Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Teorema del invariante: corrección de ciclos

Ejercicio 1. ★ Consideremos el problema de sumar los elementos de un arreglo y la siguiente implementación en SmallLang, con el invariante del ciclo.

Especificación

proc sumar (in s: $seq\langle\mathbb{Z}\rangle$, out result: $\mathbb{Z}$) {
  Pre {true}
  Post $\{result = \sum_{j=0}^{|s|-1} s[j]\}$
}

Implementación en SmallLang

```
result := 0;
i := 0;
while (i < s.size()) do
  result := result + s[i];
  i := i + 1
endwhile
```

Invariante de Ciclo

$$I \equiv 0 \leq i \leq |s| \wedge_L result = \sum_{j=0}^{i-1} s[j]$$

a) Escribir la precondición y la postcondición del ciclo.

b) ¿Qué punto falla en la demostración de corrección si el primer término del invariante se reemplaza por $0 \leq i < |s|$?

c) ¿Qué punto falla en la demostración de corrección si el límite superior de la sumatoria $(i-1)$ se reemplaza por i?

d) ¿Qué punto falla en la demostración de corrección si se invierte el orden de las dos instrucciones del cuerpo del ciclo?

e) Demostrar formalmente la corrección parcial del ciclo, usando los primeros puntos del teorema del invariante.

f) Proponer una función variante y demostrar formalmente la terminación del ciclo, utilizando la función variante.

Ejercicio 2. ★ Dadas la especificación y la implementación del problema `sumarParesHastaN`, escribir la precondición y la postcondición del ciclo, y demostrar formalmente su corrección a través del teorema del invariante.

Especificación

proc sumarParesHastaN (in n: $\mathbb{Z}$, out result: $\mathbb{Z}$) {
  Pre $\{n \geq 0\}$
  Post $\{result = \sum_{j=0}^{n-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})\}$
}

Implementación en SmallLang

```
result := 0;
i := 0;
while (i < n) do
  result := result + i;
  i := i + 2
endwhile
```

Invariante de ciclo

$$I \equiv 0 \leq i \leq n+1 \wedge i \bmod 2 = 0 \wedge result = \sum_{j=0}^{i-1} (\text{if } j \bmod 2 = 0 \text{ then } j \text{ else } 0 \text{ fi})$$

Ejercicio 3. Supongamos que se desea implementar la función `exponenciacion`, cuya especificación es la siguiente:

proc `exponenciacion` (in m: $\mathbb{Z}$, in n: $\mathbb{Z}$, out result: $\mathbb{Z}$) {
    Pre $\{n \geq 0 \wedge \neg(m = 0 \wedge n = 0)\}$
    Post $\{result = m^n\}$
}

Consideremos además el siguiente invariante: $I \equiv 0 \leq i \leq n \wedge result = m^i$

a) Escribir un programa en SmallLang que resuelva este problema, y que incluya un ciclo que tenga a I como invariante. Demostrar formalmente la corrección de este ciclo.

b) La siguiente implementación en SmallLang es trivialmente errónea. ¿Qué punto del teorema del invariante falla en este caso?[1]

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

while( i < m ) do
  result := result * n;
  i := i + 1
endwhile
```

c) ¿Qué puede decir de la siguiente implementación en SmallLang? En caso de que sea correcta, proporcione una demostración. En caso de que sea incorrecta, explique qué punto del teorema del invariante falla.

```
i := 0;
result := 1;
while( i < n ) do
  i := i + 1;
  result := result * m
endwhile
```

d) ¿Qué puede decir de la siguiente implementación? En caso de que sea incorrecta, ¿se puede reforzar la precondición del problema para que esta especificación pase a ser correcta?

```
i := 2;
result := m*m;

while( i < n ) do
  result := result * m;
  i := i + 1
endwhile
```

Ejercicio 4. ★ Considere el problema `sumaDivisores`, dado por la siguiente especificación:

proc `sumaDivisores` (in n: $\mathbb{Z}$, out result: $\mathbb{Z}$) {
    Pre $\{n \geq 1\}$
    Post $\{result = \sum_{j=1}^{n}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi})\}$
}

a) Escribir un programa en SmallLang que satisfaga la especificación del problema y que contenga exactamente un ciclo.

b) El ciclo del programa propuesto, ¿puede ser demostrado mediante el siguiente invariante?

$$I \equiv 1 \leq i \leq n \wedge result = \sum_{j=1}^{i}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi})$$

Si no puede, ¿qué cambios se le deben hacer al invariante para que se corresponda con el ciclo propuesto?

[1] Recordar que para mostrar que una implicación $A \rightarrow B$ no es cierta alcanza con dar valores de las variables libres que hagan que A sea verdadero y que B sea falso. Para mostrar que una tripla de Hoare $\{P\}S\{Q\}$ no es válida, alcanza con dar valores de las variables que satisfacen P, y tales que luego de ejecutar S, el estado final no satisface Q.

Ejercicio 5. Considere la siguiente especificación de la función `sumarPosicionesImpares`.

proc `sumarPosicionesImpares` (in s: $seq\langle\mathbb{Z}\rangle$, out result: $\mathbb{Z}$) {
  Pre {true}
  Post $\{result = \sum_{i=0}^{|s|-1}(\text{if } i \bmod 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})\}$
}

a) Implementar un programa en SmallLang para resolver este problema, que incluya exactamente un ciclo con el siguiente invariante:
$I \equiv 0 \leq j \leq |s| \wedge_L result = \sum_{i=0}^{j-1}(\text{if } i \bmod 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})$

b) Demostrar formalmente la corrección del ciclo propuesto.

Ejercicio 6. Considere la siguiente especificación e implementación del problema `maximo`.

Especificación

proc `maximo` (in s: $seq\langle\mathbb{Z}\rangle$, out i: $\mathbb{Z}$) {
  Pre $\{|s| \geq 1\}$
  Post $\{0 \leq i < |s| \wedge_L$
  $(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \leq s[i])\}$
}

Implementación en SmallLang

```
i := 0;
j := 1;
while (j < s.size()) do
  if (s[j] > s[i])
    i := j
  else
    skip
  endif;
  j:=j+1
endwhile
```

a) Escribir la precondición y la postcondición del ciclo.

b) Demostrar que el ciclo es parcialmente correcto, utilizando el siguiente invariante:

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

c) Proponer una función variante que permita demostrar que el ciclo termina.

Ejercicio 7. ★ Considere la siguiente especificación e implementación del problema `copiarSecuencia`.

Especificación

proc `copiarSecuencia` (in s: $seq\langle\mathbb{Z}\rangle$, inout r: $seq\langle\mathbb{Z}\rangle$) {
  Pre $\{|s| = |r| \wedge r = r_0\}$
  Post $\{|s| = |r| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = r[j])\}$
}

Implementación en SmallLang

```
i := 0;
while (i < s.size()) do
  r[i]:=s[i];
  i:=i+1
endwhile
```

a) Escribir la precondición y la postcondición del ciclo.

b) Proponer un invariante y demostrar que el ciclo es parcialmente correcto.

c) Proponer una función variante que permita demostrar que el ciclo termina.

Ejercicio 8. Considere la siguiente especificación e implementación del problema `llenarSecuencia`.

Especificación

proc `llenarSecuencia` (inout s: $seq\langle\mathbb{Z}\rangle$, in d: $\mathbb{Z}$, in e: $\mathbb{Z}$) {
  Pre $\{d \geq 0 \wedge d < |s| \wedge s = S_0\}$
  Post $\{|s| = |S_0| \wedge_L$
  $(\forall j : \mathbb{Z})(0 \leq j < d \rightarrow_L s[j] = S_0[j]) \wedge$
  $(\forall j : \mathbb{Z})(d \leq j < |s| \rightarrow_L s[j] = e)\}$
}

Implementación en SmallLang

```
i := d;
while (i < s.size()) do
  s[i]:=e;
  i:=i+1
endwhile
```

a) Escribir la precondición y la postcondición del ciclo.

b) Proponer un invariante y demostrar que el ciclo es parcialmente correcto.

c) Proponer una función variante que permita demostrar que el ciclo termina.

Ejercicio 9. ★ Sea el siguiente ciclo con su correspondiente precondición y postcondición:

