

# Verificación de programas II: Teorema del Invariante

Diego Bendersky (TM) y Fermín Schlottmann (TN)

Algoritmos y Estructuras de Datos

24 de Septiembre de 2025

# Plan del día

## Plan del día

- Cierre wp
- Precondición más débil de ciclos
- Teorema del Invariante
- Teorema de Terminación
- Ejercicios de la práctica y de parcial

# Precondición más débil

## Precondición más débil – Idea informal

Es la  $P$  que permite que el programa **S** funcione correctamente, pero restringiendo lo menos posible.

## Principio de diseño

Ser cuidadoso con los resultados que se emiten y generoso con los parámetros que se reciben.

# Axiomas

## Definiciones (copiadas de la teórica)

- **Axioma 1:**  $wp(x := E, Q) \equiv \text{def}(E) \wedge_L Q_E^x$
- **Axioma 2:**  $wp(\text{skip}, Q) \equiv Q$
- **Axioma 3:**  $wp(\mathbf{S1}; \mathbf{S2}, Q) \equiv wp(\mathbf{S1}, wp(\mathbf{S2}, Q))$
- **Axioma 4:** Si  $\mathbf{S} = \text{if } B \text{ then } \mathbf{S1} \text{ else } \mathbf{S2} \text{ endif}$ , entonces

$$wp(\mathbf{S}, Q) \equiv \text{def}(B) \wedge_L \left( (B \wedge wp(\mathbf{S1}, Q)) \vee (\neg B \wedge wp(\mathbf{S2}, Q)) \right)$$

# Ejercicio

**Ejercicio 7.** Dado el siguiente condicional determinar la precondición más débil que permite hacer valer la poscondición (Q) propuesta. Se pide:

- Describir en palabras la WP esperada
- Derivarla formalmente a partir de los axiomas de precondición más débil. Para obtener el puntaje máximo deberá simplificarla lo más posible.

a)  $Q \equiv \{(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] > 0)\}$

```
if s[i] < 0 then
  | s[i] := -s[i]
else
  | s[i] := 0;
end
```

# Solución

La precondition más débil debería requerir que  $i$  esté en rango para que no se indefina el código, que el valor en la posición  $i$ -ésima sea negativo y que las demás posiciones del arreglo tengan valores positivo.

Se puede ver que si el valor de la posición  $i$ -ésima no fuera negativo, entonces el código  $S$  pondría un cero en esa posición y no se cumpliría la poscondición  $Q$ .

Ahora vamos a calcularla **formalmente**.

# Solución

Sean  $\mathbf{B} \equiv s[i] < 0$ ,  $\mathbf{S1} \equiv s[i] := -s[i]$ ,  $\mathbf{S2} \equiv s[i] := 0$ ,  $\mathbf{Q} \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] > 0)$ .

$wp(\text{IfThenElseFi}(\mathbf{B}, \mathbf{S1}, \mathbf{S2}), \mathbf{Q})$

$$\begin{aligned} &\equiv \text{def}(\mathbf{B}) \wedge_L ((\mathbf{B} \wedge wp(\mathbf{S1}, \mathbf{Q})) \vee (\neg \mathbf{B} \wedge wp(\mathbf{S2}, \mathbf{Q}))) \\ &\equiv ((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L ( \\ &\quad (s[i] < 0 \wedge \text{def}(s) \wedge \text{def}(i) \wedge 0 \leq i < |s| \wedge_L Q_{\text{setAt}(s, i, -s[i])}^s) \vee \\ &\quad (s[i] \geq 0 \wedge \text{def}(s) \wedge \text{def}(i) \wedge 0 \leq i < |s| \wedge_L Q_{\text{setAt}(s, i, 0)}^s) \\ &\quad ) \\ &\equiv 0 \leq i < |s| \wedge_L ( \\ &\quad (s[i] < 0 \wedge Q_{\text{setAt}(s, i, -s[i])}^s) \vee \\ &\quad (s[i] \geq 0 \wedge Q_{\text{setAt}(s, i, 0)}^s) \\ &\quad ) \\ &\equiv 0 \leq i < |s| \wedge_L ( \\ &\quad (s[i] < 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |\text{setAt}(s, i, -s[i])| \rightarrow_L \text{setAt}(s, i, -s[i])[j] > 0)) \vee \\ &\quad (s[i] \geq 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |\text{setAt}(s, i, 0)| \rightarrow_L \text{setAt}(s, i, 0)[j] > 0)) \\ &\quad ) \\ &\equiv 0 \leq i < |s| \wedge_L ( \\ &\quad (s[i] < 0 \wedge ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge -s[i] > 0)) \vee \\ &\quad (s[i] \geq 0 \wedge ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge 0 > 0)) \\ &\quad ) \end{aligned}$$

Continúa en la siguiente diapositiva.

