## Verificación de programas II: Teorema del Invariante

Román Gorojovsky

Algoritmos y Estructuras de Datos

4 de septiembre de 2024



#### Plan del día

#### Plan del día

- Precondición más débil de ciclos
- Teorema del Invariante
- Teorema de Terminación
- Un ejercicio de un parcial
- Correctitud de programas

#### Precondición más débil

#### Precondición más débil – Idea informal

Es la P que permite que el programa  ${\bf S}$  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:  $wp(\mathbf{x} := \mathbf{E}, Q) \equiv def(E) \wedge_L Q_E^x$
- Axioma 2:  $wp(\mathsf{skip}, Q) \equiv Q$
- Axioma 3:  $wp(S1; S2, Q) \equiv wp(S1, wp(S2, Q))$
- Axioma 4: Si S = if B then S1 else S2 endif, entonces

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

## ¿Qué hacemos con los ciclos?

¿Cómo calculo la WP de este programa?

```
\operatorname{proc\ sumar}(\operatorname{in} s: \operatorname{seq}\langle \mathbb{Z}\rangle): \mathbb{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

## ¿Qué hacemos con los ciclos?

¿Cómo calculo la WP de este programa?

```
proc sumar(in s: seq\langle \mathbb{Z} \rangle): \mathbb{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  $\longrightarrow WP = \{res = 0 \land i = 0\}$
- ¿Formalmente? ¿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

```
while (B) do

$1;

$2;

// ...
```

#### El Invariante del ciclo es

- Un predicado *I* que se cumple:
  - Antes de "entrar" en el ciclo, es decir, antes de cada iteración
  - Al terminar cada iteración (si se cumplía B)

#### Teorema del Invariante

#### Teorema del invariante

Si existe un predicado I tal que ...

- $\mathbf{0} P_c \Rightarrow \mathbf{I}$
- **2**  $\{I \land B\} \ S \ \{I\}$

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

#### Teorema del Invariante

#### Teorema del invariante

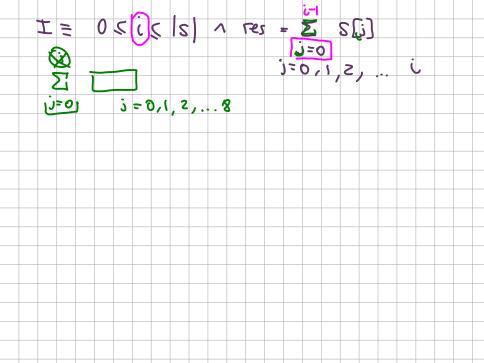
Si existe un predicado I tal que ...

- $\mathbf{0} P_c \Rightarrow \mathbf{1}$
- **2**  $\{I \land B\}$  S  $\{I\}$

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

 $T = \frac{\text{itemciones}}{0} \quad \text{itemciones} \quad \text{$ 



```
proc sumar(in s : seq\langle \mathbb{Z} \rangle) : \mathbb{Z}

res := 0

i := 0

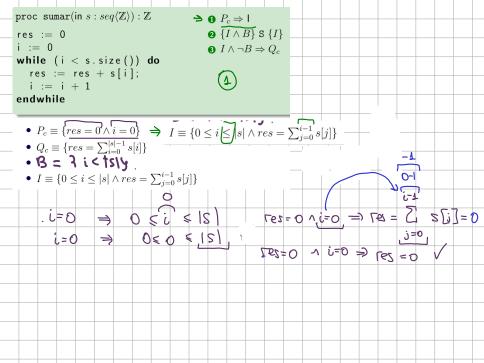
while (i < s.size()) do

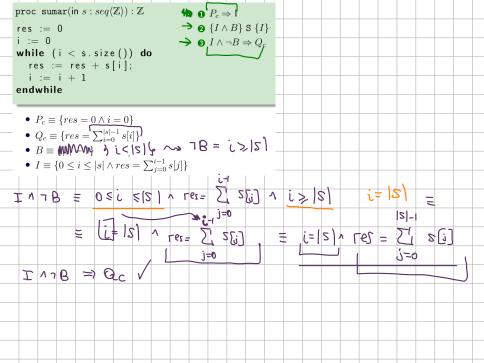
res := res + s[i];

i := i + 1

endwhile
```