```
while (i >= length(s) / 2) do
  suma := suma + s[length(s)-1-i];
  i := i - 1
endwhile
```

$$P_c : \{|s| \ mod \ 2 = 0 \wedge i = |s| - 1 \wedge suma = 0\}$$

$$Q_c : \{|s| \ mod \ 2 = 0 \wedge i = |s|/2 - 1 \ \wedge_L \ suma = \sum_{j=0}^{|s|/2-1} s[j]\}$$

a) Especificar un invariante de ciclo que permita demostrar que el ciclo cumple la postcondición.

b) Especificar una función variante que permita demostrar que el ciclo termina.

c) Demostrar formalmente la corrección y terminación del ciclo usando el Teorema del invariante.

Ejercicio 10. Considere la siguiente especificación del problema `reemplazarTodos`.

proc `reemplazarTodos` (inout s: $seq\langle\mathbb{Z}\rangle$, in a: $\mathbb{Z}$, in b: $\mathbb{Z}$) {
  Pre $\{s = S_0\}$
  Post $\{|s| = |S_0| \wedge_L$
  $(\forall j : \mathbb{Z})((0 \leq j < |s| \wedge_L S_0[j] = a) \rightarrow_L s[j] = b) \wedge$
  $(\forall j : \mathbb{Z})(d \leq j < |s| \wedge_L S_0[j] \neq a) \rightarrow_L s[j] = S_0[j])\}$
}

a) Dar un programa en SmallLang que implemente la especificación dada.

b) Escribir la precondición y la postcondición del ciclo.

c) Proponer un invariante y demostrar que el ciclo es parcialmente correcto.

d) Proponer una función variante que permita demostrar que el ciclo termina.

Demostración de correctitud: programas completos

Ejercicio 11. ★ Demostrar que el siguiente programa es **correcto** respecto a la especificación dada.

Especificación

proc `indice` (in s: $seq\langle\mathbb{Z}\rangle$, in e: $\mathbb{Z}$, out r: $\mathbb{Z}$) {
  Pre $\{True\}$
  Post $\{r = -1 \rightarrow$
  $(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \neq e)$
  $\wedge$
  $r \neq -1 \rightarrow$
  $(0 \leq r < |s| \wedge_L s[r] = e)\}$
}

Implementación en SmallLang

```
i := s.size()-1;
j := -1;
while (i>=0) do
  if (s[i]=e) then
     j:=i
  else
     skip
  endif;
  i:=i-1
endwhile;
r := j;
```

Ejercicio 12. ★ Demostrar que el siguiente programa es correcto respecto a la especificación dada.

Especificación

`proc existeElemento` (in s: $seq\langle\mathbb{Z}\rangle$, in e: $\mathbb{Z}$, out r: Bool) {
Pre $\{True\}$
Post $\{r = True \leftrightarrow ((\exists k : \mathbb{Z})(0 \leq k < |s|) \wedge_L s[k] = e)\}$
}

Implementación en SmallLang

```
i := 0;
j := -1;
while (i < s.size()) do
   if (s[i] = e) then
      j := i
  else
     skip
  endif;
  i := i + 1
endwhile;
if (j != -1)
   r := true
else
   r := false
endif
```

Ejercicio 13. Demostrar que el siguiente programa es correcto respecto a la especificación dada.

Especificación

`proc esSimetrico` (in s: $seq\langle\mathbb{Z}\rangle$, out r:Bool) {
Pre $\{True\}$
Post $\{r = True \leftrightarrow (\forall i : \mathbb{Z})(0 \leq i < |s| \rightarrow_L s[i] = s[|s| - (i+1)])\}$
}

Implementación en SmallLang

```
i := 0;
j := s.size() - 1;
r := true;
while (i < s.size()) do
   if (s[i]!=s[j]) then
      r := false
   else
      skip
   endif;
   i := i + 1;
   j := j - 1;
endwhile
```

Ejercicio 14. ★ Demostrar que el siguiente programa es correcto respecto a la especificación dada.

Especificación

`proc concatenarSecuencias` (in a: $seq\langle\mathbb{Z}\rangle$,
in b: $seq\langle\mathbb{Z}\rangle$,
inout r:$seq\langle\mathbb{Z}\rangle$) {
Pre $\{|r| = |a| + |b| \wedge r = R_0\}$
Post $\{|r| = |R_0| \wedge (\forall j : \mathbb{Z})(0 \leq j < |a| \rightarrow_L r[j] = a[j]) \wedge (\forall j : \mathbb{Z})(0 \leq j < |b| \rightarrow_L r[j + |a|] = b[j])\}$
}

Implementación en SmallLang

