DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo práctico 1: Especificación y WP

Algoritmos y Estructuras de Datos - DC - UBA

20 de octubre de 2023 Algoritmos y Estructuras de Datos

QueGrupoGerson

Integrante	LU	Correo electrónico
Andres, Sebastián	1028/22	sebastian.ignacio.andres@gmail.com
Cellerino, Juan	697/22	jcellerino@gmail.com
Fuentes Urfeig, Pedro	1088/22	pedrofuentes7799@gmail.com
Tenconi, Vicente	1171/22	tenconivini@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. Especificaciones

1.1. hayBallotage

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

`requiere` $\{esEscrutinioValido(escrutinio)\}$

`asegura` $\{res = True \iff \neg(hayMas45(escrutinio) \lor hayDif10Mas40(escrutinio))\}$

`pred hayMas45` (s :$seq\langle\mathbb{Z}\rangle$) {

$(\exists i : \mathbb{Z})(0 \leq i < |s| - 1 \land_L \frac{s[i]}{suma(s)} > 0{,}45)$

}

`pred hayDif10Mas40` (s :$seq\langle\mathbb{Z}\rangle$) {

$(\exists i : \mathbb{Z})(0 \leq i < |s| - 1 \land_L \frac{s[i]}{suma(s)} > 0{,}4 \land_L (\forall j : \mathbb{Z})(j \neq i \land 0 \leq j < |s| - 1 \rightarrow_L \frac{s[i]-s[j]}{suma(s)} > 0{,}1)$

}

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

Observación: Definimos estos predicados fuera del procedimiento para reutilizarlos en los proximos ejercicios (tal como nos indicaron en la corrección).

`pred esEscrutinioValido` (escrutinio:$seq\langle\mathbb{Z}\rangle$) {

$(|escrutinio| \geq 3) \land \neg(esPrimero(|i| - 1, escrutinio)) \land$

$(\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L \neg(\exists j : \mathbb{Z})(0 \leq j < |s| \land_L i \neq j \land s[i] = s[j])) \land$

$((\exists i, j : \mathbb{Z})(0 \leq i < |s| - 1) \land (0 \leq j < |s| - 1) \land_L esPrimero(i, escrutinio) \land esSegundo(i, j, escrutinio))$

}

`pred esPrimero` (max : $\mathbb{Z}$, s : $seq\langle\mathbb{Z}\rangle$) {

$(0 \leq max < |s| \land (\forall j \in \mathbb{Z})(0 \leq j < |s| \land j \neq max \rightarrow_L s[j] < s[max]))$

}

`pred esSegundo` (max: $\mathbb{Z}$, snd : $\mathbb{Z}$, s : $seq\langle\mathbb{Z}\rangle$) {

$(0 \leq max < |s| \land (\forall j \in \mathbb{Z})(0 \leq j < |s| \land j \neq max \land j \neq snd \rightarrow_L s[j] < s[snd]))$

}

1.2. hayFraude

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

`requiere` {

$esEscrutinioValido(escrutinio\_presidencial) \land$

$esEscrutinioValido(escrutinio\_diputados) \land$

$esEscrutinioValido(escrutinio\_senadores)$

}

`asegura` {

$res = False \iff$

$(suma(escrutinio\_presidenciales) == suma(escrutinio\_diputados) \land$

$suma(escrutinio\_diputados) == suma(escrutinio\_senadores))$

}

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

1.3. obtenerSenadoresEnProvincia

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

`requiere` $\{esEscrutinioValido(escrutinio)\}$

`asegura` $\{$

$(res_0 \neq res_1 \wedge_L esPrimero(res_0, escrutinio) \wedge_L esSegundo(res_0, res_1, escrutinio))$

$\}$

1.4. calcularDHondtEnProvincia

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

`requiere` $\{cantBancas > 0 \wedge esEscrutinioValido(escrutinio)\}$

`asegura` $\{|res| = |escrutinio| - 1\}$

`asegura` $\{$

$(\forall i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \rightarrow_L$

$(escrutinio[i] > umbral3p(escrutinio) \wedge_L esCocienteSimple(res[i], escrutinio[i], cantBancas)) \vee_L$

$(escrutinio[i] \leq umbral3p(escrutinio)) \wedge_L esListaDeNCeros(res[i], cantBancas))$

$\}$

`asegura` $\{noHayCocientesRepetidosExceptoCero(res)\}$

`pred esCocienteSimple` (cocientes: $seq\langle\mathbb{Z}\rangle$, votosObtenidos : $\mathbb{Z}$, cantBancas : $\mathbb{Z}$) $\{$

$(\forall i : \mathbb{Z})(0 < i \leq cantBancas \rightarrow_L cocientes[i] = \frac{votosObtenidos}{i})$

$\}$

`pred esListaDeNCeros` (cocientes, N: $\mathbb{Z}$) $\{$

$|cocientes| = N \wedge_L (\forall i : \mathbb{Z})(0 \leq i < |cocientes| \rightarrow_L cocientes[i] = 0)$

$\}$

`pred noHayCocientesRepetidosExceptoCero` (dhont: $seq\langle seq\langle\mathbb{Z}\rangle\rangle$) $\{$

$(\forall j, i, j', i' \in \mathbb{Z})(0 \leq j, i, j', i' < |dhont| \wedge dhont[j][i] == dhont[j'][i'] \iff$

