



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Practico 1

Especificaciones, Implementaciones y Demostraciones

3 de noviembre de 2023

Algoritmos y Estructuras de Datos

## Grupo BP++facts

Integrante	LU	Correo electrónico
Berton, Juan Cruz	835/22	juanceberton@gmail.com
Bakal, Ariel	1024/22	bakalariel2002@gmail.com
Gilardon, Bautista	742/21	bautistagilardon@gmail.com
Pomi, Franco	935/22	pomifranco@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>

# Preliminares

Algunos predicados preliminares que seran utilizados en muchos procesos:

```
pred sinRepetidos (lista: seq⟨ℤ⟩) {  
  (∀i, j : ℤ)(0 ≤ i, j < |lista| ∧ i ≠ j →L lista[i] ≠ lista[j])  
}
```

```
pred votosValidos (votos: seq⟨ℤ⟩) {  
  (∀i : ℤ)(0 ≤ i < |votos| → votos[i] ≥ 0)  
}
```

```
pred matrizNoVacía (matriz: seq⟨seq⟨ℤ⟩⟩) {  
  (|matriz| > 0) ∧ (∀k : ℤ)(0 ≤ k < |matriz| →L |matriz[k]| > 0)  
}
```

# 1. Ejercicio 1: hayBallotage

hayBallotage: verifica si hay ballotage en la elección presidencial.

proc HayBallotage ( in escrutinio:  $seq(\mathbb{Z})$  ) : Bool

Donde:

- escrutinio: es la cantidad de votos de cada partido a nivel nacional para la elección presidencial.
- devuelve verdadero sii hay ballotage en la elección presidencial.

## 1.1. Especificación

proc HayBallotage ( in escrutinio:  $seq(\mathbb{Z})$  ) : Bool

requiere  $\{|escrutinio| \geq 3 \wedge \text{sinRepetidos}(escrutinio) \wedge \text{votosValidos}(escrutinio)\}$

asegura  $\{res = False \iff ((\exists j : \mathbb{Z})(0 \leq j < |escrutinio| \longrightarrow_L escrutinio[j] > 0,45 * totalVotos)) \vee ((\forall i : \mathbb{Z})(\exists k : \mathbb{Z})(0 \leq i < |escrutinio| \wedge_L 0 \leq k < |escrutinio| \wedge i \neq k \longrightarrow_L (escrutinio[k] > 0,4 * totalVotos \wedge escrutinio[i] < escrutinio[k] - 0,1 * totalVotos)))\}$

aux totalVotos(escrutinio:  $seq(\mathbb{Z})$ ):  $\mathbb{Z} = \sum_{k=0}^{|escrutinio|-1} escrutinio[k]$

## 1.2. Implementación

```
1  res := True;
2  i:= 0;
3  j:= 0;
4  mas_votos:= 0;
5  2do_mas_votos:= 0;
6  total_votos:= 0;
7  10%_votos:= 0;
8  40%_votos:= 0;
9  45%_votos:= 0;
10
11 while (j<escrutinio.size()) do
12     total_votos:= escrutinio[j] + total_votos;
13     j:= j + 1;
14
15 endwhile
16
17 10%_votos:= total_votos * 0.1;
18 40%_votos:= total_votos * 0.4;
19 45%_votos:= total_votos * 0.45;
20
21 while (i<escrutinio.size()) do
22     if (escrutinio[i] > 45%_votos) then
23         res:= False;
24     else
25         if (escrutinio[i] > mas_votos) then
26             2do_mas_votos:= mas_votos;
27             mas_votos:= escrutinio[i];
28         else
29             if ( escrutinio[i] < mas_votos && escrutinio[i]>2do_mas_votos) then
30                 2do_mas_votos:= escrutinio[i];
31
32             else
33                 skip
34
35             endif
36         endif
37     endif
38     i:= i + 1;
39
40 endwhile
41
```

```

42 | if (mas_votos > 40%_votos) then
43 |     if (mas_votos - 2do_mas_votos > 10%_votos) then
44 |         res:= False;
45 |     else
46 |         skip
47 |     endif
48 | else
49 |     skip
50 | endif

```

Código 1: Codigo de Ej1

## 2. Ejercicio 2: hayFraude

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

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

Donde:

- los tres escrutinios son a nivel nacional.
- devuelve verdadero sii hay al menos dos escrutinios con diferente cantidad de votos

### 2.1. Especificación

```
proc hayFraude ( in escrutinio_presidencial:  $seq\langle\mathbb{Z}\rangle$  , in escrutinio_senadores:  $seq\langle\mathbb{Z}\rangle$ , in escrutinio_diputados:  $seq\langle\mathbb{Z}\rangle$  ) : Bool
  requiere {votosValidos(escrutinio_presidencial)  $\wedge$  votosValidos(escrutinio_senadores)  $\wedge$ 
  votosValidos(escrutinio_diputados)}
  requiere {sinRepetidos(escrutinio_presidencial)  $\wedge$  sinRepetidos(escrutinio_senadores)  $\wedge$ 
  sinRepetidos(escrutinio_diputados)}
  asegura {res = true  $\iff$ 
  ( $\sum_{i=0}^{|\text{escrutinio\_presidencial}|-1} \text{escrutinio\_presidencial}[i] \neq \sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i]$ )  $\vee$ 
  ( $\sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i] \neq \sum_{i=0}^{|\text{escrutinio\_diputados}|-1} \text{escrutinio\_diputados}[i]$ )}
```

Observación: No compare el largo de las listas: escrutinio\_presidencial, escrutinio\_senadores y escrutinio\_diputados, ya que, existe la posibilidad de que un partido presente lista presidencial, pero no para senadores. Por ejemplo, podría haber 10 partidos presentándose para presidente, pero solo 7 presentan diputados. En este caso, los escrutinios serian validos, pero el largo de las listas no seria el mismo.

### 2.2. Implementación

```
1 sumaEP := 0;
2 a := 0;
3 while ( a < escrutinio_presidencial.size() ) do
4   sumaEP := sumaEP + escrutinio_presidencial[a];
5   a := a + 1;
6 endwhile
7
8 sumaES := 0;
9 b := 0;
10 while ( b < escrutinio_senadores.size() ) do
11   sumaES := sumaES + escrutinio_senadores[b];
12   b := b + 1;
13 endwhile
14
15 sumaED := 0;
16 c := 0;
17 while ( c < escrutinio_diputados.size() ) do
18   sumaED := sumaED + escrutinio_diputados[c];
19   c := c + 1;
20 endwhile
21
22 res := ( (sumaEP  $\neq$  sumaES)  $\vee$  (sumaES  $\neq$  sumaED) )
```

Código 2: Codigo de Ej2

### 2.3. Demostración de Correctitud

Paso 1:

Elección de  $P_c, Q_c, B, I, fv$  para cada uno de los 3 ciclos. Como los 3 ciclos son similares, sus Invariantes también, por lo tanto trataremos el caso de un ciclo representativo y luego analizaremos el caso particular de cada uno. Llamemos a este programa  $S_C$

```

1 suma := 0;
2 i := 0;
3 while ( i < S.size() ) do
4     suma := suma + S[i];
5     i := i + 1;
6 endwhile

```

Luego, nos queda que los componentes del Invariante serian:

- $P_c \equiv \text{suma} = 0 \wedge i = 0$
- $Q_c \equiv \text{suma} = \sum_{k=0}^{|S|-1} S[k]$
- $B \equiv i < |S|$
- $I \equiv (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k])$
- $fv \equiv |S| - i$

Analicemos si el Invariante cumple con las pruebas de correctitud.