```
i := 0;
while (i < a.size()) do
   r[i]:=a[i];
   i:=i+1
endwhile;
i:= 0;
while (i < b.size()) do
   r[a.size()+i]:=b[i];
   i:=i+1
endwhile
```

Ejercicio 15. Dar dos programas en SmallLang que satisfagan la siguiente especificación, y demostrar que ambos son correctos.

`proc buscarPosicionUltimoMaximo` (in s: $seq\langle\mathbb{Z}\rangle$, out r:$\mathbb{Z}$) {
Pre $\{|s| > 0\}$
Post $\{0 \leq r < |s| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < r \rightarrow_L s[r] \geq s[j]) \wedge (\forall j : \mathbb{Z})(r < j < |s| \rightarrow_L s[r] > s[j])\}$
}

RESOLUCIONES.

Ejercicio 1

a)

$$P_c : \{result = 0 \wedge i = 0\}$$
$$Q_c : \{result = \sum_{j=0}^{|s|-1} s[j]\}$$

b) Si le sacas el igual a la segunda desigualdad no te agarra el ultimo elemento la sumatoria.

c) Si le sacas el -1 al supraindice de la sumatoria entonces va a tratar de sumar la posicion del tamaño de la secuencia, como la secuencia no posee esta posicion se va a indefinir el resultado.

d) Si invertis de lugar las instrucciones del cuerpo del ciclo entonces no te agarra el primer elemento de la secuencia.

e) Vamos a demostrar los primeros 3 puntos del teorema del invariante. Esto nos permite afirmar que el ciclo, si termina entonces es correcto respecto a su especificacón. Especificacion del ciclo:

- $P_c : result = 0 \wedge i = 0$
- $Q_c : result = \sum_{j=0}^{|s|-1} s[j]$
- $I \equiv 0 \leq i \leq |s| \wedge_L result = \sum_{j=0}^{i-1} s[j]$

$P_c \Rightarrow I$

Tenemos que vale P_c como hipótesis. Queremos probar que vale I. Vamos a probarlo por partes:

- Queremos ver que vale $0 \leq i \leq |s|$

  Sabemos que i=0 (es información que nos da P_c). Entonces reemplazando, lo que queremos ver es que $0 \leq 0 \leq |s|$. La primera parte, $0 \leq 0$, es una tautología. Nos resta ver si $0 \leq |s|$. Pero si la lista es vacia entonces $0 \leq 0$, caso contrario si la lista fuera no vacia entonces $|s| > 0$ y la desigualdad $0 \leq |s|$ se cumple.
- Queremos ver que $result = \sum_{j=0}^{i-1} s[j]$

  Como $i = 0$, entonces $\sum_{j=0}^{i-1} s[j] = 0 = result$.

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que valen tanto I como $\neg B$.
Es decir, queremos probar que $result = \sum_{j=0}^{|s|-1} s[j]$
Como sabemos que vale I, podemos afirmar que $0 \leq i \leq |s|$.
Además sabemos que vale $\neg B \equiv i \geq |s|$.
Luego $|s| \leq i \leq |s|$, entonces el unico valor que cumple esta condición es $i = |s|$. Analicemoslo.

- $i = |s|$

En este caso, podemos reemplazar i por $|s|$ en la sumatoria del invariante y llegamos a $result = \sum_{j=0}^{|s|-1} s[j]$, exactamante lo que queríamos probar.

$\{I \wedge B\}$ ciclo $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: $result := result + s[i]$;
S2: $i := i + 1$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{1}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$wp(S2, I) \overset{Ax1}{\equiv} def(i+1) \wedge_L I^i_{i+1}$$

$$true \wedge_L (0 \leq i+1 \leq |s| \wedge_L result = \sum_{j=0}^{i+1-1} s[j])$$

$$(0 \leq i+1 \leq |s| \wedge_L result = \sum_{j=0}^{i+1-1} s[j])$$

Volviendo a (1), reemplazamos $wp(S2, I)$ y nos queda:

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \equiv wp(S1, 0 \leq i+1 \leq |s| \wedge_L result = \sum_{j=0}^{i+1-1} s[j])$$

$$\overset{Ax1}{\equiv} def(result + s[i]) \wedge_L 0 \leq i+1 \leq |s| \wedge_L result + s[i] = \sum_{j=0}^{i} s[j]$$

$$\equiv 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge_L result + s[i] = \sum_{j=0}^{i} s[j]$$

$$\equiv 0 \leq i < |s| \wedge_L result = \sum_{j=0}^{i} s[j] - s[i]$$

$$\equiv 0 \leq i < |s| \wedge_L result = \sum_{j=0}^{i-1} s[j]$$

Una vez calculada la precondición más débil, debemos ver si $(I \wedge B)$ implican dicha precondición. Probamos cada parte por separado:

- $0 \leq i < |s|$

  Sabemos por I que $0 \leq i$, entonces la primera parte ya esta demostrada, ahora la interseccion entre la guarda y el invariante da como cota superior $i < |s|$, que es lo mismo que tengo en mi wp.

- $result = \sum_{j=0}^{i-1} s[j]$

  Este $result$ es igual al del I. Como $(I \wedge B) \Rightarrow wp(ciclo, I)$, podemos afirmar que el cuerpo del ciclo preserva el invariante.

f) Vamos a demostrar los ultimos 2 puntos del teorema del invariante. Esto nos permite afirmar que el ciclo termina. Especificación del ciclo:

- $f_v : |s| - i$

$\{(I \wedge B \wedge v_0 = f_v)\}$ **ciclo** $\{(f_v < v_0)\}$

Dado que queremos demostrar que vale una tripla de Hoare, comenzaremos calculando la precondición más débil $wp(ciclo, f_v < v_0)$.

$$wp(S1; S2, f_v < v_0) \overset{Ax3}{\equiv} wp(S1, wp(S2, |s| - i < v_0))$$

$$\overset{Ax1}{\equiv} wp(S1, true \wedge_L |s| - (i+1) < v_0)$$

$$\overset{Ax3}{\equiv} true \wedge_L (true \wedge_L |s| - (i+1) < v_0)$$

$$\equiv |s| - i - 1 < v_0$$

Es decir, $wp(S1; S2, f_v < v_0) = |s| - i - 1 < v_0$. Ahora debemos ver que $(I \wedge B \wedge v_0 = f_v)$ implican dicha WP. Parte de la hipótesis es que $v_0 = f_v$, es decir $v_0 = |s| - i$. Restando 1 a ambos lados, $|s| - 1 - i = v_0 - 1 < v_0$.

$(I \wedge f_v \leq 0) \Rightarrow \neg B$

Debemos mostrar que vale $\neg B$, es decir $i \geq |s|$.
Sabemos que $f_v \leq 0$, es decir $|s| - i \leq 0$, luego $|s| \leq i$, como queriamos demostrar.

Ejercicio 2 Vamos a demostrar los 5 puntos del teorema del invariante. Esto nos permite afirmar que el ciclo termina y es correcto respecto a su especificacón (no demuestra que el programa entero sea correcto!). Especificacion del ciclo:

- $P_c : n \geq 0 \wedge result = 0 \wedge i = 0$
- $Q_c : result = \sum_{j=0}^{n-1}$(if j mod 2=0 then j else 0 fi)
- $I \equiv 0 \leq i \leq n+1 \wedge i \text{ mod } 2 = 0 \wedge result = \sum_{j=0}^{i-1}$(if j mod 2=0 then j else 0 fi)
- $f_v : n - i$

$P_c \Rightarrow I$

Tenemos que vale P_c como hipótesis. Queremos probar que vale I. Vamos a probarlo por partes:

- Queremos ver que vale $0 \leq i \leq n+1$

  Sabemos que i=0 (es información que nos da P_c). Entonces reemplazando, lo que queremos ver es que $0 \leq 0 \leq n+1$. La primera parte, $0 \leq 0$, es una tautología. Nos resta ver si $0 \leq n+1$. Pero P_c indica también que $n \leq 0$, luego $n + i \leq 1 \leq 0$.

- Queremos ver que vale$i \text{ mod } 2 = 0$
  Sabemos que $i = 0$,luego podemos afirmar que $i \text{ mod } 2 = 0$.

- Queremos ver que $result = \sum_{j=0}^{i-1}$(if j mod 2=0 then j else 0 fi)
  Como $i = 0$, $result = \sum_{j=0}^{i-1}$(if j mod 2=0 then j else 0 fi)=$result = \sum_{j=0}^{-1}$(if j mod 2=0 then j else 0 fi). Recordemos que si el rango de una sumatoria es vacío (como en este caso), la sumatoria tiene valor 0. Luego $result = \sum_{j=0}^{i-1}$(if j mod 2=0 then j else 0 fi)=0. Pero además sabemos que result = 0 (por P_c), asi que podemos afirmar que 0=result=$result = \sum_{j=0}^{i-1}$(if j mod 2=0 then j else 0 fi)=0

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que valen tanto I como $\neg B$.
Es decir, queremos probar que $result = \sum_{j=0}^{n-1}$(if j mod 2=0 then j else 0 fi)
Como sabemos que vale I, podemos afirmar que $0 \leq i \leq n+1$.
Además sabemos que vale $\neg B \equiv i \geq n$.
Luego $n \leq i \leq n+1$. Hay dos valores de i que cumplen esa condición. Analicemos ambos casos:

- $i = n$

En este caso, podemos reemplazar i por n en la parte de la sumatoria del invariante y llegamos a $result = \sum_{j=0}^{n-1}$(if j mod 2=0 then j else 0 fi), exactamante lo que queríamos probar.

- $i = n+1$

En este caso, su hacemos el reemplazo, llegamos a $result = \sum_{j=0}^{n}$(if j mod 2=0 then j else 0 fi) (y esto no es a lo que queremos llegar!).
$result = \sum_{j=0}^{n}$(if j mod 2=0 then j else 0 fi) =$result = \sum_{j=0}^{n-1}$(if j mod 2=0 then j else 0 fi) +(if n mod 2=0 then j else 0 fi). Sabemos además que i mod 2=0 (información del invariante). Pero estamos en el caso en el cual $i = n+1$. Entonces podemos afirmar que si i es par, n es impar. Luego n no cumple la guada del IF y podemos afirmar que (if n mod 2=0 then j else 0 fi) = 0.
$result = \sum_{j=0}^{n}$(if j mod 2=0 then j else 0 fi) =$result = \sum_{j=0}^{n-1}$(if j mod 2=0 then j else 0 fi) +0.

$\{I \wedge B\}$ **ciclo** $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: $result := result + i$;
S2: $i := i + 2$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{2}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$wp(S2, I) \overset{Ax1}{\equiv} def(i+2) \wedge_L I^i_{i+2}$$
$$\equiv true \wedge_L (0 \leq i+2 \leq n+1 \wedge i+2 \bmod 2 = 0 \wedge_L result = \sum_{j=0}^{i+2-1} \text{(if j mod 2=0 then j else 0 fi)})$$
$$\equiv (0 \leq i+2 \leq n+1 \wedge i+2 \bmod 2 = 0 \wedge_L result = \sum_{j=0}^{i+1} \text{(if j mod 2=0 then j else 0 fi)})$$

Volviendo a (2), reemplazamos $wp(S2, I)$ y nos queda:

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \equiv wp(S1, 0 \leq i+2 \leq n+1 \wedge i+2 \bmod 2 = 0 \wedge result = \sum_{j=0}^{i+1} \text{(if j mod 2=0 then j else 0 fi)})$$
$$\overset{Ax1}{\equiv} def(result+i) \wedge_L (0 \leq i+2 \leq n+1 \wedge i+2 \bmod 2 = 0 \wedge result = \sum_{j=0}^{i+1} \text{(if j mod 2=0 then j else 0 fi)})$$
$$\equiv true \wedge_L (0 \leq i+2 \leq n+1 \wedge i+2 \bmod 2 = 0 \wedge result = \sum_{j=0}^{i+1} \text{(if j mod 2=0 then j else 0 fi)})$$

Una vez calculada la precondición más débil, debemos ver si $(I \wedge B)$ implican dicha precondición. Probamos cada parte por separado:

- $0 \leq i+2 \leq n+1$

  Sabemos por I que $i > 0$, luego podemos afirmar que $0 \leq i+2$
  Sabemos por B que $i < n$, luego (sumando 2 en ambos términos): $i+2 < n+2$, lo cual es equivalente a decir que $i+2 \leq n+1$

- $i+2$ mod 2=0

  Sabemos por I que i mod 2=0. Si i es par, al sumarle 2 sigue siendo par, luego i +2 mod 2=0 vale.

- $result + i = \sum_{j=0}^{i-1}$ (if j mod 2=0 then j else 0 fi)

  La sumatoria puede separarse en 3 términos:
  $\sum_{j=0}^{i-1}$ (if j mod 2=0 then j else 0 fi)+
  if i mod 2=0 then i else 0 fi
  if i+1 mod 2=0 then i+1 else 0 fi
  El primero de los 3 términos es igual a $result$ (lo sabemos por I).
  El segundo término es i (ya que por I sabemos que i es par).
  El tercer término es 0 (ya que por I sabemos que i es par, y por lo tanto $i+1$ es impar).
  Entonces sumando los 3 términos nos queda: $result + i + 0$, que es lo que esperábamos que valiera la sumatoria $\sum_{j=0}^{i+1}(...)$
  Como $(I \wedge B) \Rightarrow wp(ciclo, I)$,podemos afirmar que el cuerpo del ciclo preserva el invariante.

$\{(I \wedge B \wedge v_0 = f_v)\}$ **ciclo** $\{(f_v < v_0)\}$

Dado que queremos demostrar que vale una tripla de Hoare, comenzaremos calculando la precondición más débil $wp(ciclo, f_v < v_0)$.

$$wp(S1; S2, f_v < v_0) \overset{Ax3}{\equiv} wp(S1, wp(S2, n-i < v_0))$$
$$\overset{Ax1}{\equiv} wp(S1, true \wedge_L n-(i+2) < v_0)$$
$$\overset{Ax3}{\equiv} true \wedge_L (true \wedge_L n-(i+2) < v_0)$$
$$\equiv n-i-2 < v_0$$

Es decir, $wp(S1; S2, f_v < v_0) = n-i-2 < v_0$. Ahora debemos ver que $(I \wedge B \wedge v_0 = f_v)$ implican dicha WP. Parte de la hipótesis es que $v_0 = f_v$, es decir $v_0 = n-i$. Restando 2 a ambos lados, $n-i-2 = v_0 - 2 < v_0$.

$(I \wedge f_v \leq 0) \Rightarrow \neg B$

Debemos mostrar que vale $\neg B$, es decir $i \geq n$.
Sabemos que $f_v \leq 0$, es decir $n - i \leq 0$, luego $n \leq i$, como queriamos demostrar.

Ejercicio 3

proc exponenciacion(in $s : \mathbb{Z}$,in $n : \mathbb{Z}$, out $result : \mathbb{Z}$) {
Pre$\{n \geq 0 \wedge \neg(m = 0 \wedge n = 0)\}$
Post$\{result = m^n\}$
}

a)
```
i:=1;
result:=m;