$((dhont[j][i] == 0) \vee (j = j' \wedge i = i'))$

$\}$

`aux umbral3p` (in escrutinio: $seq\langle\mathbb{Z}\rangle$) : $\Re$ $=(\sum_{i=0}^{|escrutinio|-1} escrutinio[i]) * 0{,}03$;

1.5. obtenerDiputadosEnProvincia

`proc obtenerDiputadosEnProvincia` (in cantBancas : $\mathbb{Z}$, in dHont : $seq\langle seq\langle \mathbb{Z}\rangle\rangle$, in escrutinio: $seq\langle \mathbb{Z}\rangle$) : $seq\langle \mathbb{Z}\rangle$

`requiere` $\{esEscrutinioValido(escrutinio) \land (cantBancas > 0) \land (|dHont| = |escrutinio| - 1) \land$
$(\forall i : \mathbb{Z})(0 \leq i < |escrutinio| - 1 \rightarrow_L$
$(escrutinio[i] > umbral3p(escrutinio) \land_L esCocienteSimple(dHont[i], escrutinio[i], cantBancas)) \lor_L$
$(escrutinio[i] \leq umbral3p(escrutinio)) \land_L esListaDeNCeros(dHont[i], cantBancas))\}$

`asegura` $\{|res| = |dHont|\}$

`asegura` $\{\sum_{i=0}^{|res|-1} res[i] = cantBancas\}$

`asegura` $\{(\forall i : \mathbb{Z})(0 \leq i < |dHont| \land esCantidadDeEscaños(res[i], dHont[i], dHont))\}$

`pred esCocienteSimple` (cocientes: $seq\langle \mathbb{Z}\rangle$, votosObtenidos : $\mathbb{Z}$, cantBancas : $\mathbb{Z}$) {
$(\forall i : \mathbb{Z})(0 < i \leq cantBancas \rightarrow_L cocientes[i] = \frac{votosObtenidos}{i})$
}

`pred esListaDeNCeros` (cocientes: $seq\langle \mathbb{Z}\rangle$, N: $\mathbb{Z}$) {
$(|cocientes| == N) \land_L (\forall i : \mathbb{Z})(0 \leq i < |cocientes| \rightarrow_L cocientes[i] = 0)$
}

`pred esCantidadDeEscaños` (x: $\mathbb{Z}$, bancasEnDisputa: $\mathbb{Z}$, cocientes: $seq\langle \mathbb{Z}\rangle$, dhont: $seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$x == \sum_{k=0}^{|cocientes|-1} \text{if } estaEntreLosNMasGrandes(cocientes[k], bancasEnDisputa, dhont) \text{ then } 1 \text{ else } 0 \text{ fi}$
}

`pred estaEntreLosNMasGrandes` (c: $\mathbb{Z}$, N: $\mathbb{Z}$, dhont:$seq\langle seq\langle \mathbb{Z}\rangle\rangle$) {
$N < \sum_{j=0}^{|dhont|-1} \sum_{i=0}^{|dhont[j]|-1} \text{if } c > dhont[j][i] \text{ then } 1 \text{ else } 0 \text{ fi}$
}

1.6. validarListasDiputadosEnProvincia

`proc validarListasDiputadosEnProvincia` (in cantBancas : $\mathbb{Z}$, in listas : $seq\langle seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle\rangle$) : Bool

`requiere` $\{|listas| > 0 \land generosSon1o2(listas)\}$

`asegura` $\{res = True \iff$
$(\forall i : \mathbb{Z})(0 \leq i < |listas| \rightarrow_L cantidadValida(cant\_bancas, listas[i]) \land alternancia(listas[i])\}$

`pred cantidadValida` (cant_bancas: $\mathbb{Z}$, lista: $seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle$) {
$cant\_bancas = |lista|$
}

`pred alternancia` (lista: $seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle$) {
$(\forall i : \mathbb{Z})(0 \leq i < |lista| - 1 \rightarrow_L lista[i]_1 \neq lista[i+1]_1)$
}

`pred generosSon1o2` (listas: $seq\langle seq\langle dni : \mathbb{Z} \times genero : \mathbb{Z}\rangle\rangle\rangle$) {
$(\forall i : \mathbb{Z})(0 \leq i < |listas| - 1 \land$
$(\forall j : \mathbb{Z})(0 \leq j \leq |listas[i]| \rightarrow_L listas[i][j]_1 \in \{1, 2\}))$
}

2. Implementaciones

2.1. hayBallotage

xs $seq\langle\mathbb{Z}\rangle$

```
totales := 0
max := 0

while (i < xs.size()) do
     if (s[i] > s[max])
          max := i
     else
          skip
     endif
     totales := totales + s[i]
     i := i + 1
endwhile

j := 0
if (max = 0)
     snd := 1
else
     snd := 0
endif

while (j < xs.size())
     if (j != max && s[j] > s[snd])
          snd := j
     else
          skip
     endif
endwhile

if (s[max] / totales > 0.45) || ((s[max] / totales - s[snd] / totales) > 0.1) && (s[max] / totales > 0.4))
     res := false
else
     res := true
endif
```