1.  $P_c \longrightarrow I$

$$P_c \equiv \text{suma} = 0 \wedge i = 0$$

$$I \equiv 0 \leq i \leq |S| \wedge \text{suma} = \sum_{k=0}^{i-1} S[k]$$

$$P_c \longrightarrow I \equiv ((\text{suma} = 0) \wedge (i = 0)) \longrightarrow ((0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]))$$

$$P_c \longrightarrow I \equiv ((0 \leq 0 \leq |S|) \wedge (0 = \sum_{k=0}^{0-1} S[k]))$$

$$P_c \longrightarrow I \equiv ((True) \wedge (0 = \sum_{k=0}^{-1} S[k]))$$

$$P_c \longrightarrow I \equiv ((True) \wedge (0 = 0))$$

$$P_c \longrightarrow I \equiv ((True) \wedge (True)) \equiv True$$

2.  $I \wedge \neg B \longrightarrow Q_c$   $I \equiv 0 \leq i \leq |S| \wedge \text{suma} = \sum_{k=0}^{i-1} S[k]$

$$B \equiv i < |S|$$

$$Q_c \equiv \text{suma} = \sum_{k=0}^{|S|-1} S[k]$$

$$I \wedge \neg B \longrightarrow Q_c \equiv (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \wedge \neg(i < |S|) \longrightarrow \text{suma} = \sum_{k=0}^{|S|-1} S[k]$$

$$I \wedge \neg B \longrightarrow Q_c \equiv (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \wedge (i \geq |S|) \longrightarrow \text{suma} = \sum_{k=0}^{|S|-1} S[k]$$

$$I \wedge \neg B \longrightarrow Q_c \equiv (0 \leq i \leq |S|) \wedge (i \geq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \longrightarrow \text{suma} = \sum_{k=0}^{|S|-1} S[k]$$

$$I \wedge \neg B \longrightarrow Q_c \equiv (i = |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \longrightarrow \text{suma} = \sum_{k=0}^{|S|-1} S[k]$$

$$I \wedge \neg B \longrightarrow Q_c \equiv \text{suma} = \sum_{k=0}^{|S|-1} S[k] \longrightarrow \text{suma} = \sum_{k=0}^{|S|-1} S[k]$$

$$I \wedge \neg B \longrightarrow Q_c \equiv True$$

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

$$I \wedge fv \leq 0 \equiv (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \wedge |S| - i \leq 0 \equiv (i = |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k])$$

$$I \wedge fv \leq 0 \longrightarrow \neg B \equiv (i = |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \longrightarrow (i \geq |S|)$$

$$I \wedge fv \leq 0 \longrightarrow \neg B \equiv (\text{suma} = \sum_{k=0}^{i-1} S[k]) \longrightarrow (|S| \geq |S|) \equiv True$$

4.  $\{I \wedge B\} S_C \{I\}$   
 Quiero ver que  $I \wedge B \longrightarrow wp(S_C, I)$

4.a  $wp(S_C, I)$   
 $wp(S_C, I) \equiv wp(\text{suma} := \text{suma} + S[i]; i := i + 1; (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]))$   
 $wp(S_C, I) \equiv wp(\text{suma} := \text{suma} + S[i], wp(i := i + 1, (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k])))$   
 $wp(S_C, I) \equiv wp(\text{suma} := \text{suma} + S[i], wp(i := i + 1, (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k])))$

Veamos  $wp(i := i + 1, (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]))$   
 $\equiv def(i + 1) \wedge (0 \leq i + 1 \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i+1-1} S[k])$   
 $\equiv (0 \leq i \leq |S| - 1) \wedge (\text{suma} = \sum_{k=0}^i S[k])$   
 $\equiv (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^i S[k]) \equiv E_1$

Ahora veamos  $wp(\text{suma} := \text{suma} + S[i], E_1)$   
 $\equiv def(\text{suma} + S[i]) \wedge (0 \leq i < |S|) \wedge (\text{suma} + S[i] = \sum_{k=0}^i S[k])$   
 $\equiv (0 \leq i < |S|) \wedge (0 \leq i < |S|) \wedge (\text{suma} + S[i] = \sum_{k=0}^{i-1} S[k] + S[i])$   
 $\equiv (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k])$   
 Entonces  $wp(S, I) \equiv (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k])$ .

4.b  
 $I \wedge B \longrightarrow wp(S_C, I)$   
 $\equiv (0 \leq i \leq |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \wedge (i < |S|) \longrightarrow (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k])$   
 $\equiv (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \longrightarrow (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \equiv True$

5.  $\{I \wedge B \wedge v_0 = |S| - i\} S_C \{|S| - i < v_0\}$  Veamos la  $wp(S_C, |S| - i < v_0)$

5.a  $wp(S_C, |S| - i < v_0)$   
 $wp(S_C, I) \equiv wp(\text{suma} := \text{suma} + S[i]; i := i + 1; |S| - i < v_0)$   
 $wp(S_C, I) \equiv wp(\text{suma} := \text{suma} + S[i], wp(i := i + 1, |S| - i < v_0))$   
 $wp(S_C, I) \equiv wp(\text{suma} := \text{suma} + S[i], wp(i := i + 1, |S| - i < v_0))$

Veamos  $wp(i := i + 1, |S| - i < v_0)$   
 $\equiv def(i + 1) \wedge |S| - i - 1 < v_0$   
 $\equiv |S| - i < v_0 + 1$   
 Ahora veamos  $wp(\text{suma} := \text{suma} + S[i], |S| - i < v_0 + 1)$   
 $\equiv def(\text{suma} + S[i]) \wedge |S| - i < v_0 + 1$   
 $\equiv (0 \leq i \leq |S|) \wedge (|S| - i < v_0 + 1)$

Entonces  $wp(S_C, |S| - i < v_0) \equiv (0 \leq i < |S|) \wedge (|S| - i < v_0 + 1)$

Veamos ahora que  $(I \wedge B \wedge v_0 = |S| - i) \longrightarrow (0 \leq i < |S|) \wedge (|S| - i < v_0 + 1)$   
 $(I \wedge B \wedge v_0 = |S| - i) \equiv (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \wedge v_0 = |S| - i$   
 $(0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \wedge (v_0 = |S| - i) \longrightarrow (0 \leq i < |S|) \wedge (|S| - i < v_0 + 1)$   
 $\equiv (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \longrightarrow (0 \leq i < |S|) \wedge (|S| - i < |S| - i + 1)$   
 $\equiv (0 \leq i < |S|) \wedge (\text{suma} = \sum_{k=0}^{i-1} S[k]) \longrightarrow (0 \leq i < |S|) \wedge (|S| < |S| + 1) \equiv True$