# Solución

$wp(\text{IfThenElseFi}(B, S1, S2), Q)$

$$\begin{aligned} &\equiv 0 \leq i < |s| \wedge_L ( \\ &\quad (s[i] < 0 \wedge ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge -s[i] > 0)) \vee \\ &\quad (s[i] \geq 0 \wedge ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge 0 > 0)) \\ &\quad ) \\ &\equiv 0 \leq i < |s| \wedge_L ( \\ &\quad (s[i] < 0 \wedge ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge s[i] \leq 0)) \vee \\ &\quad (s[i] \geq 0 \wedge ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge \text{False})) \\ &\quad ) \end{aligned}$$

Juntamos las dos cotas sobre  $s[i]$  que están en azul (nos quedamos con la más fuerte).

$$\begin{aligned} &\equiv 0 \leq i < |s| \wedge_L ( \\ &\quad (s[i] < 0 \wedge (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0)) \vee \\ &\quad (s[i] \geq 0 \wedge \text{False}) \\ &\quad ) \\ &\equiv 0 \leq i < |s| \wedge_L ( \\ &\quad (s[i] < 0 \wedge (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0)) \vee \\ &\quad \text{False} \\ &\quad ) \\ &\equiv 0 \leq i < |s| \wedge_L s[i] < 0 \wedge (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0). \end{aligned}$$



# ¿Qué hacemos con los ciclos?

¿Cómo calculo la WP de este programa?

```
proc sumar(in s : seq⟨ℤ⟩) : ℤ
  while (i < s.size()) do
    res := res + s[i];
    i := i + 1
  endwhile
```

$$Q \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$$

- A ojo  $\longrightarrow WP = \{res = 0 \wedge i = 0\}$
- Formalmente no existe un Axioma 5, termina siendo una fórmula infinita (detalles en la teórica)
- Sólo voy a poder probar que la tripla  $\{P\} S \{Q\}$  es válida

# Invariante de un ciclo

Dado un ciclo de la forma

```
 $P_c$   
while (B) do  
    S1;  
    S2;  
    // ...  
endwhile  
 $Q_c$ 
```

El **Invariante** del ciclo es

- Un predicado  $I$  que se cumple:
  - Antes de “entrar” en el ciclo
  - Al terminar cada iteración (si se cumplía B)
  - Al salir del ciclo hace que se cumpla la postcondición

# Teorema del Invariante

## Teorema del invariante

Si existe un predicado  $I$  tal que ...

- ❶  $P_c \Rightarrow I$
- ❷  $\{I \wedge B\} S \{I\}$
- ❸  $I \wedge \neg B \Rightarrow Q_c$

entonces el ciclo **while(B)** {**S**} es *parcialmente correcto* respecto de la especificación  $(P_c, Q_c)$ .

Más tarde vemos qué falta para que sea totalmente correcto

# Ejemplo

```
proc sumar(in  $s : seq\langle \mathbb{Z} \rangle$ ) :  $\mathbb{Z}$   
  res := 0  
  i := 0  
  while (i < s.size()) do  
    res := res + s[i];  
    i := i + 1  
  endwhile
```

# Ejemplo

```
proc sumar(in s : seq⟨ℤ⟩) : ℤ
  res := 0
  i := 0
  while (i < s.size()) do
    res := res + s[i];
    i := i + 1
  endwhile
```

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

# Ejemplo

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

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

- $P_c \Rightarrow I$

# Ejemplo

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

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

- $P_c \Rightarrow I$ 
  - $0 \leq i \leq |s| \equiv 0 \leq 0 \leq |s|$  ✓
  - $res = \sum_{j=0}^{i-1} s[j] \equiv 0 = \sum_{j=0}^{0-1} s[j] = 0$  ✓

# Ejemplo

```

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

```

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

- $\{I \wedge B\} \text{ S } \{I\} \leftrightarrow \{I \wedge B\} \rightarrow WP(S, I)$