Código 1: Código en SmallLang del Ejercicio 1

2.2. hayFraude

escrutinio_presidencial: $seq\langle\mathbb{Z}\rangle$, escrutinio_diputados: $seq\langle\mathbb{Z}\rangle$, escrutinio_senadores: $seq\langle\mathbb{Z}\rangle$

```
presidencial := Sumatoria(escrutinio_presidencial)
diputados := Sumatoria(escrutinio_diputados)
senadores := Sumatoria(escrutinio_senadores)

if (presidencial = diputados && diputados = senadores)
    res := false
else
    res := true
```

,

```
proc Sumatoria(s: List[Z]): Z
    i := 0
    res := 0
    while (i < s.size()) do
        res := res + s[i]
        i := i + 1
    endwhile
```

2.3. obtenerSenadoresEnProvincia

[H] escrutinio $seq\langle\mathbb{Z}\rangle$

```
i := 0
max := 0

while (i < s.size()) do
    if (s[i] > s[max])
        max := i
    else
        skip
    endif
    i := i + 1
endwhile

j := 0
if(max = 0)
    snd := 1
else
    snd := 0
endif

while (j < s.size())
    if( j != max && s[j] > s[snd])
        snd := j
    else
        skip
    endif
endwhile

res := (max, snd)
```

Código 2: Código en SmallLang del Ejercicio 3

2.4. validarListasDiputadosEnProvincia

[H] in cant_bancas: $\mathbb{Z}$, in listas: $seq\langle seq\langle dni : \mathbb{Z} genero : \mathbb{Z}>>$

```
res := true

i := 0
while (i < listas.size()) do
    if (listas[i].size() == cant_bancas)
        fst := listas[i][0]
        if (fst == 1):
            snd := 2
        else:
            snd := 1
        values := (fst, snd)
        j := 1
        while (j < cant_bancas) do
            if (listas[i][j] != values[j % 2])
                res := false
            else
                skip
            j := j + 1
    else:
        res := false
    i := i + 1
```

Código 3: Código en SmallLang del Ejercicio 6

Observación: Consideramos al operador % como el módulo aritmético o resto. Es decir, $A\%B = N \iff A(modB) \equiv N$.

3. Demostraciones

Observación: Nos referiremos a $S_{i\to j}$ como el bloque de código correspondiente a las lineas desde i a j.

3.1. hayFraude

Demostración: Para demostrar que el programa es válido vamos a utilizar los axiomas para asignación usando procedimientos.

De este modo, la demostración del ciclo la hacemos dentro del procedimiento **sumatoria** y podemos demostrar el programa mediante weakest precondition anidadas.

QvQ: $Pre \longrightarrow wp(S_{1\longrightarrow 8}, Post)$

Donde:

$$wp(S_{1\longrightarrow 8}, Post) = wp(S_1, wp(S_2, wp(S_3, wp(S_4, wp(S_{5\longrightarrow 8}, Post)))))$$

La weakest precondition del condicional,

$$wp(S_{5\longrightarrow 8}, Post) \equiv ((p = d \land d = s) \land wp(res := False, Post)) \lor (\neg(p = d \land d = s) \land wp(res := True, Post))$$

$$wp(S_{5\longrightarrow 8}, Post) \equiv ((p = d \land d = s) \land Post^{res}_{False}) \lor (\neg(p = d \land d = s) \land Post^{res}_{True})$$

Utilizo una variable auxiliar D tal que ...

$$D \equiv p = d \land d = s$$

Además, defino M tal que...

$$M \equiv suma(xp) = suma(xd) \land suma(xd) = suma(xs)$$

$$wp(S_{5\longrightarrow 8}, Post) \equiv (D \land M) \lor (\neg D \land \neg M)$$

Entonces,

$$wp(S_1, wp(S_2, wp(S_3, wp(S_4, wp(S_{5\longrightarrow 8}, Post)))))$$

$$\equiv wp(p := sumatoria(xp), wp(d := sumatoria(xd), wp(s := sumatoria(xs), (D \land M) \lor (\neg D \land \neg M)))))$$

Aplicando el Axioma 5, equivale a reemplazar en cada variable los valores:

$$\equiv ((D \land M) \lor (\neg D \land \neg M))^{s,d,p}_{sumatoria(xs), sumatoria(xd), sumatoria(xp)}$$

Luego, vamos a demostrar que la tripla de Hoare {Pre} Sumatoria {Post} es valida, entonces puedo decir que:

$$wp(x := \text{Call Sumatoria(ls)}, Q) \equiv def(E) \land Pre^{input}_{ls} \land (\forall r : Post^{input,result}_{E,r} \longrightarrow Q^x_r)$$

O, en términos más prácticos:

$$wp(x := \text{Call Sumatoria(ls)}, Q) \equiv Q^x_{res}$$

Donde res es el resultado del procedimiento sumatoria, que cumple la postcondición $res = \sum_{j=0}^{|ls|-1} ls[j]$.