Habiendo demostrado cada uno de los pasos, queda demostrado que la tripla de Hoare:

$$\{True\} S_C \{ \text{suma} = \sum_{k=0}^{|S|-1} S[k] \}. \quad (1)$$

Teniendo esto en mente si vemos el código escrito (Codigo 2) en SmallLang, podemos ver que si remplazamos  $i$  por  $a, b$  y  $c$  y  $suma$  por  $sumaEP, sumaES$  y  $sumaED$  en cada uno de los tres ciclos, nos queda que las tres triplas de Hoare son validad, por tanto nos falta demostrar que la tripla:

$$\begin{aligned} & \{(sumaEP = \sum_{i=0}^{|\text{escrutinio\_presidente}|-1} \text{escrutinio\_presidente}[i]) \wedge \\ & (sumaES = \sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i]) \wedge \\ & (sumaED = \sum_{i=0}^{|\text{escrutinio\_diputados}|-1} \text{escrutinio\_diputados}[i]) \equiv \text{Pre}\} \\ \text{res} := & ((sumaEP \neq sumaES) \vee (sumaES \neq sumaED)) \\ \{res = \text{true} \iff & (\sum_{i=0}^{|\text{escrutinio\_presidencial}|-1} \text{escrutinio\_presidencial}[i] \neq \sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i]) \\ \vee (\sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i] \neq & \sum_{i=0}^{|\text{escrutinio\_diputados}|-1} \text{escrutinio\_diputados}[i]) \equiv \text{Post}\} \end{aligned}$$

$$\begin{aligned} & \text{veamos el } wp(\text{res} := ((sumaEP \neq sumaES) \vee (sumaES \neq sumaED)), \text{Post}) \\ \equiv & \text{def}(sumaEP) \wedge \text{def}(sumaES) \wedge \text{def}(sumaED) \wedge ((sumaEP \neq sumaES) \vee (sumaES \neq sumaED)) = \text{true} \iff \\ & (\sum_{i=0}^{|\text{escrutinio\_presidencial}|-1} \text{escrutinio\_presidencial}[i] \neq \sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i]) \\ & \vee (\sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i] \neq \sum_{i=0}^{|\text{escrutinio\_diputados}|-1} \text{escrutinio\_diputados}[i]) \\ \equiv & ((sumaEP \neq sumaES) \vee (sumaES \neq sumaED)) = \text{true} \iff \\ & (\sum_{i=0}^{|\text{escrutinio\_presidencial}|-1} \text{escrutinio\_presidencial}[i] \neq \sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i]) \\ & \vee (\sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i] \neq \sum_{i=0}^{|\text{escrutinio\_diputados}|-1} \text{escrutinio\_diputados}[i]) \end{aligned}$$

Podemos ver que esta expresión es tautológica en el caso en que:

- $sumaEP = \sum_{i=0}^{|\text{escrutinio\_presidente}|-1} \text{escrutinio\_presidente}[i]$
- $sumaES = \sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i]$
- $sumaED = \sum_{i=0}^{|\text{escrutinio\_diputados}|-1} \text{escrutinio\_diputados}[i]$

Por tanto llegamos a que:  $wp(\text{res} := ((sumaEP \neq sumaES) \vee (sumaES \neq sumaED)), \text{Post}) \equiv$

$$\begin{aligned} & \{(sumaEP = \sum_{i=0}^{|\text{escrutinio\_presidente}|-1} \text{escrutinio\_presidente}[i]) \wedge \\ & (sumaES = \sum_{i=0}^{|\text{escrutinio\_senadores}|-1} \text{escrutinio\_senadores}[i]) \wedge \\ & (sumaED = \sum_{i=0}^{|\text{escrutinio\_diputados}|-1} \text{escrutinio\_diputados}[i])\} \end{aligned}$$

Que es exactamente nuestra precondition, por lo tanto queda demostrado que el código (Codigo 2) es correcto.



### 3. Ejercicio 3: obtenerSenadoresEnProvincia

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

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

Donde:

- escrutinio: es la cantidad de votos de cada partido en la provincia.
- devuelve una tupla que contiene el id de los dos partidos con mayor cantidad de votos.

#### 3.1. Especificación

```
proc obtenerSenadoresEnProvincia ( in escrutinio: seq( $\mathbb{Z}$ )) :  $\mathbb{Z} \times \mathbb{Z}$ 
  requiere { |escrutinio|  $\geq 2 \wedge$  votosValidos(escrutinio)  $\wedge$  sinRepetidos(escrutinio) }
  asegura { (res = (res0, res1)  $\leftrightarrow$  ( $\exists$  res0 :  $\mathbb{Z}$ ) ( $\exists$  res1 :  $\mathbb{Z}$ ) ( $\forall x$  :  $\mathbb{Z}$ ) ( $x \neq$  res0  $\wedge$   $x \neq$  res1)  $\wedge_L$  ( $0 \leq$  res0, res1,  $x \leq$  |escrutinio|)  $\rightarrow_L$  (escrutinio[x] < escrutinio[res1])  $\wedge$  (escrutinio[res1] < escrutinio[res0])) }
```

#### 3.2. Implementacion

```
1 pos := 0;
2 res_0 := 0;
3 res_1 := 1;
4
5 if (escrutinio [res_0] < escrutinio [res_1]) do
6   res_0 := 1;
7   res_1 := 0;
8 else
9   skip
10 endif
11
12 while pos < escrutinio.size() do
13   if escrutinio [pos] > escrutinio [res_0] then
14     res_0 := posicion;
15   else
16     skip;
17   endif
18   pos := pos + 1;
19 endwhile
20
21 pos := 0;
22
23 while pos < escrutinio.size() do
24   if escrutinio [pos] > escrutinio [res_1] && escrutinio [res_1] != escrutinio [res_0] then
25     res_1 := posicion;
26   else
27     skip;
28   endif
29   pos := pos + 1;
30 endwhile
31
32 res := (res_0, res_1)
```

Código 3: Codigo de Ej3

### 3.3. Demostración de Correctitud

Elección de  $P_c, Q_c, B, I, fv$  para ambos ciclos. Como los ambos ciclos son similares, sus Invariantes también, por lo tanto trataremos el caso de un ciclo representativo y luego analizaremos el caso particular de cada uno. Llamemos a este programa  $S_C$

```

1 | pos := 0
2 |
3 | while pos < escrutinio.size() do
4 |   if escrutinio [pos] > escrutinio [res] then
5 |     res := pos;
6 |   else
7 |     skip;
8 |   endif
9 |
10 | pos := pos + 1;
11 | endwhile

```

Los componentes del Invariante, para este caso, serian:

- $P_c \equiv (|escrutinio| \geq 2) \wedge (res = 0 \vee res = 1) \wedge (pos = 0)$
- $Q_c \equiv (|escrutinio| \geq 2) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$
- $B \equiv pos < |escrutinio|$
- $I \equiv (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$
- $fv \equiv |escrutinio| - pos$

Analicemos si el Invariante cumple con las pruebas de correctitud.