# Ejemplo

```

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

```

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

- $\{I \wedge B\} S \{I\} \leftrightarrow \{I \wedge B\} \rightarrow WP(S, I)$ 
  - $WP(S, I)$ 

$$\begin{aligned} &\equiv WP(res := res + s[i]; i := i + 1, \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}) \\ &\equiv WP(res := res + s[i], WP(i := i + 1, \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\})) \\ &\equiv WP(res := res + s[i], 0 \leq i + 1 \leq |s| \wedge res = \sum_{j=0}^i s[j]) \\ &\equiv \text{def}(s[i]) \wedge_L (-1 \leq i \leq |s| - 1 \wedge res + s[i] = \sum_{j=0}^i s[j]) \\ &\equiv 0 \leq i < |s| \wedge_L (-1 \leq i \leq |s| - 1 \wedge res = \sum_{j=0}^i s[j] - s[i]) \\ &\equiv 0 \leq i < |s| \wedge_L res = \sum_{j=0}^{i-1} s[j] \text{ (Combino los rangos y resto de la sumatoria)} \end{aligned}$$
  - $\{I \wedge B\} \equiv \{0 \leq i < |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$
  - $\{I \wedge B\} \rightarrow WP(S, I)?$  ✓

# Ejemplo

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

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

- $I \wedge \neg B \Rightarrow Q_C$

# Ejemplo

```

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

```

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

- $I \wedge \neg B \Rightarrow Q_C$ 
  - $I \wedge \neg B \equiv |s| \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]$   
 $\equiv i = |s| \wedge res = \sum_{j=0}^{i-1} s[j]$   
 $\equiv res = \sum_{j=0}^{|s|-1} s[j] \equiv Q_c \checkmark$

# ¿Qué podemos demostrar hasta ahora?

- Correctitud parcial: Probando las hipótesis que vimos hasta acá sabemos que **si el ciclo termina** la tripla de Hoare  $\{P_c\} \text{ S } \{Q_c\}$  es válida
- Falta probar que el ciclo efectivamente termine
- Teorema de Terminación

# Teorema de terminación

Dado un ciclo de la forma

```
 $P_c$   
while (B) do  
    S1;  
    S2;  
    // ...  
endwhile  
 $Q_c$ 
```

La **función variante** es:

- Una función de estados del programa a enteros que cumple:
  - Se achica en cada iteración
  - Si vale cero o menos (y se cumple el invariante), entonces se cumple la postcondición

# Teorema de Terminación

## Teorema de Terminación

Si existe una función  $f_v : \mathbb{V} \rightarrow \mathbb{Z}$  tal que

- ❶  $\{I \wedge B \wedge f_v = v_0\} \text{ S } \{f_v < v_0\},$
- ❷  $I \wedge f_v \leq 0 \rightarrow \neg B,$

entonces la ejecución del ciclo **while B do S endwhile** siempre termina.

- La función  $f_v$  se llama **función variante** del ciclo.
- $\mathbb{V}$  son valores que toman las variables del programa

# Ejemplo

```
proc sumar(in  $s : seq\langle \mathbb{Z} \rangle$ ) :  $\mathbb{Z}$   
  res := 0  
  i := 0  
  while (i < s.size()) do  
    res := res + s[i];  
    i := i + 1  
  endwhile
```

- $P_c \equiv \{res = 0 \wedge i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j]\}$

# Ejemplo

```
proc sumar(in  $s : seq\langle \mathbb{Z} \rangle$ ) :  $\mathbb{Z}$   
  res := 0  
  i := 0  
  while (i < s.size()) do  
    res := res + s[i];  
    i := i + 1  
  endwhile
```

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



# Ejemplo

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

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

- $\{I \wedge B \wedge f_v = v_0\} \mathbf{S} \{f_v < v_0\} \leftrightarrow$   
 $\{I \wedge B \wedge f_v = v_0\} \rightarrow WP(\mathbf{S}, f_v < v_0)$