Entonces, volviendo con la WP:

$$\equiv ((D \land M) \lor (\neg D \land \neg M))^{s,d,p}_{sumatoria(xs), sumatoria(xd), sumatoria(xp)}$$

$$\equiv (((p = d \land d = s) \land (suma(xp) = suma(xd) \land suma(xd) = suma(xs)))$$
$$\lor(\neg(p = d \land d = s) \land \neg(suma(xp) = suma(xd) \land suma(xd) = suma(xs))))^{s,d,p}_{sumatoria(xs), sumatoria(xd), sumatoria(xp)}$$

$$\equiv ((sumatoria(xp) = sumatoria(xd) \land sumatoria(xd) = sumatoria(xs)) \land (suma(xp) = suma(xd) \land suma(xd) = suma(xs))$$
$$\lor(\neg(sumatoria(xp) = sumatoria(xd) \land sumatoria(xd) = sumatoria(xs)) \land \neg(suma(xp) = suma(xd) \land suma(xd) = suma(xs)))$$

Reemplazamos sumatoria por la postcondición del procedimiento.

$$\equiv ((\sum_{j=0}^{|xp|-1} xp[j] = \sum_{j=0}^{|xd|-1} xp[j] \land \sum_{j=0}^{|xd|-1} xd[j] = \sum_{j=0}^{|xs|-1} xs[j]) \land (suma(xp) = suma(xd) \land suma(xd) = suma(xs))$$

$$\lor(\neg(\sum_{j=0}^{|xp|-1} xp[j] = \sum_{j=0}^{|xd|-1} xd[j] \land \sum_{j=0}^{|xd|-1} xd[j] = \sum_{j=0}^{|xs|-1} xs[j]) \land \neg(suma(xp) = suma(xd) \land suma(xd) = suma(xs)))$$

Teniendo en cuenta nuestra auxiliar cumple $suma(ls) = \sum_{j=0}^{|ls|-1} ls[j]$, entonces los términos dentro de el OR lógico son equivalentes.

$$\equiv ((\sum_{j=0}^{|xp|-1} xp[j] = \sum_{j=0}^{|xd|-1} xp[j] \wedge \sum_{j=0}^{|xd|-1} xd[j] = \sum_{j=0}^{|xs|-1} xs[j])$$

$$\vee(\neg(\sum_{j=0}^{|xp|-1} xp[j] = \sum_{j=0}^{|xd|-1} xd[j] \wedge \sum_{j=0}^{|xd|-1} xd[j] = \sum_{j=0}^{|xs|-1} xs[j]))$$

Lo cual es una tautología de la forma $A \vee \neg A$.
Entonces...

$$wp(S_{1 \longrightarrow 8}, Post) \equiv True$$

Luego, $(Pre \rightarrow wp(S_{1 \longrightarrow 8}, Post)) \equiv True$
Entonces la tripla de Hoare es válida.

3.2. Sumatoria

`proc Sumatoria` (in s: $\mathbb{Z}$) : $\mathbb{Z}$
 `requiere` $\{True\}$
 `asegura` $\{res = \sum_{j=0}^{|s|-1} s[j]\}$

Tenemos que probar que $\{True\}$ Sumatoria $\{res = \sum_{j=0}^{|s|-1} s[j]\}$ es una tripla de Hoare valida. Para eso, tienen que cumplirse los postulados del **Teorema del invariante** y el **Teorema de Terminación de un ciclo**.

Ademas debe valer el codigo antes del comienzo del ciclo (dentro del procedimiento):
0. {Pre} $S_{2\to3}$ $\{P_C\}$

Teorema del invariante

1. $P_C \longrightarrow I$
2. $\{I \wedge B\}$ $S_{5\to6}$ $\{I\}$
3. $I \wedge \neg B \longrightarrow Q_C$

Teorema de la terminacion de un ciclo

1. $\{I \wedge B \wedge fv = v_0\}$ $S_{5\to6}$ $\{fv < v_0\}$
2. $I \wedge fv \leq 0 \longrightarrow \neg B$

Predicados

- $Pre \equiv True$
- $P_C \equiv i = 0 \wedge res = 0$
- $I \equiv 0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]$
- $Q_C \equiv i = |s| \wedge res = \sum_{j=0}^{|s|-1} s[j]$

Demostración. Ítem 0

$$\{Pre\}S_{2\to 3}\{P_C\}$$

$$Pre \longrightarrow wp(i := 0, wp(res := 0, P_C))$$

$$Pre \longrightarrow 0 = 0 \land 0 = 0$$

$$Pre \longrightarrow True$$

□

Demostración. Teorema del Invariante. Ítem 1

$$P_C \longrightarrow I$$

$$i = 0 \land res = 0 \longrightarrow 0 \leq 0 \leq |s| \land 0 = \sum_{j=0}^{0-1} s[j]$$

$$P_C \longrightarrow True$$

□

Demostración. Teorema del Invariante. Ítem 2

$$\{I \land B\}S_{5\to 6}\{I\}$$

$$I \land B \longrightarrow (res := res + s[i], wp(i := i + 1, I))$$

$$0 \leq i < |s| \land res = \sum_{j=0}^{i-1} s[j] \longrightarrow 0 \leq i + 1 \leq |s| \land res + s[i] = \sum_{j=0}^{i} s[j]$$

El primer término es implicado por $I \land B$.

$$0 \leq i < |s| \land res = \sum_{j=0}^{i-1} s[j] \longrightarrow res = \sum_{j=0}^{i-1} s[j]$$

Lo cual es trivialmente cierto, pues la sumatoria del consecuente se encuentra en el precedente.