#### 1. $P_c \rightarrow I$

$$P_c \equiv (|escrutinio| \geq 2) \wedge (res = 0 \vee res = 1) \wedge (pos = 0)$$

$$I \equiv (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$$

$$P_c \rightarrow I \equiv (|escrutinio| \geq 2) \wedge (res = 0 \vee res = 1) \wedge (pos = 0) \rightarrow (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$$

$$P_c \rightarrow I \equiv (|escrutinio| \geq 2) \wedge (pos = 0) \rightarrow (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq 0 < |escrutinio| \vee 0 \leq 1 < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$$

$$P_c \rightarrow I \equiv (|escrutinio| \geq 2) \rightarrow (|escrutinio| \geq 2) \wedge (0 \leq 0 \leq |escrutinio|) \wedge (0 \leq 0 < |escrutinio| \vee 0 \leq 1 < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < 0) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$$

$$P_c \rightarrow I \equiv (\mathbf{True}) \wedge (\mathbf{True}) \wedge (\mathbf{True} \vee \mathbf{True}) \wedge (\forall X : \mathbb{Z})(\mathbf{False}) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$$

$$P_c \rightarrow I \equiv (\mathbf{True}) \wedge (\mathbf{True}) \wedge (\mathbf{True}) \wedge (\mathbf{True}) = \mathbf{True}$$

#### 2. $I \wedge \neg B \rightarrow Q_c$

$$I \equiv (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$$

$$B \equiv pos < |escrutinio|$$

$$Q_c \equiv (|escrutinio| \geq 2) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] < escrutinio[res])$$

$$I \wedge \neg B \rightarrow Q_c \equiv (((|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])) \wedge (pos \geq |escrutinio|)) \rightarrow Q_c$$

$$I \wedge \neg B \rightarrow Q_c \equiv \text{def}(pos = |escrutinio|) \wedge_L (((|escrutinio| \geq 2) \wedge (0 \leq |escrutinio| \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] \leq escrutinio[res])) \wedge (|escrutinio| \geq |escrutinio|)) \rightarrow Q_c$$

$$I \wedge \neg B \rightarrow Q_c \equiv (((|escrutinio| \geq 2) \wedge (\mathbf{True}) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] \leq escrutinio[res])) \wedge (\mathbf{True})) \rightarrow Q_c$$

$$I \wedge \neg B \rightarrow Q_c \equiv ((0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] \leq escrutinio[res])) \rightarrow Q_c$$

$$I \wedge \neg B \rightarrow Q_c \equiv Q_c \rightarrow Q_c = \mathbf{True}$$

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

$I \equiv (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$

$fv \equiv |escrutinio| - pos$

$B \equiv pos < |escrutinio|$

$(I \wedge fv \leq 0) \longrightarrow \neg B \equiv (I \wedge (|escrutinio| - pos) \leq 0) \longrightarrow \neg B$

$(I \wedge fv \leq 0) \longrightarrow \neg B \equiv (I \wedge |escrutinio| \leq pos) \longrightarrow \neg B$

$(I \wedge fv \leq 0) \longrightarrow \neg B \equiv (I \wedge \neg B) \longrightarrow \neg B \equiv \mathbf{True}$

4.  $\{I \wedge B\}S_C\{I\}$

$I \equiv (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$

$B \equiv pos < |escrutinio|$

Quiero ver que  $I \wedge B \longrightarrow wp(S_C, I)$

4.a  $I \wedge B$

$I \wedge B$

$\equiv (((|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])) \wedge (pos < |escrutinio|))$

$\equiv \text{def}(pos < |escrutinio|) \wedge_L (((|escrutinio| \geq 2) \wedge (0 \leq pos < |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] \leq escrutinio[res])) \wedge (\mathbf{True}))$

$\equiv (|escrutinio| \geq 2) \wedge (0 \leq pos < |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$

4.b  $wp(S_C, I)$

$wp(S_C, I) \equiv wp(\text{if}(escrutinio[pos] > escrutinio[res])\text{then}(res = pos)\text{else}(\text{skip})\text{endif}; (pos = pos + 1), I)$

$wp(S_C, I) \equiv wp(\text{if}(escrutinio[pos] > escrutinio[res])\text{then}(res = pos)\text{else}(\text{skip})\text{endif}, wp(pos = pos + 1, I))$

En un principio analizo:

$wp(pos = pos + 1, I) \equiv (pos = pos + 1) \wedge (|escrutinio| \geq 2) \wedge (0 \leq pos \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$

$wp(pos = pos + 1, I) \equiv \text{def}(pos = pos + 1) \{ (|escrutinio| \geq 2) \wedge (0 \leq pos + 1 \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos + 1) \rightarrow_L (escrutinio[X] \leq escrutinio[res]) \}$

$wp(pos = pos + 1, I) \equiv \{ (|escrutinio| \geq 2) \wedge (0 \leq pos + 1 \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos + 1) \rightarrow_L (escrutinio[X] \leq escrutinio[res]) \}$

$wp(pos = pos + 1, I) \equiv E1$

Al tratarse de un IF, se divide la implicancia para ambos casos

$((I \wedge B) \wedge (escrutinio[pos] > escrutinio[res])) \longrightarrow wp(res := pos, E1)$

$((I \wedge B) \wedge (escrutinio[pos] > escrutinio[res])) \longrightarrow \text{def}(res) \wedge_L (|escrutinio| \geq 2) \wedge (0 \leq pos + 1 \leq |escrutinio|) \wedge (0 \leq pos < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos + 1) \rightarrow_L (escrutinio[X] \leq escrutinio[pos])$

$((I \wedge B) \wedge (escrutinio[pos] > escrutinio[res])) \longrightarrow \text{def}(res = pos) \wedge_L (|escrutinio| \geq 2) \wedge (0 \leq pos + 1 \leq |escrutinio|) \wedge (0 \leq pos < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos + 1) \rightarrow_L (escrutinio[X] \leq escrutinio[pos])$

$((I \wedge B) \wedge (escrutinio[pos] > escrutinio[res])) \longrightarrow \text{def}(res = pos) \wedge_L (|escrutinio| \geq 2) \wedge (0 \leq pos < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos + 1) \rightarrow_L (escrutinio[X] \leq escrutinio[pos])$

$((|escrutinio| \geq 2) \wedge (0 \leq pos < |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio|) \rightarrow_L (escrutinio[X] \leq escrutinio[res])) \wedge (escrutinio[pos] > escrutinio[res]) \longrightarrow \text{def}(res = pos) \wedge_L (|escrutinio| \geq 2) \wedge (0 \leq pos < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos + 1) \rightarrow_L (escrutinio[X] \leq escrutinio[pos]) \equiv \mathbf{True}$

$((I \wedge B) \wedge (escrutinio[pos] \leq escrutinio[res])) \longrightarrow wp(\text{skip}, E1)$

