

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 **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

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

¿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? ¿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  
    S1;  
    S2;  
    // ...  
endwhile
```

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 ...

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

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

Ejemplo

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

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

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

Ejemplo

```

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

```

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

- $\{I \wedge B\} \text{ S } \{I\}$
 - $I \wedge B \equiv 0 \leq i_0 < |s| \wedge res_0 = \sum_{j=0}^{i_0-1} s[j]$
 - $i = i_0 + 1 \Rightarrow 0 \leq i_0 + 1 < |s| \rightarrow 0 \leq i \leq |s|$ ✓
 - $res = res_0 + s[i_0] \Rightarrow res = \left(\sum_{j=0}^{i_0-1} s[j] \right) + s[i_0]$
 $\rightarrow res = \sum_{j=0}^{i_0} s[j] \rightarrow res = \sum_{j=0}^{i-1} s[j]$ ✓

Ejemplo

```

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

```

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

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

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

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

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

$$\equiv 0 \leq i + 1 \leq |s| \wedge res = \sum_{j=0}^{i-1} s[j])$$
 - $\{I \wedge B\} \equiv \{0 \leq |s| - 1 \wedge res = \sum_{j=0}^{|s|-1} s[j]\}$
 - $i\{I \wedge B\} \rightarrow WP(S, I)? \checkmark$

Ejemplo

```

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

```

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

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

Ejercicios

(Sólo la primer parte)

Práctica 3 segunda parte:

- Ejercicio 6
- Ejercicio 11.c

Ejercicios

(Sólo los tres primeros pasos)

Ejercicio 6. 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

a) `res := 0`
`i := 0`
while (`i > s.size()`) **do**
 `res := res + s[i];`
 `i := i + 1`
endwhile

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

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

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

Ejercicios

(Sólo los tres primeros pasos)

Ejercicio 11. Dados los siguientes ciclos y sus respectivas precondition (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

$$\text{I) } P_c \rightarrow I$$

$$\text{II) } (I \wedge \neg B) \rightarrow Q_c$$

$$\text{III) } (I \wedge f_v \leq 0) \rightarrow \neg B$$

$$\text{c) } P_c \equiv \{i = |s| - 1 \wedge res = 0\}$$

while $i \geq 0$ **do**

$$\begin{array}{|l} res := res + s[i] + 1; \\ i := i - 1; \end{array}$$

end

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

¿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

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
```

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

Ejemplo

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

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

- $\{I \wedge B \wedge f_v = v_0\} \text{ S } \{f_v < v_0\}$
 - $I \wedge B \wedge f_v = v_0 \equiv 0 \leq |s| - 1 \wedge res = \sum_{j=0}^{i-1} s[j] \wedge |s| - i = v_0$
 - Tras ejecutar S, $f_v = |s| - i - 1 \Rightarrow f_v < |s| - i \Rightarrow f_v < v_0$ ✓

Ejemplo

```

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

```

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

- $\{I \wedge B \wedge f_v = v_0\} \text{ S } \{f_v < v_0\} \leftrightarrow$
 $\{I \wedge B \wedge f_v = v_0\} \rightarrow WP(\text{S}, f_v < v_0)$
 - $WP(\text{S}, f_v < v_0) \equiv WP(\text{res} := \text{res} + s[i]; i := i + 1, |s| - i < v_0)$
 $\equiv WP(\text{res} := \text{res} + s[i], WP(i := i + 1, |s| - i < v_0))$
 $\equiv WP(\text{res} := \text{res} + s[i], |s| - i + 1 < v_0) \equiv |s| - i + 1 < v_0$
 - $\{I \wedge B \wedge f_v = v_0\} \equiv \{0 \leq |s| - 1 \wedge |s| - 1 = v_0 \wedge res = \sum_{j=0}^{i-1} s[j]\}$
 - $\{I \wedge B \wedge f_v = v_0\} \rightarrow WP(\text{S}, f_v < v_0) \leftrightarrow |s| - i + 1 < |s| - 1 \leftrightarrow 1 < 0 \checkmark$

Ejemplo

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

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

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

Ejercicios

(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

$$P_c \equiv \{n > 0 \wedge n \bmod 2 = 0 \wedge i = 1 \wedge res = 1\}$$

```
While ( i <= n/2 ) {
    res := res * i * (n+1-i);
    i := i+1;
}
```

$$Q_c \equiv \{res = n!\}$$

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

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

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