□

Demostración. Teorema del Invariante. Ítem 3

$$I \land \neg B \longrightarrow Q_C$$

$$i = |s| \land res = \sum_{j=0}^{|s|-1} s[j] \longrightarrow i = |s| \land res = \sum_{j=0}^{|s|-1} s[j]$$

Esto es trivialmente cierto, puesto que ambos términos son iguales. □

Demostración. Teorema de terminación de un ciclo. Ítem 1

Sea $fv = |s| - i$

$$\{I \land B \land fv = v_0\}S_{5\to 6}\{fv < v_0\}$$

$$I \land B \land |s| - i = v_0 \longrightarrow wp(res := res + s[i], wp(i := i + 1, |s| - i < v_0))$$

$$I \land B \land |s| - i = v_0 \longrightarrow |s| - i - 1 < v_0$$

$$I \land B \land |s| - i = v_0 \longrightarrow |s| - i - 1 < |s| - i$$

$$I \land B \land |s| - i = v_0 \longrightarrow True$$

□

Demostración. Teorema de terminación de un ciclo. Ítem 2

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

$$I \wedge |s| \leq i \longrightarrow |s| \leq i$$

$$I \wedge |s| \leq i \longrightarrow True$$

$\square$

3.3. obtenerSenadoresProvincia

Para este ejercicio decidimos implementar dos ciclos para facilitar la demostración.
Tenemos que demostrar:

- Código inicial: $Pre \longrightarrow wp(S_{1\to 2}, P_{C_1})$
- Ciclo 1:
  1. $P_{C_1} \longrightarrow I_1$
  2. $\{I_1 \wedge B_1\}S_{5\to 10}\{I_1\}$
  3. $I_1 \wedge \neg B_1 \longrightarrow Q_{C_1}$
  4. $\{I_1 \wedge B_1 \wedge fv_1 = v_0\}S_{5\to 10}\{fv_1 < v_0\}$
  5. $I_1 \wedge fv_1 \leq 0 \longrightarrow \neg B_1$
- Código intermedio: $\{Q_{C_1}\}S_{13\to 17}\{P_{C_2}\}$
- Ciclo 2:
  1. $P_{C_2} \longrightarrow I_2$
  2. $\{I_2 \wedge B_2\}S_{21\to 25}\{I_2\}$
  3. $I_2 \wedge \neg B_2 \longrightarrow Q_{C_2}$
  4. $\{I_2 \wedge B_2 \wedge fv_2 = v_0\}S_{21\to 25}\{fv_2 < v_0\}$
  5. $I_2 \wedge fv_2 \leq 0 \longrightarrow \neg B_2$
- Código final: $\{Q_{C_2}\}S_{28}\{Post\}$

Predicados: $P_{C_1} \equiv i = 0 \wedge max = 0 \wedge sonTodosDistintos(s)$
$Q_{C_1} \equiv i = |s| \wedge esPrimero(max, s) \wedge sonTodosDistintos(s)$
$I_1 \equiv 0 \leq i \leq |s| \wedge 0 \leq max < |s| \wedge (\forall k : \mathbb{Z})(0 \leq k < i \wedge k \neq max \longrightarrow s[max] > s[k]) \wedge sonTodosDistintos(s)$
$B_1 \equiv i < |s|$
$P_{C2} \equiv j = 0 \wedge 0 \leq snd \leq 1 \wedge snd \neq max \wedge esPrimero(max, s) \wedge sonTodosDistintos(s) \wedge_L s[max] > s[snd]$
$Q_{C_2} \equiv j = |s| \wedge max \neq snd \wedge esPrimero(max, s) \wedge_L esSegundo(max, snd, s) \wedge sonTodosDistintos(s)$
$I_2 \equiv 0 \leq j \leq |s| \wedge 0 \leq snd < |s| \wedge max \neq snd \wedge esPrimero(max, s) \wedge (\forall k : \mathbb{Z})(0 \leq k < j \wedge k \neq max \wedge k \neq snd \longrightarrow s[snd] > s[k]) \wedge sonTodosDistintos(s)$
$B_2 \equiv j < |s|$

Notación: Para simplificar la notación vamos a llamar:
Sea S_1 las primeras líneas de código antes del primer while.
Sea S_2 el código dentro del primer while.
Sea S_3 el código entre el primer while y el segundo while
Sea S_4 el código dentro del segundo while.
Sea S_5 el código después del segundo while.

Demostración. Código inicial

$$Pre \longrightarrow wp(S_{1\longrightarrow 2}, P_{C_1})$$

$$Pre \longrightarrow 0 = 0 \land 0 = 0 \land sonTodosDistintos(s)$$

$$Pre \longrightarrow sonTodosDistintos(s)$$

Puesto que $sonTodosDistintos(s)$ está en la Pre, esto es cierto.

□

Demostración. Ciclo 1: Ítem 1

$$P_{C_1} \longrightarrow I_1$$

$$P_{C_1} \longrightarrow 0 \leq 0 \leq |s| \land 0 \leq 0 < |s| \land (\forall k : \mathbb{Z})(0 \leq k < 0 \land k \neq max \longrightarrow s[max] > s[j]) \land sonTodosDistintos(s)$$

$$P_{C_1} \longrightarrow sonTodosDistintos(s)$$

Puesto que $sonTodosDistintos(s)$ está en P_{C_1}, esto es cierto. □

Demostración. Ciclo 1: Ítem 2
$\{I_1 \land B_1\}S_{5\to 10}\{I_1\}$, siendo S_2 el código del cuerpo del while.

$$I_1 \land B_1 \longrightarrow wp(S_{5\to 10}, I_1)$$

Por simplicidad, sea $F \equiv s[i] > s[max]$. También omitimos el término de $sonTodosDistintos(s)$, puesto que no cambia al asignarle nuevos valores en el if, y además es directamente implicado por el antecedente.