$((I \wedge B) \wedge (escrutinio[pos] \leq escrutinio[res])) \longrightarrow (|escrutinio| \geq 2) \wedge (0 \leq pos + 1 \leq |escrutinio|) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < pos + 1) \rightarrow_L (escrutinio[X] \leq escrutinio[res])$

$$((I \wedge B) \wedge (\text{escrutinio}[\text{pos}] \leq \text{escrutinio}[\text{res}])) \longrightarrow (|\text{escrutinio}| \geq 2) \wedge (0 \leq \text{pos} + 1 \leq |\text{escrutinio}|) \wedge (0 \leq \text{res} < |\text{escrutinio}|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |\text{escrutinio}|) \rightarrow_L (\text{escrutinio}[X] \leq \text{escrutinio}[\text{res}])$$

$$(((|\text{escrutinio}| \geq 2) \wedge (0 \leq \text{pos} < |\text{escrutinio}|) \wedge (0 \leq \text{res} < |\text{escrutinio}|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |\text{escrutinio}|) \rightarrow_L (\text{escrutinio}[X] \leq \text{escrutinio}[\text{res}])) \wedge (\text{escrutinio}[\text{pos}] \leq \text{escrutinio}[\text{res}])) \longrightarrow (0 \leq \text{pos} + 1 \leq |\text{escrutinio}|) \wedge (0 \leq \text{res} < |\text{escrutinio}|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |\text{escrutinio}|) \rightarrow_L (\text{escrutinio}[X] \leq \text{escrutinio}[\text{res}]) \equiv \mathbf{True}$$

$$5. \{I \wedge B \wedge vo = fv\} S_C \{fv < vo\}$$

$$I \equiv (|\text{escrutinio}| \geq 2) \wedge (0 \leq \text{pos} \leq |\text{escrutinio}|) \wedge (0 \leq \text{res} < |\text{escrutinio}|) \wedge (\forall X : \mathbb{Z})(0 \leq X < \text{pos}) \rightarrow_L (\text{escrutinio}[X] \leq \text{escrutinio}[\text{res}])$$

$$B \equiv \text{pos} < |\text{escrutinio}|$$

$$fv \equiv |\text{escrutinio}| - \text{pos}$$

$$\{I \wedge B \wedge vo = |\text{escrutinio}| - \text{pos}\} S_C \{|\text{escrutinio}| - \text{pos} < vo\}$$

$$\text{Quiero ver que } (I \wedge B \wedge vo = fv) \rightarrow wp(S_C, (fv < vo))$$

$$5.a \ I \wedge B \wedge vo = fv$$

$$I \wedge B \wedge vo = fv \equiv ((|\text{escrutinio}| \geq 2) \wedge (0 \leq \text{pos} < |\text{escrutinio}|) \wedge (0 \leq \text{res} < |\text{escrutinio}|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |\text{escrutinio}|) \rightarrow_L (\text{escrutinio}[X] \leq \text{escrutinio}[\text{res}])) \wedge (vo = |\text{escrutinio}| - \text{pos})$$

$$5.b \ wp(S_C, fv < vo)$$

$$wp(S_C, fv < vo) \equiv wp(\text{if}(\text{escrutinio}[\text{pos}] > \text{escrutinio}[\text{res}]) \text{ then } (\text{res} = \text{pos}) \text{ else } (\text{skip}) \text{ endif}, wp(\text{pos} = \text{pos} + 1, fv < vo))$$

En un picipio analizo:

$$wp(\text{pos} = \text{pos} + 1, fv < vo) \equiv (\text{pos} = \text{pos} + 1) \wedge (|\text{escrutinio}| - \text{pos} < vo)$$

$$wp(\text{pos} = \text{pos} + 1, fv < vo) \equiv \text{def}(\text{pos} = \text{pos} + 1) \wedge_L (\mathbf{True}) \wedge (|\text{escrutinio}| - \text{pos} - 1 < vo)$$

$$wp(\text{pos} = \text{pos} + 1, fv < vo) \equiv (|\text{escrutinio}| - \text{pos} - 1 < vo) \equiv E2$$

Al tratarse de un IF, se divide la implicancia para ambos casos

$$(I \wedge B \wedge vo = fv \wedge (\text{escrutinio}[\text{pos}] > \text{escrutinio}[\text{res}])) \rightarrow wp(\text{res} := \text{pos}, E2)$$

$$(I \wedge B \wedge vo = fv \wedge (\text{escrutinio}[\text{pos}] > \text{escrutinio}[\text{res}])) \rightarrow \text{def}(\text{res} := \text{pos}) \wedge_L (|\text{escrutinio}| - \text{res} - 1 < vo)$$

$$(I \wedge B \wedge vo = (|\text{escrutinio}| - \text{pos}) \wedge (\text{escrutinio}[\text{pos}] > \text{escrutinio}[\text{res}])) \rightarrow \text{def}(\text{res} := \text{pos}) \wedge_L (|\text{escrutinio}| - \text{res} - 1 < vo)$$

$$(I \wedge B \wedge vo = (|\text{escrutinio}| - \text{pos}) \wedge (\text{escrutinio}[\text{pos}] > \text{escrutinio}[\text{res}])) \rightarrow \text{def}(vo := |\text{escrutinio}| - \text{pos}) \wedge_L \text{def}(\text{res} := \text{pos}) \wedge_L (|\text{escrutinio}| - \text{res} - 1 < |\text{escrutinio}| - \text{pos}) \equiv \mathbf{True}$$

$$(I \wedge B \wedge vo = fv \wedge (\text{escrutinio}[\text{pos}] \leq \text{escrutinio}[\text{res}])) \rightarrow wp(\text{res} := \text{pos}, E2)$$

$$(I \wedge B \wedge vo = fv \wedge (\text{escrutinio}[\text{pos}] \leq \text{escrutinio}[\text{res}])) \rightarrow \text{def}(\text{res} := \text{pos}) \wedge_L (|\text{escrutinio}| - \text{res} - 1 < vo)$$

$$(I \wedge B \wedge vo = (|\text{escrutinio}| - \text{pos}) \wedge (\text{escrutinio}[\text{pos}] \leq \text{escrutinio}[\text{res}])) \rightarrow \text{def}(\text{res} := \text{pos}) \wedge_L (|\text{escrutinio}| - \text{res} - 1 < vo)$$

$$(I \wedge B \wedge vo = (|\text{escrutinio}| - \text{pos}) \wedge (\text{escrutinio}[\text{pos}] \leq \text{escrutinio}[\text{res}])) \rightarrow \text{def}(vo := |\text{escrutinio}| - \text{pos}) \wedge_L \text{def}(\text{res} := \text{pos}) \wedge_L (|\text{escrutinio}| - \text{res} - 1 < |\text{escrutinio}| - \text{pos}) \equiv \mathbf{True}$$

Habiendo demostrado cada uno de los pasos, queda demostrado que la tripla de Hoare:

$$\{\mathbf{True}\} \tag{2}$$

$$S_C \tag{3}$$

$$\{(|\text{escrutinio}| \geq 2) \wedge (0 \leq \text{res} < |\text{escrutinio}|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |\text{escrutinio}|) \rightarrow_L (\text{escrutinio}[X] \leq \text{escrutinio}[\text{res}])\}. \tag{4}$$

Teniendo esto en mente, si vemos el código escrito (Codigo 3) en SmallLang, podemos ver que si remplazamos  $res$  por  $res_0$  y  $res_1$  para cada uno de los ciclos, nos queda que las dos triplas de Hoare son validad, siempre y cuando analicemos el segundo ciclo como si en el escrutinio ignorara el valor de  $res_0$ , dado por la segunda condición en su IF, agregando a su postcondición el hecho de que  $(res_1 \neq res_0) \wedge (escrutinio[res_1] < escrutinio[res_0])$ . Por tanto queda demostrar la tripla:  $\{Pre\}S\{Post\}$