- $P_c \equiv \{res = 0 \land i = 0\}$
- $Q_c \equiv \{ res = \sum_{i=0}^{|s|-1} s[i] \}$
- $\bullet \ B \equiv \{|s|-1\}$
- $I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$





$$\frac{1}{2} + \frac{1}{2} = \frac{1}$$

• Λ	vioma	1. 2020	v I	F (0)	= def(	F\ ∧ .	O <sup>z</sup>											
• Axioma 1: $wp(\mathbf{x} := \mathbf{E}, Q) \equiv \operatorname{def}(E) \wedge_L Q_E^*$ • Axioma 2: $wp(\operatorname{skip}, Q) \equiv Q$ • Axioma 3: $wp(\operatorname{S1}; \operatorname{S2}, Q) \equiv wp(\operatorname{S1}, wp(\operatorname{S2}, Q))$ • Axioma 4: Si S = if B then S1 else S2 endif, entonces																		
• A>																		H
$wp(\mathbf{S}, Q) \equiv def(B) \wedge_L \left( (B \wedge wp_{AB}) \right) $																		-
																		L
																		L
																	_	L
																		L

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

• 
$$P_c \equiv \{res = 0 \land i = 0\}$$

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

• 
$$B \equiv \{|s| - 1\}$$

• 
$$I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$$

• 
$$P_c \Rightarrow 1$$

• 
$$0 \le i \le |s| \equiv 0 \le 0 \le |s|$$

• 
$$res = \sum_{j=0}^{i-1} s[j] \equiv 0 = \sum_{j=0}^{0-1} s[j] = 0$$
  $\checkmark$ 

## Eiemplo

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

• 
$$P_c \equiv \{res = 0 \land i = 0\}$$

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

• 
$$B \equiv \{|s| - 1\}$$

• 
$$I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$$

• 
$$\{I \wedge B\}$$
 S  $\{I\}$ 

• 
$$I \wedge B \equiv 0 \le i_0 < |s| \wedge res_0 = \sum_{j=0}^{i_0-1} s[j]$$

• 
$$i = i_0 + 1 \Rightarrow 0 \le i_0 + 1 < |s| \rightarrow 0 \le i \le |s| \checkmark$$

• 
$$res = res_0 + s[i_0] \Rightarrow res = \left(\sum_{j=0}^{i_0-1} s[j]\right) + s[i_0]$$
  
 $\rightarrow res = \sum_{j=0}^{i_0} s[j] \rightarrow res = \sum_{j=0}^{i-1} s[j] \checkmark$ 

$$\to res = \sum_{j=0}^{i_0} s[j] \to res = \sum_{j=0}^{i-1} s[j]$$
  $\checkmark$ 

## Eiemplo

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

- $P_c \equiv \{res = 0 \land i = 0\}$
- $Q_c \equiv \{res = \sum_{i=0}^{|s|-1} s[i]\}$ 
  - $B \equiv \{|s| 1\}$
- $I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$
- $\{I \land B\} \ S \ \{I\} \leftrightarrow \{I \land B\} \rightarrow WP(S,I)$ 
  - WP(S, I)

$$\equiv WP(\text{res} := \text{res} + \text{s[i]}; i := i + 1, \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\})$$
  
 $\equiv WP(\text{res} := \text{res} + \text{s[i]}, WP(i := i + 1, \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}))$ 

$$\equiv WP(\text{res} := \text{res} + \text{s[i]}, 0 \le i+1 \le |s| \land res = \sum_{j=0}^{i} s[j])$$

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

• 
$$\{I \land B\} \equiv \{0 \le |s| - 1 \land res = \sum_{j=0}^{i-1} s[j]\}$$

•  $i\{I \land B\} \rightarrow WP(S,I)? \checkmark$ 

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

• 
$$P_c \equiv \{res = 0 \land i = 0\}$$

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

• 
$$B \equiv \{|s| - 1\}$$

• 
$$I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$$

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

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

(Sólo la primer parte)

#### Práctica 3 segunda parte:

- Ejercicio 6
- Ejercicio 11.c

#### (Sólo los tres primeros pasos)

```
Eiercicio 6. Dado el siguiente problema
proc sumarElementos (in s: array < \mathbb{Z} >) : \mathbb{Z} {
        requiere \{|s| \ge 1 \land |s| \mod 2 = 0\}
        asegura \{res = \sum\limits_{i=n}^{|s|-1} s[j]\}
```