while(i<n) do
  result=:result*m;
  i:=i+1
endwhile
```

Especificacion del ciclo:

- $P_c : result = m \wedge i = 1 \wedge n \geq 0 \wedge \neg(m = 0 \wedge n = 0)$
- $Q_c : result = m^n$
- $I \equiv 1 \leq i \leq n \wedge result = m^i$
- $f_v : n - i$

Solo demostraremos correción parcial de este ciclo.

$P_c \Rightarrow I$

Debemos demostrar que vale I sabiendo que vale P_c

- $1 \leq i \leq n$
  Por P_c sabemos que $i = 1$, entonces, la primera desigualdad es una tautologia, $1 \leq 1$, para la segunda vemos que $n \geq 0 \wedge \neg(m = 0 \wedge n = 0)$, luego como n, no puede ser 0 y es mayor o igual a cero tengo que $1 \leq n$.
- $result = m^i$
  Trivial, por P_c $result = m$, y como $i = 1$, entonces $m^i = m$, por lo que la igualdad vale.

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que vale $I \wedge \neg B$.

- cuando deja de valer la guarda, $i = n$ sabemos por el invariante que $result = m^i$, como $i = n$, entonces, vale que $result = m^n$, que es exactamente lo mismo que Q_c

$\{I \wedge B\}$ ciclo $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: $result := result * m;$
S2: $i := i + 1$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{3}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$\begin{aligned} wp(S2, I) &\overset{Ax1}{\equiv} def(i+1) \wedge_L I^i_{i+1} \\ &\equiv true \wedge_L (1 \leq i+1 \leq n \wedge_L result = m^{i+1}) \\ &\equiv (1 \leq i+1 \leq n \wedge_L result = m^{i+1}) \end{aligned}$$

Volviendo a (3), reemplazamos $wp(S2, I)$ y nos queda:

$$\begin{aligned} wp(S1; S2, I) &\stackrel{Ax3}{\equiv} wp(S1, wp(S2, I)) \equiv wp(S1, 1 \leq i+1 \leq n \wedge_L result = m^{i+1}) \\ &\stackrel{Ax1}{\equiv} def(result * m) \wedge_L 1 \leq i+1 \leq n \wedge_L result * m = m^{i+1} \\ &\equiv 1 \leq i+1 \leq n \wedge_L result * m = m^{i+1} \end{aligned}$$

Una vez calculada la precondición más débil, debemos ver si $(I \wedge B)$ implican dicha precondición. Probamos cada parte por separado:

- $0 \leq i+1 \leq n$

  Sabemos por I que $i > 0$, luego podemos afirmar que $1 \leq i+1$
  Sabemos por B que $i < n$, luego $i+1 \geq n$.
- $result * m = m^{i+1}$

  Divido en ambos miembros por m, y tengo lo mismo que en mi invariante.

  Como $(I \wedge B) \Rightarrow wp(ciclo, I)$, podemos afirmar que el cuerpo del ciclo preserva el invariante.

$\{(I \wedge B \wedge v_0 = f_v)\}$ ciclo $\{(f_v < v_0)\}$

Dado que queremos demostrar que vale una tripla de Hoare, comenzaremos calculando la precondición más débil $wp(ciclo, f_v < v_0)$.

$$\begin{aligned} wp(S1; S2, f_v < v_0) &\stackrel{Ax3}{\equiv} wp(S1, wp(S2, n - i < v_0)) \\ &\stackrel{Ax1}{\equiv} wp(S1, true \wedge_L n - (i+1) < v_0) \\ &\stackrel{Ax3}{\equiv} true \wedge_L (true \wedge_L n - (i+1) < v_0) \\ &\equiv n - i - 1 < v_0 \end{aligned}$$

Es decir, $wp(S1; S2, f_v < v_0) = n - i - 1 < v_0$. Ahora debemos ver que $(I \wedge B \wedge v_0 = f_v)$ implican dicha WP. Parte de la hipótesis es que $v_0 = f_v$, es decir $v_0 = n - i$. Restando 1 a ambos lados, $n - 1 - i = v_0 - 1 < v_0$.

$(I \wedge f_v \leq 0) \Rightarrow \neg B$

Debemos mostrar que vale $\neg B$, es decir $i \geq n$.
Sabemos que $f_v \leq 0$, es decir $n - i \leq 0$, luego $n \leq i$, como queriamos demostrar.

b) falla el primero, el segundo el tercero el cuarto y el quinto, falla todo para ser más preciso.

c) creo que no falla, empieza con el caso $m^0 = 1$, y sigue como en mi demostración en a), no lo voy a probar de vuelta.

d) habria que agregar en la pre que $n \geq 2$.

Ejercicio 4

proc sumaDivisores(in $n : \mathbb{Z}$, out $result : \mathbb{Z}$) {
  Pre$\{n \geq 1\}$
  Post$\{result = \sum_{j=1}^{n}(\text{if } n \; mod \; j = 0 \text{ then } j \text{ else } 0 \text{ fi})\}$
}

a)
```
i:=0;
result:=0;