Sea $E_1 \equiv 0 \leq i+1 \leq |s| \land 0 \leq max < |s| \land (\forall k : \mathbb{Z})(0 \leq k < i+1 \land k \neq i \longrightarrow s[i] > s[k]))$

Sea $E_2 \equiv 0 \leq i+1 \leq |s| \land 0 \leq max < |s| \land (\forall k : \mathbb{Z})(0 \leq k < i+1 \land k \neq max \longrightarrow s[max] > s[k]))$

$$I_1 \land B_1 \longrightarrow (F \land E_1) \lor (\neg F \land E_2)$$

Analizamos $F \land E_1$. El primer término después de F es implicado por el rango $0 \leq i < |s|$ del precedente, lo podemos eliminar. Lo mismo aplica para el segundo término (el rango de max), que está en el antecedente.

$$(F \land E_1) \equiv s[i] > s[max] \land (\forall k : \mathbb{Z})(0 \leq k < i \longrightarrow s[i] > s[k]))$$

Luego, sabemos que vale el término del $\forall$ porque vale F y porque en el antecedente nos dicen que vale
$(\forall k : \mathbb{Z})(0 \leq k < i \land k \neq max \longrightarrow s[max] > s[k])$. Por lo tanto, $s[i] > s[k]$ para todo k entre 0 e i.

$$(F \land E_1) \equiv s[i] > s[max]$$

Analizamos $\neg F \land E_2$. El primer término después de $\neg F$ es implicado por el rango $0 \leq i < |s|$ del precedente, lo podemos eliminar. Lo mismo aplica para el segundo término (el rango de max), que está en el antecedente.

$$(\neg F \land E_2) \equiv s[i] \leq s[max] \land (\forall k : \mathbb{Z})(0 \leq k \leq i \land k \neq max \longrightarrow s[max] > s[k]))$$

.

Luego, sabemos que vale el término del $\forall$ por el antecedente, excepto cuando $k = i$. Para este caso, usamos que $s[i] \leq s[max]$. Osea que $s[max] \geq s[k]$. Como sabemos que $k \neq max$ y que $sonTodosDistintos(s)$, concluimos que $s[max] > s[k]$.

$$(\neg F \land E_2) \equiv s[i] \leq s[max]$$

Luego, armamos toda la estructura de la implicación de este modo.

$$I_1 \land B_1 \longrightarrow (F \land E_1) \lor (\neg F \land E_2)$$

$$I_1 \land B_1 \longrightarrow (s[i] > s[max]) \lor (s[i] \leq s[max]$$

$$I_1 \land B_1 \longrightarrow True$$

□

Demostración. Ciclo 1: Ítem 3

$$I_1 \wedge \neg B_1 \longrightarrow Q_{C_1}$$

$$I_1 \wedge \neg B_1 \longrightarrow i = |s| \wedge esPrimero(max, s) \wedge sonTodosDistintos(s)$$

Sabemos que $I_1 \wedge \neg B_1$ implica $i = |s|$ y que $sonTodosDistintos(s)$ está en I_1.

$$I_1 \wedge \neg B_1 \longrightarrow 0 \leq max < |s| \wedge (\forall k : \mathbb{Z})(0 \leq k < |s| \wedge k \neq max \longrightarrow s[max] > s[k])$$

El término de $0 \leq max < |s|$ es implicado directamente por I_1. Además, el término del $\forall$ es igual al del precedente, puesto que $i = |s|$

$$I_1 \wedge \neg B_1 \longrightarrow True$$

□

Demostración. Ciclo 1: Ítem 4

$$\{I_1 \wedge B_1 \wedge fv_1 = v_0\}S_{5\to 10}\{fv_1 < v_0\}$$

Sea $fv_1 = |s| - i$

$$I_1 \wedge B_1 \wedge |s| - i = v_0 \longrightarrow wp(S_{5\longrightarrow 10}, |s| - i < v_0)$$

$$I_1 \wedge B_1 \wedge |s| - i = v_0 \longrightarrow (F \wedge |s| - i - 1 < v_0) \vee (\neg F \wedge |s| - i - 1 < v_0)$$

$$I_1 \wedge B_1 \wedge |s| - i = v_0 \longrightarrow |s| - i - 1 < v_0$$

Reemplazamos $v_0 = |s| - i$ en el consecuente.

$$I_1 \wedge B_1 \wedge |s| - i = v_0 \longrightarrow |s| - i - 1 < |s| - i - 1$$

$$I_1 \wedge B_1 \wedge |s| - i = v_0 \longrightarrow True$$

□

Demostración. Ciclo 1: Ítem 5

$$I_1 \wedge fv_1 \leq 0 \longrightarrow \neg B_1$$

Sea $fv_1 = |s| - i$

$$I_1 \wedge |s| \leq i \longrightarrow i \geq |s|$$

Luego, podemos deducir del antecedente que $i = |s|$. Por ende, la implicación es cierta.