Para ello primero es necesario demostrar la validez de:

```

1 | pos := 0;
2 | res_0 := 0;
3 | res_1 := 1;
4 |
5 | if (escrutinio [res_0] < escrutinio [res_1]) do
6 |     res_0 := 1;
7 |     res_1 := 0;
8 | else
9 |     skip
10| endif

```

Código 4: Codigo de Ej3

Quedando una tripla de Hoare del tipo:

$$\{Pre_c\}S_c\{Post_c\}$$

$$Pre_c \equiv \{(|escrutinio| \geq 2) \wedge (pos = 0) \wedge (res_0 = 0) \wedge (res_1 = 1)\}$$

$$S_c \equiv if(escrutinio[res_0] < escrutinio[res_1])then(res_0 := 1; res_1 := 0)else(skip)endif$$

$$Post_c \equiv \{(|escrutinio| \geq 2) \wedge (pos = 0) \wedge ((res_0 = 0 \wedge res_1 = 1) \vee (res_0 = 1 \wedge res_1 = 0)) \wedge (escrutinio[res_0] > escrutinio[res_1])\}$$

Para probar su correctitud se debe demostrar que:

$$(Pre_c \wedge (escrutinio[res_0] < escrutinio[res_1])) \rightarrow wp(res_0 := 1; res_1 := 0, Post_c)$$

$$(Pre_c \wedge (escrutinio[res_0] \geq escrutinio[res_1])) \rightarrow wp(skip, Post_c)$$

Primero analizo cada  $wp$  individualmente:

$$\begin{aligned}
& wp(res_0 := 1; res_1 := 0, Post_c) \\
& \equiv wp(res_0 := 1, wp(res_1 := 0, Post_c)) \\
& \equiv wp(res_0 := 1, def(res_1 := 0) \wedge_L (((res_0 = 0 \wedge 0 = 1) \vee (res_0 = 1 \wedge 0 = 0)) \wedge (escrutinio[res_0] > escrutinio[0]))) \\
& \equiv def(res_0 := 1) \wedge_L ((\mathbf{False}) \vee (\mathbf{True})) \wedge (escrutinio[1] > escrutinio[0]) \\
& \equiv (\mathbf{True}) \wedge (escrutinio[1] > escrutinio[0])
\end{aligned}$$

$$wp(skip, Post) \equiv def(skip) \wedge_L ((res_0 = 0 \wedge res_1 = 1) \vee (res_0 = 1 \wedge res_1 = 0)) \wedge (escrutinio[res_0] > escrutinio[res_1])$$

Volviendo a las implicancias:

$$\begin{aligned}
& (Pre_c \wedge (escrutinio[res_0] < escrutinio[res_1])) \rightarrow wp(res_0 := 1; res_1 := 0, Post_c) \\
& ((|escrutinio| \geq 2) \wedge (pos = 0) \wedge (res_0 = 0) \wedge (res_1 = 1) \wedge (escrutinio[res_0] < escrutinio[res_1])) \rightarrow (|escrutinio| \geq 2) \wedge (pos = 0) \wedge ((res_0 = 0 \wedge 0 = 1) \vee (res_0 = 1 \wedge 0 = 0)) \wedge (escrutinio[1] > escrutinio[0]) \equiv \mathbf{True}
\end{aligned}$$

$$\begin{aligned}
& (Pre_c \wedge (escrutinio[res_0] \geq escrutinio[res_1])) \rightarrow wp(skip, Post_c) \\
& ((|escrutinio| \geq 2) \wedge (pos = 0) \wedge (res_0 = 0) \wedge (res_1 = 1) \wedge (escrutinio[res_0] \geq escrutinio[res_1])) \rightarrow (|escrutinio| \geq 2) \wedge (pos = 0) \wedge ((res_0 = 0 \wedge 0 = 1) \vee (res_0 = 1 \wedge 0 = 0)) \wedge (escrutinio[res_0] > escrutinio[res_1]) \equiv \mathbf{True}
\end{aligned}$$

Con esto queda demostrado la validez de la tripla de Hoare:

$$\{(|escrutinio| \geq 2) \wedge (pos = 0) \wedge (res_0 = 0) \wedge (res_1 = 1)\} \quad (5)$$

$$S_c \quad (6)$$

$$\{(|escrutinio| \geq 2) \wedge (pos = 0) \wedge ((res_0 = 0 \wedge res_1 = 1) \vee (res_0 = 1 \wedge res_1 = 0)) \wedge (escrutinio[res_0] > escrutinio[res_1])\}. \quad (7)$$

Con la validez del condicional demostrada, se puede comenzar a analizar la validez del programa completo, con la tripla de Hoare de  $\{Pre\}S\{Post\}$

$$Pre \equiv \{|escrutinio| \geq 2\}$$

$$Post \equiv \{(res = (res_0, res_1) \leftrightarrow (\exists res_0 : \mathbb{Z})(\exists res_1 : \mathbb{Z})(\forall x : \mathbb{Z})(x \neq res_0 \wedge x \neq res_1)(0 \leq res_0, res_1, x \leq |escrutinio|) \rightarrow_L (escrutinio[x] < escrutinio[res_1]) \wedge (escrutinio[res_1] < escrutinio[res_0]))\}$$

Para demostrar  $\{Pre\}S\{Post\}$  es necesario que  $Pre \rightarrow wp(S, Post)$ , y para demostrar ello se deben cumplir:

$$Pre \rightarrow wp(pos := 0; res_0 := 0; res_1 := 1, Pre_{Cond})$$

$$Pre_{Cond} \rightarrow wp(SC_{Cond}, Post_{Cond})$$

$$Post_{Cond} \rightarrow wp(skip, Pre_{Ci1})$$

$$Pre_{Ci1} \rightarrow wp(SC_{i1}, Post_{Ci1})$$

$$Post_{Ci1} \rightarrow wp(pos := 0, Pre_{Ci2})$$

$$Pre_{Ci2} \rightarrow wp(SC_{i2}, Post_{Ci2})$$

$$Post_{Ci2} \rightarrow wp(res := (res_0, res_1), Post)$$

Por las demostraciones de correctitud antes dadas, solo quedaria demostrar que:

$$Pre \rightarrow wp(pos := 0; res_0 := 0; res_1 := 1, Pre_{Cond})$$

$$Post_{Cond} \rightarrow wp(skip, Pre_{Ci1})$$

$$Post_{Ci1} \rightarrow wp(pos := 0, Pre_{Ci2})$$

$$Post_{Ci2} \rightarrow wp(res := (res_0, res_1), Post)$$

Entonces:

$$Pre \rightarrow wp(pos := 0; res_0 := 0; res_1 := 1, Pre_{Cond})$$

$$Pre \rightarrow wp(pos := 0, wp(res_0 := 0, wp(res_1 := 1, Pre_{Cond})))$$

$$Pre \rightarrow wp(pos := 0, wp(res_0 := 0, def(res_1 := 1) \wedge_L Pre_{Cond}))$$

$$Pre \rightarrow wp(pos := 0, wp(res_0 := 0, (|escrutinio| \geq 2) \wedge (pos = 0) \wedge (res_0 = 0) \wedge (1 = 1)))$$