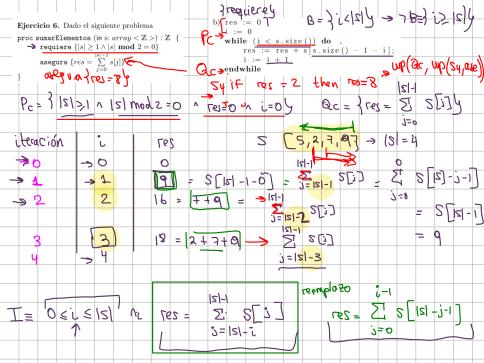
Dar un invariante y función variante para cada una de estas posibles implementaciones

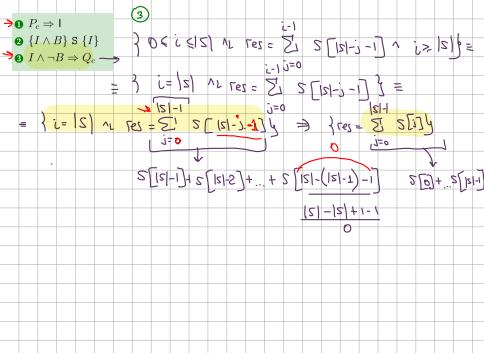
```
a) res := 0
  i := 0
  while (i < s.size()) do
       res := res + s[i];
  endwhile
```

endwhile

endwhile

$$I = I \qquad \text{in } I =$$





#### (Sólo los tres primeros pasos)

**Ejercicio 11.** Dados los siguientes ciclos y sus respectivas precondición  $(P_c)$  y poscondición  $(Q_c)$ .

- 1. Proponer un invariante (I) y una función variante  $(f_v)$  para el ciclo
- 2. Demostrar los siguientes pasos de la demostración de correctitud del ciclo
  - I)  $P_c \to I$
  - II)  $(I \wedge \neg B) \rightarrow Q_c$
  - III)  $(I \wedge f_v < 0) \rightarrow \neg B$
- c)  $P_c \equiv \{i = |s| 1 \land res = 0\}$

while  $i \geq 0$  do

$$res := res + s[i] + 1;$$

$$i := i - 1;$$

end

$$Q_c \equiv \{ res = |s| + \sum_{j=0}^{|s|-1} s[j] \}$$

# Nos tomemos 15 minutos que

(Si es que no nos tomamos el recreo antes)

llegamos a la mitad

## ¿Qué podemos demostrar hasta ahora?

• Correctitud parcial: Probando las hipótesis que vimos hasta acá sabemos que si el ciclo termina la tripla de Hoare  $\{P_c\}$  S  $\{Q_c\}$  es válida

### ¿Qué podemos demostrar hasta ahora?

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

## ¿Qué podemos demostrar hasta ahora?

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

#### Teorema de Terminación

Si existe una función  $fv:\mathbb{V}\to\mathbb{Z}$  tal que

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

#### Teorema de Terminación

Si existe una función  $fv:\mathbb{V}\to\mathbb{Z}$  tal que

- **1**  $\{I \wedge B \wedge f_v = v_0\}$  S  $\{f_v < v_0\}$ ,
- $2 I \wedge f_v \leq 0 \rightarrow \neg B,$

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

#### Teorema de Terminación

Si existe una función  $fv: \mathbb{V} \to \mathbb{Z}$  tal que

- **1**  $\{I \wedge B \wedge f_v = v_0\}$  S  $\{f_v < v_0\}$ ,
- $2 I \wedge f_v \leq 0 \rightarrow \neg B,$

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

• La función  $f_v$  se llama función variante del ciclo.

