wedge B \wedge f_v = v_0) \implies wp(S, f_v < v_0)$

proponemos
$f_v = |escrutinio| - 1 - j$

$veamos wp(\textsf{if } (escrutinio[j] > escrutinio[id]) \textsf{ then } id := j \textsf{ else } skip \textsf{ fi}, wp(S_2, f_v < v_0))$

$\equiv def(H) \wedge_L ((H) \wedge wp(S_0, wp(S_2, f_v < v_0))) \vee ((\neg H) \wedge wp(S_1, wp(S_2, f_v < v_0))) \equiv def(H) \wedge_L ((H) \wedge wp(S_0, wp(S_1, f_v < v_0))) \vee ((\neg H) \wedge wp(S_2, f_v < v_0))$

tenemos que ver por partes el desarrollo de la weakest-preconditions para lograr mayor claridad en la demostración

1. wp ($S_2, f_v < v_0) \equiv wp(j := j+1, f_v < v_0) \equiv |escrutinio| - 1 - (j+1) < v_0 \equiv |escrutinio| - 2 - j < v_0$

$wp(S_0, wp(S_2, f_v < v_0)) \equiv wp(S_0, |escrutinio|-2-j < v_0) \equiv wp(id_1 := j, |escrutinio|-2-j < v_0) \equiv |escrutinio|-2-j < v_0$

ahora estamos en buen momento de ir reemplazando

$\equiv def(H) \wedge_L ((H \wedge |escrutinio| - 2 - j < v_0) \vee ((\neg H) \wedge |escrutinio| - 2 - j < v_0))$

$\equiv (0 \leq id < |escrutinio| - 1) \wedge (0 \leq j < |escrutinio| - 1) \wedge_L ((escrutinio[j] > escrutinio[id]) \wedge |escrutinio| - 2 - j < v_0)$
$\vee ((escrutinio[j] \leq escrutinio[id]) \wedge |escrutinio| - 2 - j < v_0))$

dado que (p $\wedge q) \vee (\neg p \wedge q) \equiv q$

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

procedemos al final de la demostración
$v_0 = |escrutinio| - 1 - j \implies |escrutinio| - 2 - j < v_0 \equiv -1 < 0 \equiv True$

Demostracion 5
$I \wedge f_v \leq 0 \implies \neg B$

I $\wedge f_v \leq 0 \implies j \geq |escrutinio| - 1$

$(0 \leq id < |escrutinio| - 1 \wedge (1 \leq j \leq |escrutinio| - 1)) \wedge |escrutinio| - 1 - j \leq 0) \implies j \geq |escrutinio| - 1$

$|escrutinio| - 1 \leq j \implies j \geq |escrutinio| - 1$

3.2.6. Q indiceMaximo $\implies wp(asignación, Pre_{IndiceSegundoMaximo})$

- $Q_c(indiceMaximo) \equiv \{j = |escrutinio| - 1 \wedge 0 \leq id_1 < |escrutinio| - 1 \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id])$

- $P_c(indiceSegundoMaximo) \equiv \{id_2 = 0 \wedge j = 1 \wedge escrutinio[id_1] := 0\}$
  vampos a observar

wp($id_2 := 0, wp(escrutinio[id_1] := 0, wp(j := 1, Pc(indiceSegundoMaximo)))$)
$\equiv 0 \leq id_1 < |escrutinio|$
$\equiv 0 \leq id_1 < |escrutinio| \implies 0 \leq id_1 < |escrutinio|$

3.2.7. Q Maximo $\implies POST$

En este segmento vamos a demostrar la implicancia logica entre nuestro bloque de código Maximo que se reutilizará para hallar tanto el máximo de la secuencia escrutinio como de una lista $escrutinio_0$ que se sentenciará como una metavariable en el desarrollo de la implicación.
La idea es reasignar en escrutinio[id_1] el 0 para reutilizarla en la búsqueda del id_2

{ j = $\|escrutinio\| - 1 \wedge\ 0 \leq id_1 < |escrutinio| - 1 \wedge (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id_1])\}$
Vemos que si realizamos la sustitución del elemento con el uso de la indexación, reconstruimos nuestra lista escrutinio
$escrutinio := escrutinio_0$
$escrutinio_0[id_1] := 0$
$id_2 := 0$
$m := 1$

Vamos a trabajar con esta lista y volver a hallarle el máximo.
Por tanto, la postcondición de Máximo nos lleva a definir
$m := |escrutinio_0| - 1 \wedge 0 \leq id_2 < |escrutinio_0| - 1 \wedge (\forall m : \mathbb{Z})(0 \leq m < |escrutinio_0| - 1 \longrightarrow_L escrutnio_0[m] \leq escrutinio_0[id_2])$

Tenemos entonces nuestras dos variables con sus valores esperados. El máximo supremo de escrutinio y su consecuente 2do máximo que remite al máximo de escrutinio$_0$