$$Pre \rightarrow wp(pos := 0, def(res_0 := 0) \wedge_L (|escrutinio| \geq 2) \wedge (pos = 0) \wedge (0 = 0) \wedge (\mathbf{True}))$$

$$Pre \rightarrow def(pos := 0) \wedge_L (|escrutinio| \geq 2) \wedge (0 = 0) \wedge (\mathbf{True})$$

$$(|escrutinio| \geq 2) \rightarrow (|escrutinio| \geq 2) \wedge (\mathbf{True}) \equiv \mathbf{True}$$

$$Post_{Cond} \rightarrow wp(skip, Pre_{Ci1})$$

$$Post_{Cond} \rightarrow Pre_{Ci1}$$

$$Post_{Cond} \rightarrow (|escrutinio| \geq 2) \wedge (res_0 = 0 \vee res_0 = 1) \wedge (pos = 0)$$

$$((|escrutinio| \geq 2) \wedge (pos = 0) \wedge ((res_0 = 0 \wedge res_1 = 1) \vee (res_0 = 1 \wedge res_1 = 0)) \wedge (escrutinio[res_0] > escrutinio[res_1])) \rightarrow (|escrutinio| \geq 2) \wedge (res_0 = 0 \vee res_0 = 1) \wedge (pos = 0) \equiv \mathbf{True}$$

$$Post_{Ci1} \rightarrow wp(pos := 0, Pre_{Ci2})$$

$$Post_{Ci1} \rightarrow def(pos := 0) \wedge_L (|escrutinio| \geq 2) \wedge (res_1 = 0 \vee res_1 = 1) \wedge (0 = 0)$$

$$((|escrutinio| \geq 2) \wedge (0 \leq res < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio| \rightarrow_L (escrutinio[X] \leq escrutinio[res]))) \rightarrow (|escrutinio| \geq 2) \wedge (res_1 = 0 \vee res_1 = 1) \wedge (\mathbf{True}) \equiv \mathbf{True}$$

$$Post_{Ci2} \rightarrow wp(res := (res_0, res_1), Post)$$

$$Post_{Ci2} \rightarrow def(res := (res_0, res_1)) \wedge_L ((res_0, res_1) = (res_0, res_1) \leftrightarrow (\exists res_0 : \mathbb{Z})(\exists res_1 : \mathbb{Z})(\forall x : \mathbb{Z})(x \neq res_0 \wedge x \neq res_1)(0 \leq res_0, res_1, x \leq |escrutinio|) \rightarrow_L (escrutinio[x] < escrutinio[res_1]) \wedge (escrutinio[res_1] < escrutinio[res_0]))$$

$$Post_{Ci2} \rightarrow (\mathbf{True} \leftrightarrow (\exists res_0 : \mathbb{Z})(\exists res_1 : \mathbb{Z})(\forall x : \mathbb{Z})(x \neq res_0 \wedge x \neq res_1)(0 \leq res_0, res_1, x \leq |escrutinio|) \rightarrow_L (escrutinio[x] < escrutinio[res_1]) \wedge (escrutinio[res_1] < escrutinio[res_0]))$$

$$(((|escrutinio| \geq 2) \wedge (0 \leq res_1 < |escrutinio|) \wedge (\forall X : \mathbb{Z})(0 \leq X < |escrutinio| \wedge X \neq res_0) \rightarrow_L (escrutinio[X] \leq escrutinio[res_1])) \wedge (res_1 \neq res_0) \wedge (escrutinio[res_1] < escrutinio[res_0])) \rightarrow (\mathbf{True} \leftrightarrow (\exists res_0 : \mathbb{Z})(\exists res_1 : \mathbb{Z})(\forall x : \mathbb{Z})(x \neq res_0 \wedge x \neq res_1)(0 \leq res_0, res_1, x \leq |escrutinio|) \rightarrow_L (escrutinio[x] < escrutinio[res_1]) \wedge (escrutinio[res_1] < escrutinio[res_0])) \equiv \mathbf{True}$$

Con estas ultimas demostraciones, queda demostrada la validez del codigo y el cumplimineto de la tripla de Hoarde:  $\{Pre\}S\{Post\}$

## 4. Ejercicio 4: calcularDHondtEnProvincia

calcularDHondtEnProvincia: calcula los cocientes según el método d'Hondt para diputados en una provincia (importante: no es necesario ordenar los partidos por cantidad de votos)

proc calcularDHondtEnProvincia (in cant\_bancas:  $\mathbb{Z}$ , in escrutinio: seq( $\mathbb{Z}$ )) : seq(seq( $\mathbb{Z}$ ))

Donde:

- cantBancas: es la cantidad de bancas en disputa en la provincia
- escrutinio: es la cantidad de votos de cada partido en la provincia
- devuelve la matriz de dimensión partidos  $\times$  cocientes de los cocientes del método d'Hondt.

### 4.1. Especificación

proc calcularDHondtEnProvincia (in cant\_bancas:  $\mathbb{Z}$ , in escrutinio: seq( $\mathbb{Z}$ )) : seq(seq( $\mathbb{Z}$ ))

  requiere {sinRepetidos(escrutinio)  $\wedge$  votosValidos(escrutinio)}

  asegura { $(\forall i : \mathbb{Z})(\forall j : \mathbb{Z})(0 \leq i < |\text{escrutinio}| - 1 \wedge_L 0 \leq j < \text{cant\_bancas} \longrightarrow (|\text{res}| = |\text{escrutinio}| \wedge_L |\text{res}[i]| = \text{cant\_bancas} \wedge \text{res}[i][j] = \frac{\text{escrutinio}[i]}{j+1}))$ }

## 5. Ejercicio 5: obtenerDiputadosEnProvincia

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

proc obtenerDiputadosEnProvincia (in cant\_bancas:  $\mathbb{Z}$ , in escrutinio:  $\text{seq}\langle\mathbb{Z}\rangle$ , in dHondt:  $\text{seq}\langle\text{seq}\langle\mathbb{Z}\rangle\rangle$ ) :  $\text{seq}\langle\mathbb{Z}\rangle$

Donde:

- cant\_bancas: es la cantidad de bancas en disputa en la provincia.
- escrutinio: es la cantidad de votos de cada partido en la provincia.
- dHondt: es la matriz de dimensión  $\#\text{partidos} \times \#\text{cocientes}$  de los cocientes del método dHondt.
- devuelve la cantidad de bancas obtenidas por cada partido.