□

Demostración. Código intermedio

$$\{Q_{C_1}\}S_{13\to 17}\{P_{C_2}\}$$

Recordemos que: $P_{C2} \equiv j = 0 \wedge 0 \leq snd \leq 1 \wedge snd \neq max \wedge esPrimero(max, s) \wedge sonTodosDistintos(s) \wedge_L s[max] > s[snd]$

$$Q_{C_1} \longrightarrow wp(S_{13\to 17}, P_{C_2})$$

Omitimos los términos de $sonTodosDistintos(s)$ porque son implicados por el antecedente. También los términos de $j = 0$, $0 \leq snd \leq 1$ y $snd \neq max$ puesto que al hacer la sustitución dan $True$.

$$Q_{C_1} \longrightarrow (max = 0 \wedge esPrimero(max, s) \wedge s[max] > s[snd]) \vee (max \neq 0 \wedge esPrimero(max, s) \wedge s[max] > s[snd])$$

Luego, sabemos que $s[max] > s[snd]$ pues $esPrimero(max, s)$ está en Q_{C_1} y $esSegundo(max, snd, s)$ también.

$$Q_{C_1} \longrightarrow (max = 0 \wedge esPrimero(max, s)) \vee (max \neq 0 \wedge esPrimero(max, s))$$

$$Q_{C_1} \longrightarrow esPrimero(max, s)$$

Lo cual es cierto pues $esPrimero(max, s)$ está en Q_{C_1}

□

Demostración. Ciclo 2: Ítem 1

Recordemos que:

$P_{C_2} \equiv j = 0 \land 0 \leq snd \leq 1 \land snd \neq max \land esPrimero(max, s) \land sonTodosDistintos(s) \land_L s[max] > s[snd]$

$I_2 \equiv 0 \leq j \leq |s| \land 0 \leq snd < |s| \land max \neq snd \land esPrimero(max, s) \land (\forall k : \mathbb{Z})(0 \leq k < j \land k \neq max \land k \neq snd \longrightarrow s[snd] > s[k]) \land sonTodosDistintos(s)$

$$P_{C_2} \longrightarrow I_2$$

Puesto que j=0, puedo reemplazar y eliminar algunos términos. También eliminamos $sonTodosDistintos(s)$ pues pertenece a P_{C_2}, al igual que $esPrimero(max, s)$

$$P_{C_2} \longrightarrow 0 \leq snd < |s| \land max \neq snd \land (\forall k : \mathbb{Z})(0 \leq k < 0 \land k \neq max \land k \neq snd \longrightarrow s[snd] > s[k])$$

Además, sabemos que es cierto el $\forall$, pues ningún k da $True$ en el antecedente, haciendo que todo el término sea $True$.

$$P_{C_2} \longrightarrow 0 \leq snd < |s| \land max \neq snd$$

Podemos eliminar el último término pues aparece en el antecedente. Además, el primer término es implicado por $0 \leq snd \leq 1$, que aparece en P_{C_2}.

$$P_{C_2} \longrightarrow True$$

□

Demostración. Ciclo 2: Ítem 2

$$\{I_2 \land B_2\} S_{21 \to 25} \{I_2\}$$

$$I_2 \land B_2 \longrightarrow wp(S_{21 \to 25}, I_2)$$

$$I_2 \land B_2 \longrightarrow wp(S_{21 \to 25}, I_2)$$

Para simplificar la demostración, vamos a llamar a las siguientes variables.

$D \equiv j \neq max \land s[j] > s[snd]$

$E_1 \equiv wp(snd := j, I^j_{j+1})$

$E_2 \equiv wp(skip, I^j_{j+1}) \equiv I^j_{j+1}$

Luego, la implicación queda de este modo.

$$I_2 \land B_2 \longrightarrow (D \land E_1) \lor (\neg D \land E_2)$$

Primero voy a reducir la primera expresión, $(D \land E_1)$, asumiendo que vale el antecedente.

$(D \land E_1) \equiv D \land 0 \leq j + 1 \leq |s| \land 0 \leq j < |s| \land max \neq j \land esPrimero(max, s) \land$
$(\forall k : \mathbb{Z})(0 \leq k < j + 1 \land k \neq max \land k \neq j \longrightarrow s[j] > s[k]) \land sonTodosDistintos(s)$

Podemos eliminar $sonTodosDistintos(s)$ y $esPrimero(max, s)$, al igual que ambos rangos de j, puesto que son trivialmente implicados por el antecedente.

$$D \land max \neq j \land (\forall k : \mathbb{Z})(0 \leq k < j + 1 \land k \neq max \land k \neq j \longrightarrow s[j] > s[k])$$

Podemos sacar $max \neq j$ pues está en D. Reemplazamos D por su expresión completa.

$$j \neq max \land s[j] > s[snd] \land (\forall k : \mathbb{Z})(0 \leq k < j \land k \neq maximplicas[j] > s[k])$$

Luego, como sabemos que $s[j] > s[snd]$ y que $s[snd] > s[k]$ para todo $k \in (0, j)$, por I_2.

Entonces, $D \land E_1 \equiv D$, tomando en cuenta que vale el antecedente.