#### Teorema de Terminación

Si existe una función  $fv:\mathbb{V}\to\mathbb{Z}$  tal que

- **1**  $\{I \wedge B \wedge f_v = v_0\}$  S  $\{f_v < v_0\}$ ,
- $2 I \wedge f_v \leq 0 \rightarrow \neg B,$

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

- La función  $f_v$  se llama función variante del ciclo.
- V son valores que toman las variables del programa

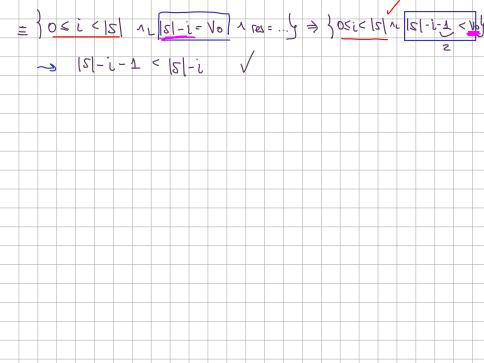
```
proc sumar(in s : seq\langle \mathbb{Z} \rangle) : \mathbb{Z}
                                 (1) TAB -> fy &crecer
                                 (2) I ~ FUED ~> 7B
res := 0
i := 0
                                        cuando fuso salpo el cido
while (i < s.size()) do
                                  fu= 151-6
   res := res + s[i];
   i := i + 1
endwhile
```

• 
$$P_c \equiv \{res = 0 \land i = 0\}$$

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

$$\bullet \ B \equiv \{|s|-1\}$$

• 
$$I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$$



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

• 
$$P_c \equiv \{res = 0 \land i = 0\}$$

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

• 
$$B \equiv \{|s| - 1\}$$

• 
$$I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$$

• 
$$f_v = |s| - i$$

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

• 
$$WP(\mathbf{S}, f_v < v_0) \equiv WP(\mathbf{res} := \mathbf{res} + \mathbf{s[i]}; i := i + 1, |s| - i < v_0)$$
  
 $\equiv WP(\mathbf{res} := \mathbf{res} + \mathbf{s[i]}, WP(i := i + 1, |s| - i < v_0))$   
 $\equiv WP(\mathbf{res} := \mathbf{res} + \mathbf{s[i]}, |s| - i + 1 < v_0) \equiv |s| - i + 1 < v_0$ 

• 
$$\{I \land B \land f_v = v_0\} \equiv \{0 \le |s| - 1 \land |s| - 1 = v_0 \land res = \sum_{j=0}^{i-1} s[j]\}$$
  
•  $\{I \land B \land f_v = v_0\} \rightarrow WP(\mathbf{S}, f_v < v_0) \leftrightarrow |s| - i + 1 < |s| - 1 \leftrightarrow 1 < 0 \checkmark$ 

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

• 
$$P_c \equiv \{res = 0 \land i = 0\}$$

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

• 
$$B \equiv \{|s|-1\}$$

• 
$$I \equiv \{0 \le i \le |s| \land res = \sum_{j=0}^{i-1} s[j]\}$$

• 
$$f_v = |s| - i$$

• 
$$I \wedge f_v < 0 \rightarrow \neg B$$

• 
$$I \wedge f_v \leq 0 \equiv 0 \leq i \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j] \wedge |s| - 1 \leq 0$$
  
  $\equiv 0 \leq i \leq |s| \wedge |s| - i \rightarrow i \leq |s| \wedge |s| \leq i$ 

• 
$$i \le |s| \land |s| \le i \leftrightarrow i = |s|$$

• 
$$i = |s| \rightarrow \neg B \checkmark$$

(La parte que faltaba)

#### Práctica 3 segunda parte:

- Ejercicio 6
- Ejercicio 11.c

## Ejercicio de parcial

#### E4. Correctitud del ciclo [30 pts]

Dado el siguiente programa con su especificación

```
\begin{split} P_c &\equiv \{n > 0 \land n \; mod \; 2 = 0 \land i = 1 \land res = 1\} \\ &\quad \text{While ( } i \; <= \; n/2 \text{)} \{ \\ &\quad \text{res } := \; \text{res } * \; i \; * \; (n+1-i \; ) \, ; \\ &\quad \text{i } := \; i+1; \\ &\quad \} \\ &\quad Q_c &\equiv \{res = n!\} \end{split}
```

Contamos con el siguiente invariante, que sabemos que es incorrecto:

$$I \equiv \{1 \leq i \leq n/2 + 1 \wedge res = \prod_{j=1}^{2(i-1)} j\}$$

- a) Señale qué axiomas del teorema del invariante no se cumplen. Justifique con palabras en forma precisa.
- b) Escriba un invariante que resulte correcto.
- c) Proponga una función variante y demuestre formalmente que es correcta.