# Ejemplo

```

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

```

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

- $\{I \wedge B \wedge f_v = v_0\} \mathbf{S} \{f_v < v_0\} \leftrightarrow$   
 $\{I \wedge B \wedge f_v = v_0\} \rightarrow WP(\mathbf{S}, f_v < v_0)$ 
  - $WP(\mathbf{S}, f_v < v_0) \equiv WP(res := res + s[i]; i := i + 1, |s| - i < v_0)$   
 $\equiv WP(res := res + s[i], WP(i := i + 1, |s| - i < v_0))$   
 $\equiv WP(res := res + s[i], |s| - i + 1 < v_0) \equiv \text{def}(s[i]) \wedge_L |s| - (i + 1) < v_0$   
 $\equiv 0 \leq i < |s| \wedge_L |s| - i - 1 < v_0$
  - $\{I \wedge B \wedge f_v = v_0\} \equiv \{0 \leq |s| - 1 \wedge |s| - 1 = v_0 \wedge res = \sum_{j=0}^{i-1} s[j]\}$ 
    - Puedo ignorar lo que corresponde al rango y a  $res$
  - $\{I \wedge B \wedge f_v = v_0\} \rightarrow WP(\mathbf{S}, f_v < v_0) \leftrightarrow |s| - i - 1 < |s| - 1 \leftrightarrow -1 < 0 \checkmark$

# Ejemplo

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

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

- $I \wedge f_v \leq 0 \rightarrow \neg B$

# Ejemplo

```

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

```

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

- $I \wedge f_v \leq 0 \rightarrow \neg B$ 
  - $I \wedge f_v \leq 0 \equiv 0 \leq i \leq |s| \wedge |s| - 1 \leq 0 \wedge res = \sum_{j=0}^{i-1} s[j]$ 
    - Una vez más ignoro lo que corresponde a  $res$  porque no lo necesito para esta demostración
  - $0 \leq i \leq |s| \wedge |s| - i \leq 0 \leftrightarrow i \leq |s| \wedge |s| \leq i \leftrightarrow i = |s|$
  - $i = |s| \rightarrow \neg(i < |s|) \rightarrow \neg B$  ✓

# Ejercicio

**Ejercicio 14.** Dado el siguiente problema

```

proc sumarElementos(in s : array( $\mathbb{Z}$ )) :  $\mathbb{Z}$  {
  requiere {  $|s| \geq 1 \wedge |s| \bmod 2 = 0$  }
  asegura {  $res = \sum_{j=0}^{|s|-1} s[j]$  }
}

```

Dar un invariante y función variante para cada una de estas posibles implementaciones

```

b) res := 0
   i := s.size() - 1
   while (i >= 0) do
     res := res + s[i];
     i := i - 1
   endwhile

```

# Ejercicio

Hagamos la tabla para ver cómo cambian las variables

$i$	$res$
$ s  - 1$	$0$
$ s  - 2$	$s[ s  - 1]$
$ s  - 3$	$s[ s  - 1] + s[ s  - 2]$
$\dots$	$\dots$
$-1$	$\sum_{j=0}^{ s -1} s[j]$

# Ejercicio

## *Solución*

- a)
- $I \equiv \{-1 \leq i < |s| \wedge_L res = \sum_{j=i+1}^{|s|-1} s[j]\}$
  - $f_v = i + 1$

# Ejercicio

**Ejercicio 15.** Dado el siguiente problema

```
proc sumarElementos(in s : array⟨ℤ⟩) : ℤ {
  requiere { |s| ≥ 1 ∧ |s| mód 2 = 0 }
  asegura { res =  $\sum_{j=0}^{|s|-1} s[j]$  }
}
```

Completar el programa para que sean correctos el invariante y la función variante

```
res := 0
i := 0
while [COMPLETAR]
  [COMPLETAR]
endwhile
```

- a)   ▪  $I \equiv \{0 \leq i \leq |s|/2 \wedge_L res = \sum_{j=0}^{i-1} s[j] + \sum_{j=|s|-i}^{|s|-1} s[j]\}$
- $f_v = |s|/2 - i$



# Ejercicio

*solucion*

```
a) res := 0
   i := 0
   while (i < s.size() / 2) do
     res := res + s[i] + s[s.size() - 1 - i];
     i := i + 1
   endwhile
```

# Ejercicio de parcial

Sea el siguiente ciclo con su correspondiente precondition y postcondición:

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

- $P_c \equiv \{i = 0 \wedge s = S_0\}$
- $Q_c \equiv \{|s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = S_0[j] + 1)\}$

- Proponer un invariante de ciclo.
- Demostrar la correctitud parcial del ciclo con ese invariante.
- Proponer una función variante y demostrar que el ciclo termina.