$(0 \leq id_1 < |escrutinio| - 1) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < |escrutinio| - 1 \longrightarrow_L escrutinio[k] \leq escrutinio[id_1]) \wedge escrutinio = escrutinio_0 \wedge escrutinio_0[id_1] = 0 \wedge 0 \leq id_2 < |escrutinio_0| - 1 \wedge (\forall m : \mathbb{Z})(0 \leq m < |escrutinio_0| - 1 \longrightarrow_L escrutinio_0[m] \leq escrutinio_0[id_2])$

Queremos ver que todo esto implique la POST de nuestro procedimiento

{ POST } $\equiv (\forall l : \mathbb{Z})(0 \leq l < |escrutinio| - 1) \wedge l \neq res_0 \wedge l \neq res_1 \longrightarrow_L (escrutinio[l] < escrutinio[res_0] \wedge escrutinio[l] < escrutinio[res_1]) \wedge escrutinio[res_0] > escrutinio[res_1]$
$\{Q_{indiceMax1} \wedge Asignaciones \wedge Q_{indiceMax2}\} \implies \{POST\}$

Conclusiones:

- $(\forall k, m : \mathbb{Z})(0 \leq k, m < |escrutinio| - 1) \wedge k \neq res_0 \wedge m \neq res_1 \longrightarrow_L (escrutinio[k] < escrutinio[res_0] \wedge escrutinio[m] < escrutinio[res_1]) \wedge escrutinio[res_0] > escrutinio[res_1]$

4. Funciones auxiliares/predicados

`pred diferenciaMenorA10` (in escrutinio : $\mathbb{Z}$, in indice: $\mathbb{Z}$) {
$(\forall i : \mathbb{Z})((0 \leq i < |escrutinio| - 1 \land i \neq indice) \longrightarrow_L (escrutinio[indice] - escrutinio[i] < total(escrutinio) * 0,1))$
}
`pred esMaximo` (in indice: $\mathbb{Z}$, in escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$(\forall i : \mathbb{Z})((0 \leq i < |escrutinio| - 1 \land i \neq indice) \longrightarrow_L (escrutinio[indice] > escrutinio[i])$
}
`aux total` (in escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{i=0}^{|escrutinio|-1} escrutinio[i]$ **;**
`pred sonDistintos` (in escrutinio : $\mathbb{Z}$) {
$(\forall j : \mathbb{Z})(\forall i : \mathbb{Z})((0 \leq j < |escrutinio| - 1 \land 0 \leq i < |escrutinio| - 1 \land i \neq j) \longrightarrow_L escrutinio[j] \neq escrutinio[i])$
}
`aux indiceMaximo` (in escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{i=0}^{|escrutinio|-2}$ if $esMaximo(i, escrutinio)$ then i else 0 fi **;**
`aux bancasAsignadas` (in indice: $\mathbb{Z}$, in escrutinio: $seq\langle\mathbb{Z}\rangle$, in bancaEnDisputa: $\mathbb{Z}$, in votosTotal:$\mathbb{Z}$, in cant_bancas) : $\mathbb{Z}$ =
if$(bancaEnDisputa > cant\_bancas)$**then**$(0)$**else**
If$(esMaximo(indice, escrutinio) \land_L (escrutinio[indice]/votosTotal > 0,3))$ **;**
then (1 + bancasAsignadas(indice, setAt(escrutinio, indice , escrutinio[indice]/(bancaEnDisputa +1)) , bancaEnDisputa + 1))
else bancasAsignadas(indice, setAt(escrutinio, escrutinio[indiceMaximo(escrutinio)], escrutinio[indiceMaximo(escrutinio)]/(bancaEnDisputa + 1)), bancaEnDisputa + 1)

`pred coincidiceDHontEscrutinio` (in D´Hont: $seq\langle\mathbb{Z}\rangle$, in escrutinio: $seq\langle\mathbb{Z}\rangle$) {
$|Dhont| = |escrutinio| - 1) \land ((\forall i : \mathbb{Z})(0 \leq i < |escrutinio| - 1) \longrightarrow_L$
$((\forall j : \mathbb{Z})(0 \leq j < |DHont|) \longrightarrow_L DHont[i][j] = divisón(escrutinio[i], j + 1)))$
}
`aux #d´HontCumplenUmbral` (in D´Hont: $seq\langle\mathbb{Z}\rangle$, in total: $\mathbb{Z}$) : $\mathbb{Z}$ $= \sum_{i=0}^{|dHont|} \sum_{j=0}^{|dHont[i]|}$ if $(DHont[i][j]/total > 0,3)$ then 1 else 0 fi **;**

`proc partidoMásVotado` (in escrutinio : $seq\langle\mathbb{Z}\rangle$) : $\mathbb{Z}$
`requiere` $\{|escrutinio| \geq 3\}$
`asegura` $\{(\forall i : \mathbb{Z})$ (
$0 \leq i < |escrutinio|$
)}

`aux #hombres` (in lista: $seq\langle\mathbb{Z} \times \mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{j=0}^{|lista|-1}$ if $lista[j]_1 == 1$ then 1 else 0 fi **;**
`aux #mujeres` (in lista: $seq\langle\mathbb{Z} \times \mathbb{Z}\rangle$) : $\mathbb{Z}$ $= \sum_{j=0}^{|lista|-1}$ if $lista[j]_1 == 2$ then 1 else 0 fi **;**