Ahora, analizamos $\neg D \land E_2$

$$\neg D \land E_2 \equiv \neg D \land I^j_{j+1}$$

Podemos eliminar algunos términos triviales como $esPrimero(max, s)$ y $sonTodosDistintos(s)$, ya que son implicados por el invariante.

$$\neg D \land E_2 \equiv \neg D \land 0 \leq j+1 \leq |s| \land 0 \leq snd < |s| \land max \neq snd \land (\forall k : \mathbb{Z})(0 \leq k < j+1 \land k \neq max \land k \neq snd \longrightarrow s[snd] > s[k])$$

Luego, eliminamos los rangos de j y de snd, al igual que $max \neq snd$.

$$(j = max \lor s[j] \leq s[snd]) \land (\forall k : \mathbb{Z})(0 \leq k < j + 1 \land k \neq max \land k \neq snd \longrightarrow s[snd] > s[k])$$

El término del $\forall$ está implicado por el invariante, excepto por el caso $k = j$. Para este caso, sé que $k \neq max$ por la condición del $\forall$.

Luego, podemos usar que $s[snd] \geq s[j]$, ya que sabemos que $j = max$ no vale en este caso particular, por ende, tiene que valer la otra parte del OR.

Luego, nos quedaría probar que es mayor estricto, cosa que podemos probar puesto que vale que $sonTodosDistintos(s)$ y que $j \neq snd$ por la condición del $\forall$.

Por ende, quedaría de este modo, equivalente a $\neg D$.

$$(j = max \lor s[j] \leq s[snd])$$

Volvemos a la implicación original.

$$I_2 \land B_2 \longrightarrow D \lor \neg D$$

$$I_2 \land B_2 \longrightarrow True.$$

□

Demostración. Ciclo 2: Ítem 3

Recordemos que $I_2 \equiv 0 \leq j \leq |s| \land 0 \leq snd < |s| \land max \neq snd \land esPrimero(max, s) \land (\forall k : \mathbb{Z})(0 \leq k < j \land k \neq max \land k \neq snd \longrightarrow s[snd] > s[k]) \land sonTodosDistintos(s)$

$$I_2 \land \neg B_2 \longrightarrow Q_{C2}$$

$$I_2 \land \neg B_2 \longrightarrow j = |s| \land max \neq snd \land esPrimero(max, s) \land_L esSegundo(max, snd, s) \land sonTodosDistintos(s)$$

Podemos eliminar $esPrimero(max, s)$, $sonTodosDistintos(s)$ y $max \neq snd$ pues están en I_2. También podemos eliminar $j = |s|$ pues está en $I_2 \land B_2$.

$$I_2 \land B_2 \longrightarrow 0 \leq snd < |s| \land (\forall k : \mathbb{Z})(0 \leq k < |s| \land k \neq max \land k \neq snd \longrightarrow s[snd] > s[k])$$

Podemos eliminar $0 \leq snd < |s|$ pues está en I_2.

$$I_2 \land B_2 \longrightarrow (\forall k : \mathbb{Z})(0 \leq k < |s| \land k \neq max \land k \neq snd \rightarrow_L s[snd] > s[k])))$$

Finalmente, este último término es implicado por el Invariante, puesto que $j = |s|$, de este modo queda exactamente igual que el término del invariante.

□

Demostración. Ciclo 2: Ítem 4

Sea $fv_2 = |s| - j$

$\{I_2 \land B_2 \land fv_2 = v_0\} S_{21 \rightarrow 25} \{fv_2 < v_0\}$

$$I_2 \land B_2 \land v_0 = |s| - j \longrightarrow wp(S_{21 \rightarrow 25}, |s| - j < v_0)$$

$$I_2 \land B_2 \land v_0 = |s| - j \longrightarrow wp(S_{21 \rightarrow 25}, |s| - j < v_0)$$

Tomamos $F \equiv j \neq max \land s[j] > s[snd]$

$$I_2 \land B_2 \land v_0 = |s| - j \longrightarrow (F \land wp(j := j + 1, |s| - j < v_0)) \lor (\neg F \land wp(j := j + 1, |s| - j < v_0))$$

$$I_2 \land B_2 \land v_0 = |s| - j \longrightarrow (F \land |s| - j - 1 < v_0) \lor (\neg F \land |s| - j - 1 < v_0)$$

$$I_2 \land B_2 \land |s| - j = v_0 \longrightarrow |s| - j - 1 < |s| - j$$

$$I_2 \land B_2 \land |s| - j = v_0 \longrightarrow True$$

□

Demostración. Ciclo 2: Ítem 5

$$I_2 \land fv_2 \leq 0 \longrightarrow \neg B_2$$

$$I_2 \land |s| \leq j \longrightarrow j \geq |s|$$

Lo cual es trivialmente cierto.

□

Demostración. Código final

Tengamos en cuenta que

$Post \equiv res = (max, snd) \iff max \neq snd \wedge_L esPrimero(max, s) \wedge_L esSegundo(max, snd, s)$

$Q_{C_2} \equiv j = |s| \wedge max \neq snd \wedge esPrimero(max, s) \wedge_L esSegundo(max, snd, s) \wedge sonTodosDistintos(s)$

$$\{Q_{C_2}\}S_{28}\{Post\}$$

$$Q_{C_2} \longrightarrow wp(S_{28}, Post)$$

$$Q_{C_2} \longrightarrow (max, snd) = (max, snd) \iff max \neq snd \wedge_L esPrimero(max, s) \wedge_L esSegundo(max, snd, s)$$

$$Q_{C_2} \longrightarrow max \neq snd \wedge_L esPrimero(max, s) \wedge_L esSegundo(max, snd, s)$$

Luego, podemos eliminar todos los términos del consecuente pues aparecen en el antecedente.

$$Q_{C_2} \longrightarrow True$$

$\square$