while(i<n) do
  i:=i+1
  if (n mod i = 0)
     result=:result +i
  else
     skip
  endif;
endwhile
```

Especificacion del ciclo:

b)
- $P_c : result = 0 \wedge i = 0 \wedge n \geq 1$
- $Q_c : result = \sum_{j=1}^{n}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi})$
- $I \equiv 0 \leq i \leq n \wedge result = \sum_{j=1}^{i}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi})$
- $f_v : n - i$

Cambio el invariante para que empieze en $0 \leq i$
Solo demostraremos correción parcial de este ciclo.

$P_c \Rightarrow I$

Debemos demostrar que vale I sabiendo que vale P_c

- $1 \leq i \leq n$
  Por P_c sabemos que $i = 1$, entonces, la primera desigualdad es una tautologia, $1 \leq 1$, para la segunda vemos que $n \geq 1$, luego como n, no puede ser 0 y es mayor o igual a 1 tengo que $1 \leq n$.
- $result = \sum_{j=1}^{i}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi})$
  Trivial, por P_c $result = 0$, y como $i = 0$, entonces $\sum_{j=1}^{0}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi}) = result = 0$.

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que vale $I \wedge \neg B$.

- cuando deja de valer la guarda, $i = n$ sabemos por el invariante que $result = \sum_{j=1}^{i}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi})$, como $i = n$, entonces, vale que $result = \sum_{j=1}^{n}(\text{if } n \text{ } mod \text{ } j = 0 \text{ then } j \text{ else } 0 \text{ fi})$, que es exactamente lo mismo que Q_c

$\{I \wedge B\}$ **ciclo** $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: $i := i + 1$
S2: **if** $(n \text{ } mod \text{ } i = 0)$ **then** $result := result + i$ **else** $skip$ **endif**;
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \stackrel{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{4}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$\begin{aligned}
wp(S2, I) &\stackrel{Ax4}{\equiv} def(n \text{ } mod \text{ } i = 0) \wedge_L \left(\Big((n \text{ } mod \text{ } i = 0) \wedge (wp(result := result + i, I)) \Big) \vee \Big(\neg(n \text{ } mod \text{ } i = 0) \wedge (wp(skip, I)) \Big) \right) \\
&\equiv True \wedge_L \left(\Big((n \text{ } mod \text{ } i = 0) \wedge (def(result + i) \wedge_L I_{result+i}^{result}) \Big) \vee \Big(\neg(n \text{ } mod \text{ } i = 0) \wedge (I) \Big) \right) \\
&\equiv \left(\Big((n \text{ } mod \text{ } i = 0) \wedge (I_{result+i}^{result}) \Big) \vee \Big(\neg(n \text{ } mod \text{ } i = 0) \wedge (I) \Big) \right)
\end{aligned}$$

Volviendo a (4), reemplazamos $wp(S2, I)$ y nos queda:

$$
\begin{aligned}
wp(S1; S2, I) &\overset{Ax3}{\equiv} wp(S1, wp(S2, I))\\
&\equiv wp(S1, \Big((n\ mod\ i = 0) \land (1 \leq i \leq n \land result + 1 = \sum_{j=1}^{i}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big)\\
&\lor \Big(\neg(n\ mod\ i = 0) \land (1 \leq i \leq n \land result = \sum_{j=1}^{i}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big))\\
&\overset{Ax1}{\equiv} def(i+1) \land_L \Big((n\ mod\ i+1 = 0) \land (1 \leq i+1 \leq n \land result + 1 = \sum_{j=1}^{i+1}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big)\\
&\lor \Big(\neg(n\ mod\ i+1 = 0) \land (1 \leq i+1 \leq n \land result = \sum_{j=1}^{i+1}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big))\\
&\equiv \Big((n\ mod\ i+1 = 0) \land (1 \leq i+1 \leq n \land result + 1 = \sum_{j=1}^{i+1}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big)\\
&\lor \Big(\neg(n\ mod\ i+1 = 0) \land (1 \leq i+1 \leq n \land result = \sum_{j=1}^{i+1}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big))
\end{aligned}
$$

Una vez calculada la precondición más débil, debemos ver si $(I \land B)$ implican dicha precondición. Probamos cada parte por separado:

- $\Big((n\ mod\ i+1 = 0) \land (1 \leq i+1 \leq n \land result + 1 = \sum_{j=1}^{i+1}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big)$
  Ya lo hice en la guia 4
- $\Big(\neg(n\ mod\ i+1 = 0) \land (1 \leq i+1 \leq n \land result = \sum_{j=1}^{i+1}(\text{if } n\ mod\ j = 0 \text{ then } j \text{ else } 0 \text{ fi}))\Big))$
  Ya lo hice en la guia 4
  Como $(I \land B) \Rightarrow wp(ciclo, I)$, podemos afirmar que el cuerpo del ciclo preserva el invariante.

Ejercicio 5

proc sumaPosicionesImpares(in $s : seq\langle\mathbb{Z}\rangle$, out $result : \mathbb{Z}$) {
  Pre$\{True\}$
  Post$\{result = \sum_{i=0}^{|s|-1}(\text{if } i\ mod\ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})\}$
}

a)
```
j:=0;
result:=0;

while(j<s.size()) do
  if (j mod 2 = 1)
     result:=result +s[j]
  else
     skip
  endif;
   j:=j+1
endwhile
```

Especificacion del ciclo:

b)
- $P_c : result = 0 \land j = 0$
- $Q_c : result = \sum_{i=0}^{|s|-1}(\text{if } i\ mod\ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi}))$

- $I \equiv 0 \leq j \leq |s| \wedge result = \sum_{i=0}^{j-1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})$
- $f_v : n - i$

Solo demostraremos correción parcial de este ciclo.

$P_c \Rightarrow I$

Debemos demostrar que vale I sabiendo que vale P_c

- $0 \leq j \leq |s|$

  Por P_c sabemos que $j = 0$, entonces, la primera desigualdad es una tautologia, $0 \leq 0$, para la segunda vemos que $|s| \geq 0$, luego como $|s|$, no puede ser menor que cero es valido.
- $result = \sum_{i=0}^{j-1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})$
  Trivial, por P_c $result = 0$, y como $j = 0$, entonces $\sum_{i=0}^{-1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi}) = result = 0$.

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que vale $I \wedge \neg B$.

- cuando deja de valer la guarda, $j = |s|$ sabemos por el invariante que $result = \sum_{i=0}^{j-1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})$, como $j = |s|$, entonces, vale que $result = \sum_{j=1}^{|s|-1}(\text{if } n \ mod \ j = 0 \text{ then } j \text{ else } 0 \text{ fi})$, que es exactamente lo mismo que Q_c

$\{I \wedge B\}$ **ciclo** $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: **if** $(j \ mod \ 2 = 1$ **then** $result := result + s[j]$ **else** $skip$ **endif**;
S2: $j := j + 1$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \stackrel{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{5}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$\begin{aligned} wp(S2, I) &\stackrel{Ax1}{\equiv} def(j+1) \wedge_L I_{j+1}^{j} \\ &\equiv true \wedge_L 0 \leq j+1 \leq |s| \wedge result = \sum_{i=0}^{j-1+1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi}) \\ &\equiv 0 \leq j+1 \leq |s| \wedge result = \sum_{i=0}^{j-1+1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})) \end{aligned}$$

Volviendo a (5), reemplazamos $wp(S2, I)$ y nos queda:

$$
\begin{aligned}
wp(S1; S2, I) &\overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \\
&\equiv wp(S1, 0 \le j+1 \le |s| \land result = \sum_{i=0}^{j-1+1} (\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})) \\
&\equiv wp(S1, I^j_{j+1}) \\
&\overset{Ax4}{\equiv} def(j \ mod \ 2 = 1) \land_L \Bigg(\Big((j \ mod \ 2 = 1) \land (wp(result := result + s[j], I^j_{j+1}))\Big) \\
&\lor \Big(\neg(j \ mod \ 2 = 1) \land (wp(skip, I^j_{j+1}))\Big)\Bigg) \\
&\overset{Ax1}{\equiv} True \land_L \Bigg(\Big((j \ mod \ 2 = 1) \land (wp(result := result + s[j], I^j_{j+1}))\Big) \lor \Big(\neg(j \ mod \ 2 = 1) \land I^j_{j+1}\Big)\Bigg) \\
&\equiv \Bigg(\Big((j \ mod \ 2 = 1) \land (wp(result := result + s[j], I^j_{j+1}))\Big) \lor \Big(\neg(j \ mod \ 2 = 1) \land I^j_{j+1}\Big)\Bigg) \\
&\equiv \Bigg(\Big((j \ mod \ 2 = 1) \land (0 \le j+1 \le |s| \land result + s[j] = \sum_{i=0}^{j-1+1} (\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})\Big) \\
&\lor \Big(\neg(j \ mod \ 2 = 1) \land 0 \le j+1 \le |s| \land result = \sum_{i=0}^{j-1+1} (\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})\Big)\Bigg)
\end{aligned}
$$

Una vez calculada la precondición más débil, debemos ver si $(I \land B)$ implican dicha precondición. Probamos cada parte por separado:

- $\Big((j \ mod \ 2 = 1) \land (0 \le j+1 \le |s| \land result + s[j] = \sum_{i=0}^{j-1+1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})\Big)$
  Supongo que vale la izquierda y entonces vale la derecha.
- $\Big(\neg(j \ mod \ 2 = 1) \land 0 \le j+1 \le |s| \land result = \sum_{i=0}^{j-1+1}(\text{if } i \ mod \ 2 = 1 \text{ then } s[i] \text{ else } 0 \text{ fi})\Big)$
  Supongo que vale la izquierda y entonces vale la derecha.
  Como $(I \land B) \Rightarrow wp(ciclo, I)$, podemos afirmar que el cuerpo del ciclo preserva el invariante.

Ejercicio 6

proc maximo(in $s : seq\langle\mathbb{Z}\rangle$, out $i : \mathbb{Z}$) {
  Pre$\{|s| \ge 1\}$
  Post$\{0 \le i < |s| \land_L (\forall j : \mathbb{Z})(0 \le j < |s| \rightarrow_L s[j] \le s[i])\}$
}

a)
```
i:=0;
j:=1;

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

Especificacion del ciclo:

b)
- $P_c : i = 0 \land j = 1 \land |s| \geq 1$
- $Q_c : 0 \leq i < |s| \land_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \leq s[i])$
- $I \equiv (0 \leq i < |s| \land 1 \leq j \leq |s|) \land_L (\forall k : \mathbb{Z})(0 \leq k < j \rightarrow_L s[k] \leq s[i])$
- $f_v : |s| + 1 - j$

Solo demostraremos correción parcial de este ciclo.