### 5.1. Especificación

```

proc obtenerDiputadosEnProvincia ( in cant_bancas:  $\mathbb{Z}$ , in escrutinio:  $\text{seq}\langle\mathbb{Z}\rangle$ , in dHondt:  $\text{seq}\langle\text{seq}\langle\mathbb{Z}\rangle\rangle$ ) :  $\text{seq}\langle\mathbb{Z}\rangle$ 
  requiere {cant_bancas > 0}
  requiere {sinRepetidos(escrutinio)  $\wedge$  votosValidos(escrutinio)}
  requiere {sinComponentesRepetidos(dHondt)}
  requiere {dHondtValido(cant_bancas, escrutinio, dHondt)}
  requiere {( $\exists i : \mathbb{Z}$ )( $0 \leq i < |\text{escrutinio}| - 1 \wedge_L \text{superaUmbral}(\text{escrutinio}[i], \text{escrutinio})$ )}
  asegura {|res| = |escrutinio| - 1}
  asegura {( $\sum_{i=0}^{|\text{res}|-1} \text{res}[i] = \text{cant\_bancas}$ )}
  asegura {( $\forall i : \mathbb{Z}$ )( $0 \leq i < |\text{escrutinio}| - 1 \rightarrow_L (\neg(\text{superaUmbral}(\text{escrutinio}[i], \text{escrutinio})) \wedge \text{res}[i] = 0) \vee$ 
    ( $\text{superaUmbral}(\text{escrutinio}[i], \text{escrutinio}) \wedge (\exists s : \text{seq}\langle\mathbb{Z}\rangle)(\text{sonLosMaximos}(s, \text{dHondt}, \text{cant\_bancas}) \wedge$ 
     $\text{res}[i] = \sum_{j=0}^{\text{cant\_bancas}} \text{if pertenece}(\text{dHondt}[i][j], s) \text{ then } 1 \text{ else } 0 \text{ fi})$ )}
  pred sonLosMaximos (s:  $\text{seq}\langle\mathbb{Z}\rangle$ , matriz:  $\text{seq}\langle\text{seq}\langle\mathbb{Z}\rangle\rangle$ , n:  $\mathbb{Z}$ ) {
    (esMaximoComponente(s[0], matriz)  $\wedge$  ( $\forall i : \mathbb{Z}$ )( $0 \leq i < n - 1 \rightarrow_L (s[i] > s[i + 1]) \wedge (\text{perteneceAMatriz}(s[i], \text{matriz}))$ 
  )
  }
  pred esMaximoComponente (elem :  $\mathbb{Z}$ , matriz:  $\text{seq}\langle\text{seq}\langle\mathbb{Z}\rangle\rangle$ ) {
    matrizNoVacia(matriz)  $\wedge$  ( $\forall i, j : \mathbb{Z}$ )( $0 \leq i < |\text{matriz}| \wedge 0 \leq j < |\text{matriz}[i]| \rightarrow_L \text{elem} > \text{matriz}[i][j]$ )
  }
  pred pertenece (elem:  $\mathbb{Z}$ , lista:  $\text{seq}\langle\mathbb{Z}\rangle$ ) {
    (|lista| > 0)  $\wedge$  ( $\exists j : \mathbb{Z}$ )( $0 \leq j < |\text{lista}| \wedge_L \text{lista}[j] = \text{elem}$ )
  }
  pred perteneceAMatriz (elem:  $\mathbb{Z}$ , matriz:  $\text{seq}\langle\text{seq}\langle\mathbb{Z}\rangle\rangle$ ) {
    matrizNoVacia(matriz)  $\wedge$  ( $\exists i : \mathbb{Z}$ )( $0 \leq i < |\text{matriz}| \wedge_L \text{pertenece}(\text{elem}, \text{matriz}[i])$ )
  }
  pred superaUmbral (votos:  $\mathbb{Z}$ , escrutinio:  $\text{seq}\langle\mathbb{Z}\rangle$ ) {
    votos  $\geq (\sum_{j=0}^{|\text{escrutinio}|-1} \text{escrutinio}[j]) * 0,03$ 
  }
  pred sinComponentesRepetidos (matriz:  $\text{seq}\langle\text{seq}\langle\mathbb{Z}\rangle\rangle$ ) {
    matrizNoVacia(matriz)  $\wedge$  ( $\forall i, j, k, s : \mathbb{Z}$ )( $0 \leq i, k < |\text{matriz}| \wedge 0 \leq j, s < |\text{matriz}[i]| \wedge (i, j) \neq (k, s) \rightarrow_L \text{matriz}[i][j] \neq$ 
     $\text{matriz}[k][s]$ )
  }
  pred dHondtValido (cant_bancas:  $\mathbb{Z}$ , escrutinio:  $\text{seq}\langle\mathbb{Z}\rangle$ , dHondt:  $\text{seq}\langle\text{seq}\langle\mathbb{Z}\rangle\rangle$ ) {
    (|escrutinio| = |dHondt|)  $\wedge$  (( $\forall i, j : \mathbb{Z}$ )( $0 \leq i < |\text{escrutinio}| - 1 \wedge 0 \leq j < \text{cant\_bancas} \rightarrow_L |\text{dHondt}[i]| = \text{cant\_bancas} \wedge$ 
     $\text{dHondt}[i][j] = \frac{\text{escrutinio}[i]}{j+1}$ ))
  }

```



## 6. Ejercicio 6: validarListasDiputadosEnProvincia

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 además se cumpla la alternancia de géneros.

proc validarListasDiputadosEnProvincia (in cant\_bancas:  $\mathbb{Z}$ , in listas:  $\text{seq}(\text{seq}(\text{dni} : \mathbb{Z} \times \text{genero} : \mathbb{Z}))$ ) : Bool  
Donde:

- cant\_bancas: es la cantidad total de bancas en disputa en la provincia.
- listas: son las listas de diputados de cada partido. Cada candidato/a está representado/a con una tupla que contiene el dni y el género.
- devuelve verdadero sii las listas de todos los partidos: 1) presentan la cantidad correcta de candidatos, y 2) verifican la alternancia de género.

### 6.1. Especificación

```
proc validarListasDiputadosEnProvincia (in cant_bancas:  $\mathbb{Z}$ , in listas:  $\text{seq}(\text{seq}(\text{dni} : \mathbb{Z} \times \text{genero} : \mathbb{Z}))$ ) : Bool
  requiere {cant_bancas > 0}
  requiere {matrizNoVacía(listas)}
  requiere {( $\forall i, j : \mathbb{Z}$ )( $0 \leq i < |listas| \wedge 0 \leq j < |listas[i]| \rightarrow_L (listas[i][j][0] > 0) \wedge ((listas[i][j][1] = 1) \vee (listas[i][j][1] = 2))$ ))}
  asegura {res = True  $\leftrightarrow$  (( $\forall i : \mathbb{Z}$ )( $0 \leq i < |listas| \rightarrow_L |listas[i]| = \text{cant\_bancas}$ )  $\wedge$  ( $\forall i : \mathbb{Z}$ )( $0 \leq i < |listas| \rightarrow_L (\forall j : \mathbb{Z})(0 \leq j < |listas[i]| - 1 \rightarrow_L listas[i][j][1] \neq listas[i][j+1][1])$ ))}}
```

### 6.2. Implementación

```
1 res := True;
2 partido_index := 0;
3
4 while ( partido_index < listas.size() ) do
5
6   partido := listas[partido_index];
7
8   if cant_bancas != partido.size() then
9     res := False;
10  else
11    skip
12  endif
13
14  diputado_index := 0;
15
16  while ( diputado_index < partido.size() - 1 ) do
17
18    if ( partido[diputado_index][1] == partido[diputado_index + 1][1] ) then
19      res := False;
20    else
21      skip
22    endif
23
24    diputado_index := diputado_index + 1;
25
26  endwhile
27
28  partido_index := partido_index + 1;
29
30 endwhile
```

Código 5: Codigo de Ej6