# Solución

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

- $P_c \equiv \{i = 0 \wedge s = S_0\}$
- $Q_c \equiv \{|s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = S_0[j] + 1)\}$
- $B \equiv \{i < |s|\}$
- $I \equiv \{0 \leq i \leq |s| \wedge \frac{|s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j])}{\wedge_L} \}$
- $f_v = |s| - i$

# Solución

- $P_c \equiv \{i = 0 \wedge s = S_0\}$
- $I \equiv \{0 \leq i \leq |s| \wedge |s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j])\}$

- $P_c \Rightarrow I$

- $i = 0 \Rightarrow 0 \leq i \leq |s| \equiv 0 \leq 0 \leq |s|$  ✓
- $s = S_0 \Rightarrow |s| = |S_0|$  ✓
- $i = 0 \Rightarrow (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \equiv (\forall j : \mathbb{Z})(0 \leq j < 0 \rightarrow_L s[j] = S_0[j] + 1)$  ✓
- $i = 0 \wedge s = S_0 \Rightarrow (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j]) \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = S_0[j]) \equiv s = S_0$  ✓

# Solución

- $I \equiv \{0 \leq i \leq |s| \wedge |s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j])\}$
- $B \equiv i < |s|$
- $C \equiv s[i] := s[i] + 1; i := i + 1$

$$\bullet \{I \wedge B\} C \{I\} \iff I \wedge B \rightarrow wp(C, I)$$

- $wp(C, I) \equiv wp(s[i] := s[i] + 1, wp(i := i + 1, I)) \equiv wp(s[i] := s[i] + 1, I_{i+1}^i)$   
 $\equiv \text{def}(setAt(s, i, s[i] + 1)) \wedge_L (I_{i+1}^i)_{setAt(s, i, s[i] + 1)}$

Ojo: solo reemplazamos  $s$  por  $setAt$ ;  $S_0$  permanece igual.

$$\begin{aligned} &\equiv 0 \leq i < |s| \wedge_L 0 \leq i + 1 \leq |setAt(s, i, s[i] + 1)| \wedge |setAt(s, i, s[i] + 1)| = |S_0| \wedge_L \\ &(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L setAt(s, i, s[i] + 1)[j] = S_0[j] + 1) \wedge \\ &(\forall j : \mathbb{Z})(i + 1 \leq j < |setAt(s, i, s[i] + 1)| \rightarrow_L setAt(s, i, s[i] + 1)[j] = S_0[j]) \end{aligned}$$

Cambiamos  $setAt(s, i, s[i] + 1)$  por  $s$  en las cláusulas que amerite.

$$\begin{aligned} &\equiv 0 \leq i < |s| \wedge_L 0 \leq i + 1 \leq |s| \wedge |s| = |S_0| \wedge_L \\ &(\forall j : \mathbb{Z})(0 \leq j < i + 1 \rightarrow_L s[j] = S_0[j] + 1) \wedge \\ &(\forall j : \mathbb{Z})(i + 1 \leq j < |s| \rightarrow_L s[j] = S_0[j]) \end{aligned}$$

Juntamos las cotas sobre  $i$ . Además, separamos el caso  $j = i$  del  $\forall$ .

$$\begin{aligned} &\equiv 0 \leq i < |s| \wedge |s| = |S_0| \wedge_L \\ &(\forall j : \mathbb{Z})(\underline{0 \leq j < i} \rightarrow_L s[j] = S_0[j] + 1) \wedge \underline{setAt(s, i, s[i] + 1)[i] = S_0[i] + 1} \\ &\wedge (\forall j : \mathbb{Z})(i + 1 \leq j < |s| \rightarrow_L s[j] = S_0[j]) \end{aligned}$$

# Solución

- $I \equiv \{0 \leq i \leq |s| \wedge |s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j])\}$
- $B \equiv i < |s|$
- $C \equiv s[i] := s[i] + 1; i := i + 1$

$$\bullet \{I \wedge B\} C \{I\} \iff I \wedge B \rightarrow wp(C, I)$$

- $wp(C, I)$  (Continuando lo de la diapo anterior)

Juntamos las cotas sobre  $i$ . Además, separamos el caso  $j = i$  del  $\forall$ .