$P_c \Rightarrow I$

Debemos demostrar que vale I sabiendo que vale P_c

- $(0 \leq i < |s| \land 1 \leq j \leq |s|)$

  Por P_c sabemos que $j = 1, i = 0$, entonces, la primera desigualdad de ambos terminos es una tautologia, $0 \leq 0 \land 1 \leq 1$, para la segunda vemos que por P_c $|s| \geq 1$, luego como i, j son o menores o iguales a 1 vale.
- $(\forall k : \mathbb{Z})(0 \leq k < j \rightarrow_L s[k] \leq s[i])$

  Parto desde que $i = 0, j = 1$ El unico .k que cumple que $k < 1$ es 0, por lo que es cierto que $s[0] \leq s[0]$.

$(I \land \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que vale $I \land \neg B$.

- cuando deja de valer la guarda, $j = |s|$ reemplazamos j por $|s|$ en el invariante y tenemos Q_c

$\{I \land B\}$ **ciclo** $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \land B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: **if** $(s[j] > s[i])$ **then** $i := j$ **else** $skip$ **endif**;
S2: $j := j + 1$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \stackrel{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{6}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$\begin{aligned} wp(S2, I) &\stackrel{Ax1}{\equiv} def(j+1) \land_L I^j_{j+1} \\ &\equiv true \land_L (0 \leq i < |s| \land 1 \leq j+1 \leq |s|) \land_L (\forall k : \mathbb{Z})(0 \leq k < j+1 \rightarrow_L s[k] \leq s[i]) \\ &\equiv (0 \leq i < |s| \land 1 \leq j+1 \leq |s|) \land_L (\forall k : \mathbb{Z})(0 \leq k < j+1 \rightarrow_L s[k] \leq s[i]) \end{aligned}$$

Volviendo a (6), reemplazamos $wp(S2, I)$ y nos queda:

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I))$$
$$\equiv wp(S1, I^j_{j+1})$$
$$\overset{Ax4}{\equiv} def(s[j] > s[i]) \wedge_L \left(\left((s[j] > s[i]) \wedge (wp(i := j, I^j_{j+1})) \right) \right.$$
$$\left. \vee \left(\neg(s[j] > s[i]) \wedge (wp(skip, I^j_{j+1})) \right) \right)$$
$$\overset{Ax1}{\equiv} (0 \leq i < |s| \wedge 0 \leq j < |s|) \wedge_L \left(\left((s[j] > s[i]) \wedge (wp(i := j, I^j_{j+1})) \right) \right.$$
$$\left. \vee \left(\neg(s[j] > s[i]) \wedge (I^j_{j+1}) \right) \right)$$
$$\equiv (0 \leq i < |s| \wedge 0 \leq j < |s|) \wedge_L$$
$$\left(\left((s[j] > s[i]) \wedge ((0 \leq j < |s| \wedge 1 \leq j < |s|) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j+1 \rightarrow_L s[k] \leq s[j]))) \right) \right.$$
$$\left. \vee \left(\neg(s[j] > s[i]) \wedge ((0 \leq i < |s| \wedge 1 \leq j+1 \leq |s|) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j+1 \rightarrow_L s[k] \leq s[i])) \right) \right)$$

Una vez calculada la precondición más débil, debemos ver si $(I \wedge B)$ implican dicha precondición. Probamos cada parte por separado suponiendo que vale o no la guarda:

- $(0 \leq i < |s| \wedge 0 \leq j < |s|)$
  la primera esta en el invariante, para la segunda por la B se que $j < |s|$, entonces puedo afirmar que $j < |s||$.
- $\left(\left((s[j] > s[i]) \wedge ((0 \leq j < |s| \wedge 1 \leq j+1 \leq |s|) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j+1 \rightarrow_L s[k] \leq s[j]))) \right) \right)$
  Supongo que vale $(s[j] > s[i])$ y entonces el rango de j lo implica I. En el para todo, se que hasta j es implicable por el I, para el termino $j+1$, $k = j$ entonces $s[j] \leq s[j]$.
- $\left(\left(\neg(s[j] > s[i]) \wedge ((0 \leq i < |s| \wedge 1 \leq j+1 \leq |s|) \wedge_L (\forall k : \mathbb{Z})(0 \leq k < j+1 \rightarrow_L s[k] \leq s[i])) \right) \right)$
  Supongo que vale $\neg(s[j] > s[i])$ y entonces el rango de j lo implica I. En el para todo, se que hasta j es implicable por el I, para el termino $j+1$, $k = j$ entonces quiero que $s[j] \leq s[i]$, pero esto es $\neg(s[j] > s[i])$ entonces vale .

  Como $(I \wedge B) \Rightarrow wp(ciclo, I)$, podemos afirmar que el cuerpo del ciclo preserva el invariante.

Ejercicio 7

proc copiarSecuencia(in $n : seq\langle\mathbb{Z}\rangle$, inout $r : seq\langle\mathbb{Z}\rangle$) {
Pre$\{|s| = |r| \wedge r = r_0\}$
Post$\{|s| = |r| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = r[j])\}$
}

a)
```
j:=0;
while(i<s.size()) do
  r[i]:=s[i];
   i:=i+1
endwhile
```

Especificacion del ciclo:

b)
- $P_c : i = 0 \wedge |s| = |r| \wedge r = r_0$

- $Q_c : i = |s| \wedge |s| = |r| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = r[j])$
- $I \equiv 0 \leq i \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L r[j] = s[j]) \wedge (\forall j : \mathbb{Z})(i \leq j < |r| \rightarrow_L r[j] = r_0[j])$
- $f_v : n - i$

Solo demostraremos correción parcial de este ciclo.

$P_c \Rightarrow I$

Debemos demostrar que vale I sabiendo que vale P_c

- $0 \leq i \leq |s|$

  Por P_c sabemos que $i = 0$, entonces $0 \leq i$ vale. Además, $|s| \geq 0$ (porque las listas no pueden tener una cantidad negativa de elementos), luegos $|s| \geq i$
- $|r_0| = |r|$

  P_c indica que $r = r_0$. Si dos secuencias son iguales, entonces tienen la misma longitud.
- $|s| = |r|$

  P_c afirma eso.
- $(\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L r[j] = s[j])$
  Dado que $i = 0$, al evaluar cualquier valor de j en $0 \leq j < i$ obtenemos $False$. Luego la implicación es verdadera para cualquier valor de j (es decir, para todos los valores).
- $(\forall j : \mathbb{Z})(i \leq j < |r| \rightarrow_L r[j] = r_0[j])$
  Dado que $i = 0$, esa expresión afirma que todas las posiciones de r son iguales a las de $r : 0$. Esto es cierto pues por P_c sabemos que $r = r_0$

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que vale $I \wedge \neg B$.

- Por I savemos que $i \leq |s|$, y por $\neg B$ savemos que $i \geq |s|$. Entonces i debe ser igual a $|s|$
- Es trivial ver que vale $|s| = |r|$ ya que el invariante expresa exactamente eso.
- Sabemos por I que $(\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L r[j] = s[j])$, y además sabemos que $i = |s|$. Reemplazando, el valor de i nos queda $(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L r[j] = s[j])$, que es lo que queriamos ver.

$\{I \wedge B\}$ **ciclo** $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: $r[i] := s[i]$;
S2: $i := i + 1$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \overset{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{7}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$\begin{aligned}
wp(S2, I) &\overset{Ax1}{\equiv} def(i+1) \wedge_L I^j_{i+1} \\
&\equiv true \wedge_L 0 \leq i + 1 \leq |s| \wedge |r_0| = |r| \wedge |s| = r| \wedge_L \\
&(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L r[j] = s[j]) \wedge (\forall j : \mathbb{Z})(i + 1 \leq j < |r| \rightarrow_L r[j] = r_0[j]) \\
&\equiv 0 \leq i + 1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\
&(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L r[j] = s[j]) \wedge (\forall j : \mathbb{Z})(i + 1 \leq j < |r| \rightarrow_L r[j] = r_0[j])
\end{aligned}$$

Volviendo a (7). Qué axioma usar? No hay ningún axioma para una instrucción con la forma de S1. Pero en realidad lo que estamos haciendo es una asignación. Podemos reescribirla como **r:= setAt(r, i, s[i])** (también podemos escribirlo como **r:= setAt(r, i, s[i])**)). Ahora sí tenemos una instrucción que permite utilizar el axioma 1.

$$\begin{aligned}
wp(S1;S2,I) &\overset{Ax3}{\equiv} wp(r := setAt(r,i,s[i]), wp(S2,I)) \\
&\overset{Ax1}{\equiv} def(setAt(r,i,s[i])) \wedge_L wp(S2,I)^{r}_{setAt(r,i,s[i])} \\
&\equiv 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |setAt(r,i,s[i])| \wedge |s| = |setAt(r,i,s[i])| \wedge_L \\
&(\forall j:\mathbb{Z})(0 \leq j < i+1 \rightarrow_L setAt(r,i,s[i])[j] = s[j]) \wedge \\
&(\forall j:\mathbb{Z})(i+1 \leq j < |r| \rightarrow_L setAt(r,i,s[i])[j] = r_0[j])
\end{aligned}$$

Revisemos las apariciones de setAt para ver si pueden ser errmplazadas/simplificadas:

- $|setAt(r,i,s[i])|$. Esta expresión es equivalente a $|r|$
- $setAt(r,i,s[i])[j]$. El valor de esta expresion depende del valor de i y el valor de j. Si $i=j$, la expresión evalúa a $s[i]$, en caso contrario evalúa a $r[j]$. En la primer aparición de dicha expresión, no podemos saber si $i=j$, ya que sabemos que $j < i+1$, asi que tendremos que trabajar un poco más en esa parte de la expresión. La segunda aparición de setAt se da cuando $i+1 \leq j$, luego podemos afirmar que $i \neq j$, es decir que en ese caso: $setAt(r,i,s[i])[j] = r[j]$

Entonces,

$$\begin{aligned}
wp(S1;S2,I) \equiv\ & 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\
&(\forall j:\mathbb{Z})(0 \leq j < i+1 \rightarrow_L setAt(r,i,s[i])[j] = s[j]) \wedge \\
&(\forall j:\mathbb{Z})(i+1 \leq j < |r| \rightarrow_L r[j] = r_0[j])
\end{aligned}$$

El cuantificador $(\forall j:\mathbb{Z})(0 \leq j < i+1...$ puede reescribirse, teniendo en cuenta el rango hasta $< i$, y la expresión con $j=i$ aparte:

$$\begin{aligned}
wp(S1;S2,I) \equiv\ & 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\
&(\forall j:\mathbb{Z})(0 \leq j < i \rightarrow_L setAt(r,i,s[i])[j] = s[j]) \wedge setAt(r,i,s[i])[i] = s[i]) \wedge \\
&(\forall j:\mathbb{Z})(i+1 \leq j < |r| \rightarrow_L r[j] = r_0[j])
\end{aligned}$$

Ahora sí, en cada aparición de setAt, sabemos con certeza si $i=j$ o no:

$$\begin{aligned}
wp(S1;S2,I) \equiv\ & 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\
&(\forall j:\mathbb{Z})(0 \leq j < i \rightarrow_L setAt(r,i,s[i])[j] = s[j]) \wedge s[i] = s[i]) \wedge \\
&(\forall j:\mathbb{Z})(i+1 \leq j < |r| \rightarrow_L r[j] = r_0[j])
\end{aligned}$$

Para tener en cuenta:

- $s[i] = s[i] \equiv True$
- Como $|s| = |r|$, entonces $\equiv 0 \leq i < |r| \equiv 0 \leq i < |s|$, y podemos dejar solo una de las expresiones.

$$\begin{aligned}
wp(S1;S2,I) \equiv\ & 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\
&(\forall j:\mathbb{Z})(0 \leq j < i \rightarrow_L setAt(r,i,s[i])[j] = s[j])) \wedge \\
&(\forall j:\mathbb{Z})(i+1 \leq j < |r| \rightarrow_L r[j] = r_0[j])
\end{aligned}$$

Veamos ahora que $(I \wedge B) \Rightarrow wp(ciclo, I)$. Usando como hipótesis $I \wedge B$, veamos que se cumplen todas las partes de $wp(S1, S2, I)$:

- $0 \leq i < |s|$
  Vale pues por I sabemos que $i \geq 0$, y por B sabemos que $i < |s|$

- $0 \leq i+1 \leq |s|$ Vale pues por I sabemos que $i \geq 0$,entonces $i+1 \geq i \geq 0$ y por transitividad $i+1 \geq 0$
  Además, por B sabemos que $i < |s|$, lo cual (dado que tratamos números enteros) es equivalente a $i+1 \leq |s|$
- $|r_0| = |r|$
  Vale porque el invariante incluye esta misma expresión
- $|s| = |r|$
  Vale porque el invariante incluye esta misma expresión
- $(\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L setAt(r, i, s[i])[j] = s[j]))$
  Vale porque el invariante incluye esta misma expresión
- $(\forall j : \mathbb{Z})(i+1 \leq j < |r| \rightarrow_L r[j] = r_0[j])$
  El invariante afirma que $r[j] = r_0[j]$ para los j que estén en el rango $rango_1 = [i..|r|)$. El rango para el cual queremos ver si vale es $[i+1..|r|)$, el cual está incluído en $rango_1$ (nuestro rango es más chico que $rango_1$), con lo cual podemos afirmar que $r[j] = r_0[j]$ para los j que nos interesan.

Ejercicio 8

proc copiarSecuencia(in $n : seq\langle\mathbb{Z}\rangle$, inout $r : seq\langle\mathbb{Z}\rangle$) {

Pre$\{|s| = |r| \land r = r_0\}$

Post$\{|s| = |r| \land_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = r[j])\}$

}

a)
```
j:=0;
while(i<s.size()) do
  r[i]:=s[i];
   i:=i+1
endwhile
```

Especificacion del ciclo:

b)
- $P_c : i = 0 \land |s| = |r| \land r = r_0$
- $Q_c : i = |s| \land |s| = |r| \land_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = r[j])$
- $I \equiv 0 \leq i \leq |s| \land |r_0| = |r| \land |s| = |r| \land_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L r[j] = s[j]) \land (\forall j : \mathbb{Z})(i \leq j < |r| \rightarrow_L r[j] = r_0[j])$
- $f_v : n - i$

Solo demostraremos correción parcial de este ciclo.

$P_c \Rightarrow I$

Debemos demostrar que vale I sabiendo que vale P_c

- $0 \leq i \leq |s|$

  Por P_c sabemos que $i = 0$, entonces $0 \leq i$ vale. Además, $|s| \geq 0$ (porque las listas no pueden tener una cantidad negativa de elementos), luegos $|s| \geq i$
- $|r_0| = |r|$

  P_c indica que $r = r_0$. Si dos secuencias son iguales, entonces tienen la misma longitud.
- $|s| = |r|$

  P_c afirma eso.
- $(\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L r[j] = s[j])$
  Dado que $i = 0$, al evaluar cualquier valor de j en $0 \leq j < i$ obtenemos $False$. Luego la implicación es verdadera para cualquier valor de j (es decir, para todos los valores).
- $(\forall j : \mathbb{Z})(i \leq j < |r| \rightarrow_L r[j] = r_0[j])$
  Dado que $i = 0$, esa expresión afirma que todas las posiciones de r son iguales a las de $r : 0$. Esto es cierto pues por P_c sabemos que $r = r_0$

$(I \wedge \neg B) \Rightarrow Q_c$

Queremos demostrar que vale Q_c, asumiendo que vale $I \wedge \neg B$.

- Por I savemos que $i \leq |s|$, y por $\neg B$ savemos que $i \geq |s|$. Entonces i debe ser igual a $|s|$
- Es trivial ver que vale $|s| = |r|$ ya que el invariante expresa exactamente eso.
- Sabemos por I que $(\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L r[j] = s[j])$, y además sabemos que $i = |s|$. Reemplazando, el valor de i nos queda $(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L r[j] = s[j])$, que es lo que queriamos ver.