$$\equiv 0 \leq i < |s| \wedge |s| = |S_0| \wedge_L$$

$$(\forall j : \mathbb{Z})(\underline{0 \leq j < i} \rightarrow_L \text{setAt}(s, i, s[i] + 1)[j] = S_0[j] + 1) \wedge \underline{\text{setAt}(s, i, s[i] + 1)[i] = S_0[i] + 1} \\ \wedge (\forall j : \mathbb{Z})(i + 1 \leq j < |s| \rightarrow_L s[j] = S_0[j])$$

Nos quitamos de encima los últimos  $\text{setAt}$ .

$$\equiv 0 \leq i < |s| \wedge |s| = |S_0| \wedge_L$$

$$(\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L \underline{s[j] = S_0[j] + 1}) \wedge \underline{s[i] + 1 = S_0[i] + 1} \wedge \\ (\forall j : \mathbb{Z})(i + 1 \leq j < |s| \rightarrow_L s[j] = S_0[j])$$

Juntamos las últimas dos cláusulas en un solo  $\forall$ .

$$\equiv 0 \leq i < |s| \wedge |s| = |S_0| \wedge_L$$

$$(\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(\underline{i \leq j < |s|} \rightarrow_L s[j] = S_0[j])$$

- Es sencillo ver que si juntamos las cotas sobre  $i$  en  $I \wedge B$  nos queda exactamente la misma expresión que obtuvimos al calcular la  $wp$ . Por lo tanto, podemos afirmar que vale la implicación  $I \wedge B \rightarrow wp(C, I)$ . ✓

# Solución

- $I \equiv \{0 \leq i \leq |s| \wedge |s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j])\}$
- $\neg B \equiv i \geq |s|$
- $Q_c \equiv \{|s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] = S_0[j] + 1)\}$

- $I \wedge \neg B \Rightarrow Q_C$ 
  - $|s| = |S_0| \Rightarrow |s| = |S_0|$  ✓
  - Obs.:  $i \leq |s|$  (De  $I$ )  $\wedge i \geq |s|$  (De  $\neg B$ )  $\Rightarrow i = |s|$
  - $i = |s| \wedge (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \Rightarrow (\forall j : \mathbb{Z})(\underline{0 \leq j < |s|} \rightarrow_L s[j] = S_0[j] + 1)$  ✓

# Solución

- $I \equiv \{0 \leq i \leq |s| \wedge |s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j])\}$
  - $f_v = |s| - i$
  - $\neg B \equiv i \geq |s|$
- 
- $I \wedge f_v \leq 0 \Rightarrow \neg B$ 
    - $f_v = |s| - i \wedge f_v \leq 0 \Rightarrow |s| - i \leq 0 \equiv |s| \leq i$  ✓



# Solución

- $I \equiv \{0 \leq i \leq |s| \wedge |s| = |S_0| \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] = S_0[j] + 1) \wedge (\forall j : \mathbb{Z})(i \leq j < |s| \rightarrow_L s[j] = S_0[j])\}$
- $B \equiv i < |s|$
- $f_v = |s| - i$
- $C \equiv s[i] := s[i] + 1; i := i + 1$

- $\{I \wedge B \wedge f_v = v_0\} C \{f_v < v_0\} \iff I \wedge B \wedge f_v = v_0 \rightarrow wp(C, f_v < v_0)$ 
  - $wp(C, f_v < v_0)$   
 $\equiv wp(s[i] := s[i] + 1, wp(i := i + 1, f_v < v_0))$   
 $\equiv wp(s[i] := s[i] + 1, (f_v < v_0)_{i+1}^i)$   
 $\equiv \text{def}(\text{setAt}(s, i, s[i] + 1)) \wedge_L ((f_v < v_0)_{i+1}^i)^s_{\text{setAt}(s, i, s[i] + 1)}$   
 $\equiv 0 \leq i < |s| \wedge_L |\text{setAt}(s, i, s[i] + 1)| - (i + 1) < v_0$   
 $\equiv \underline{0 \leq i < |s| \wedge_L |s| - i - 1 < v_0}$
  - $I \wedge B \wedge f_v = v_0 \rightarrow wp(C, f_v < v_0)$ 
    - $0 \leq i \leq |s|$  (de  $I$ ) y  $B \Rightarrow 0 \leq i < |s|$  ✓
    - $f_v = |s| - i \wedge f_v = v_0 \Rightarrow |s| - i - 1 < v_0 \equiv f_v - 1 < v_0$  ✓