$\{I \wedge B\}$ **ciclo** $\{I\}$

Queremos ver que vale la siguiente tripla de Hoare $\{I \wedge B\}$ **ciclo** $\{I\}$.
Llamemos S1 a la primer instrucción del cuerpo del ciclo, S2 a la segunda:
S1: $r[i] := s[i];$
S2: $i := i + 1$
Lo primero que haremos es calcular $wp(ciclo, I)$.

$$wp(S1; S2, I) \stackrel{Ax3}{\equiv} wp(S1, wp(S2, I)) \tag{8}$$

Antes de seguir, debemos calcular $wp(S2, I)$. Para eso usaremos el axioma 1:

$$\begin{aligned}
wp(S2, I) &\stackrel{Ax1}{\equiv} def(i + 1) \wedge_L I^j_{i+1} \\
&\equiv true \wedge_L 0 \leq i + 1 \leq |s| \wedge |r_0| = |r| \wedge |s| = r| \wedge_L \\
&(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L r[j] = s[j]) \wedge (\forall j : \mathbb{Z})(i + 1 \leq j < |r| \rightarrow_L r[j] = r_0[j]) \\
&\equiv 0 \leq i + 1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\
&(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L r[j] = s[j]) \wedge (\forall j : \mathbb{Z})(i + 1 \leq j < |r| \rightarrow_L r[j] = r_0[j])
\end{aligned}$$

Volviendo a (7). Qué axioma usar? No hay ningún axioma para una instrucción con la forma de S1. Pero en realidad lo que estamos haciendo es una asignación. Podemos reescribirla como **r:= setAt(r, i, s[i])** (también podemos escribirlo como **r:= setAt(r, i, s[i])**)). Ahora sí tenemos una instrucción que permite utilizar el axioma 1.

$$\begin{aligned}
wp(S1; S2, I) &\stackrel{Ax3}{\equiv} wp(r := setAt(r, i, s[i]), wp(S2, I)) \\
&\stackrel{Ax1}{\equiv} def(setAt(r, i, s[i])) \wedge_L wp(S2, I)^r_{setAt(r,i,s[i])} \\
&\equiv 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i + 1 \leq |s| \wedge |r_0| = |setAt(r, i, s[i])| \wedge |s| = |setAt(r, i, s[i])| \wedge_L \\
&(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L setAt(r, i, s[i])[j] = s[j]) \wedge \\
&(\forall j : \mathbb{Z})(i + 1 \leq j < |r| \rightarrow_L setAt(r, i, s[i])[j] = r_0[j])
\end{aligned}$$

Revisemos las apariciones de setAt para ver si pueden ser errmplazadas/simplificadas:

- $|setAt(r, i, s[i])|$. Esta expresión es equivalente a $|r|$
- $setAt(r, i, s[i])[j]$. El valor de esta expresion depende del valor de i y el valor de j. Si $i = j$, la expresión evalúa a $s[i]$, en caso contrario evalúa a $r[j]$. En la primer aparición de dicha expresión, no podemos saber si $i = j$, ya que sabemos que $j < i + 1$, asi que tendremos que trabajar un poco más en esa parte de la expresión. La segunda aparición de setAt se da cuando $i + 1 \leq j$, luego podemos afirmar que $i \neq j$, es decir que en ese caso: $setAt(r, i, s[i])[j] = r[j]$

Entonces,

$$\begin{aligned}
wp(S1; S2, I) \equiv\ & 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i + 1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\
&(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L setAt(r, i, s[i])[j] = s[j]) \wedge \\
&(\forall j : \mathbb{Z})(i + 1 \leq j < |r| \rightarrow_L r[j] = r_0[j])
\end{aligned}$$

El cuantificador $(\forall j:\mathbb{Z})(0 \leq j < i+1...$ puede reescribirse, teniendo en cuenta el rango hasta $< i$, y la expresión con $j = i$ aparte:

$$\begin{aligned} wp(S1;S2,I) \equiv\ & 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\ & (\forall j:\mathbb{Z})(0 \leq j < i \to_L setAt(r,i,s[i])[j] = s[j]) \wedge setAt(r,i,s[i])[i] = s[i]) \wedge \\ & (\forall j:\mathbb{Z})(i+1 \leq j < |r| \to_L r[j] = r_0[j]) \end{aligned}$$

Ahora sí, en cada aparición de setAt, sabemos con certeza si $i = j$ o no:

$$\begin{aligned} wp(S1;S2,I) \equiv\ & 0 \leq i < |r| \wedge 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\ & (\forall j:\mathbb{Z})(0 \leq j < i \to_L setAt(r,i,s[i])[j] = s[j]) \wedge s[i] = s[i]) \wedge \\ & (\forall j:\mathbb{Z})(i+1 \leq j < |r| \to_L r[j] = r_0[j]) \end{aligned}$$

Para tener en cuenta:

- $s[i] = s[i] \equiv True$
- Como $|s| = |r|$, entonces $\equiv 0 \leq i < |r| \equiv 0 \leq i < |s|$, y podemos dejar solo una de las expresiones.

$$\begin{aligned} wp(S1;S2,I) \equiv\ & 0 \leq i < |s| \wedge_L 0 \leq i+1 \leq |s| \wedge |r_0| = |r| \wedge |s| = |r| \wedge_L \\ & (\forall j:\mathbb{Z})(0 \leq j < i \to_L setAt(r,i,s[i])[j] = s[j])) \wedge \\ & (\forall j:\mathbb{Z})(i+1 \leq j < |r| \to_L r[j] = r_0[j]) \end{aligned}$$

Veamos ahora que $(I \wedge B) \Rightarrow wp(ciclo, I)$. Usando como hipótesis $I \wedge B$, veamos que se cumplen todas las partes de $wp(S1, S2, I)$:

- $0 \leq i < |s|$
  Vale pues por I sabemos que $i \geq 0$, y por B sabemos que $i < |s|$
- $0 \leq i+1 \leq |s|$ Vale pues por I sabemos que $i \geq 0$,entonces $i+1 \geq i \geq 0$ y por transitividad $i+1 \geq 0$
  Además, por B sabemos que $i < |s|$, lo cual (dado que tratamos números enteros) es equivalente a $i+1 \leq |s|$
- $|r_0| = |r|$
  Vale porque el invariante incluye esta misma expresión
- $|s| = |r|$
  Vale porque el invariante incluye esta misma expresión
- $(\forall j:\mathbb{Z})(0 \leq j < i \to_L setAt(r,i,s[i])[j] = s[j]))$
  Vale porque el invariante incluye esta misma expresión
- $(\forall j:\mathbb{Z})(i+1 \leq j < |r| \to_L r[j] = r_0[j])$
  El invariante afirma que $r[j] = r_0[j]$ para los j que estén en el rango $rango_1 = [i..|r|)$. El rango para el cual queremos ver si vale es $[i+1..|r|)$, el cual está incluído en $rango_1$ (nuestro rango es más chico que $rango_1$), con lo cual podemos afirmar que $r[j] = r_0[j]$ para los j que nos interesan.

c) Demostremos que el ciclo termina.
$\{(I \wedge B \wedge v_0 = f_v)\}$ **ciclo** $\{(f_v < v_0)\}$

Dado que queremos demostrar que vale una tripla de Hoare, comenzaremos calculando la precondición más débil $wp(ciclo, f_v < v_0)$.

$$\begin{aligned} wp(S1;S2, f_v < v_0) & \overset{Ax3}{\equiv} wp(S1, wp(S2, n-i < v_0)) \\ & \overset{Ax1}{\equiv} wp(S1, true \wedge_L n-(i+1) < v_0) \\ & \overset{Ax3}{\equiv} true \wedge_L (true \wedge_L n-(i+1) < v_0) \\ & \equiv n-i-1 < v_0 \end{aligned}$$

Es decir, $wp(S1; S2, f_v < v_0) = n - i - 1 < v_0$. Ahora debemos ver que $(I \wedge B \wedge v_0 = f_v)$ implican dicha WP. Parte de la hipótesis es que $v_0 = f_v$, es decir $v_0 = n - i$. Restando 1 a ambos lados, $n - 1 - i = v_0 - 1 < v_0$.

$(I \wedge f_v \leq 0) \Rightarrow \neg B$

Debemos mostrar que vale $\neg B$, es decir $i \geq n$.
Sabemos que $f_v \leq 0$, es decir $n - i \leq 0$, luego $n \leq i$, como queriamos demostrar.