

# Precondición más débil de ciclos

Algoritmos y Estructuras de Datos I

## Repaso: Triplas de Hoare

- Consideremos la siguiente tripla de Hoare:

$$\{P\} \text{ S } \{Q\}.$$

- Esta tripla es **válida** si se cumple que:
  1. Si el programa  $S$  comienza en un estado que cumple  $P$  ...
  2. ... entonces termina luego de un número finito de pasos ...
  3. ... Y además en un estado que cumple  $Q$ .

# Repaso: Lenguaje SmallLang

- ▶ Definimos un lenguaje imperativo basado en **variables** y las siguientes instrucciones:
  1. **Nada**: Instrucción **skip** que no hace nada.
  2. **Asignación**: Instrucción **x := E**.
- ▶ Además, tenemos las siguientes estructuras de control:
  1. **Secuencia**: **S1; S2** es un programa, si **S1** y **S2** son dos programas.
  2. **Condicional**: **if B then S1 else S2 endif** es un programa, si **B** es una expresión lógica y **S1** y **S2** son dos programas.
  3. **Ciclo**: **while B do S endwhile** es un programa, si **B** es una expresión lógica y **S** es un programa.

## Repaso: Precondición más débil

- **Definición.** La **precondición más débil** de un programa  $S$  respecto de una postcondición  $Q$  es el predicado  $P$  más débil posible tal que  $\{P\}S\{Q\}$ .
- **Notación.**  $wp(S, Q)$ .
- **Teorema:** Decimos que  $\{P\} S \{Q\}$  es válida sii  $P \Rightarrow_L wp(S, Q)$

## Repaso: Axiomas wp

- ▶ **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.**  $wp(S1; S2, Q) \equiv wp(S1, wp(S2, Q)).$
- ▶ **Axioma 4.**  $wp(\text{if } B \text{ then } S1 \text{ else } S2 \text{ endif}, Q) \equiv$

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

- ▶ **Observación:**  $wp(b[i] := E, Q) \equiv wp(b := \text{setAt}(b, i, E), Q)$

## ¿Cuál es la precondition más débil?

{???

while (x>0) do

  x := x - 1

endwhile

{x = 0}

$$wp(\text{while } \dots, x = 0) \equiv x \geq 0$$

## ¿Cuál es la precondition más débil?

{???

i := 0;

while (x<5) do

  x := x + 1;

  i := i + 1

endwhile

{x = 5 ∧ i = 5}

$wp(i:=0; \text{ while } \dots, x = 5 \wedge i = 5) \equiv x = 0$

## ¿Cuál es la precondition más débil?

{???

```
while (x==5) do
```

```
  x := 5
```

```
endwhile
```

{ $x \neq 5$ }

$$wp(\text{while } \dots, x \neq 5) \equiv x \neq 5$$



¿Es válida la siguiente tripla de Hoare?

$\{n \geq 0 \wedge i = 1 \wedge s = 0\}$

while (i <= n) do

  s := s + i;

  i := i + 1

endwhile

$\{s = \sum_{k=1}^n k\}$

## Precondición más débil de un ciclo

- Supongamos que tenemos el ciclo **while B do S endwhile**.
- **Definición.** Definimos  $H_k(Q)$  como el predicado que define el conjunto de estados a partir de los cuales la ejecución del ciclo termina en **exactamente**  $k$  iteraciones:

$$\begin{aligned}H_0(Q) &\equiv \text{def}(B) \wedge \neg B \wedge Q, \\H_{k+1}(Q) &\equiv \text{def}(B) \wedge B \wedge wp(S, H_k(Q)) \quad \text{para } k \geq 0.\end{aligned}$$

- **Propiedad:** Si el ciclo realiza a lo sumo  $k$  iteraciones, entonces

$$wp(\text{while } B \text{ do } S \text{ endwhile}, Q) \equiv \bigvee_{i=0}^k H_i(Q)$$

## Ejemplo

{???

```
while (0 < i && i < 3) do
```

```
  i := i + 1
```

```
endwhile
```

{i = 3}

- ▶ A lo sumo, se va a ejecutar 2 veces el cuerpo del ciclo
- ▶ ¿Cuál es la precondition más débil?

$$\begin{aligned} & wp(\text{while } 0 < i < 3 \text{ do } i := i + 1 \text{ endwhile}, i = 3) \\ \equiv & \bigvee_{i=0}^2 H_i(i = 3) \\ \equiv & H_0(i = 3) \vee H_1(i = 3) \vee H_2(i = 3) \\ \equiv & i = 1 \vee i = 2 \vee i = 3 \end{aligned}$$

## Otro ejemplo

{???

```
while (0<i && i<n) do
```

```
  i := i +1
```

```
endwhile
```

{ $i \geq 0$ }

- ▶ ¿Cuántas veces se va a ejecutar el cuerpo del ciclo?
- ▶ ¿Podemos usar la propiedad anterior para conocer la precondition más débil?
- ▶ **¡No!** Porque no podemos fijar a priori una cota superior a la cantidad de iteraciones que va a realizar el ciclo.

## Precondición más débil de un ciclo

- **Intuitivamente:**  $wp(\text{while } B \text{ do } S \text{ endwhile}, Q)$  tiene que ser una fórmula lógica capaz de capturar todos los estados tales que, luego de ejecutar el ciclo una cantidad arbitraria de veces, vale  $Q$ .
- **Axioma 5:**

$$wp(\text{while } B \text{ do } S \text{ endwhile}, Q) \equiv (\exists_{i \geq 0})(H_i(Q))$$

## Precondición más débil de un ciclo

- Ahora tratemos de usar el **Axioma 5**:

$$\begin{aligned}wp(\text{while } B \text{ do } S \text{ endwhile}, Q) \\ &\equiv (\exists_{i \geq 0}) H_i(Q) \\ &\equiv H_0(Q) \vee H_1(Q) \vee H_2(Q) \vee \dots \\ &\equiv \bigvee_{i=0}^{\infty} (H_i(Q)) \\ &\quad \text{¡Es una fórmula infinitaria!}\end{aligned}$$

- Por lo tanto, no podemos usar mecánicamente el **Axioma 5** para demostrar la corrección de un ciclo con una cantidad no acotada **a priori** de iteraciones :(

## Recap: Teorema del Invariante

- **Teorema.** Si  $\text{def}(B)$  y existe un predicado  $I$  tal que

1.  $P_C \Rightarrow I$ ,
2.  $\{I \wedge B\} S \{I\}$ ,
3.  $I \wedge \neg B \Rightarrow Q_C$ ,

... y **el ciclo termina**, entonces la siguiente tripla de Hoare es válida:

$$\{P_C\} \text{ while } B \text{ do } S \text{ endwhile } \{Q_C\}$$

- Esta observación es un **teorema** que se deduce de la definición anterior.
- Las condiciones **1-3** garantizan la **corrección parcial** del ciclo (la hipótesis de terminación es necesaria para garantizar corrección).

## Ejemplo: suma de índices

- Sea la siguiente tripla de Hoare:

$$\{n \geq 0 \wedge i = 1 \wedge s = 0\}$$

while (i <= n) do

    s = s + i;

    i = i + 1;

endwhile

$$\{s = \sum_{k=1}^n k\}$$

- Habíamos identificado los predicados necesarios para aplicar el Teorema del Invariante:

- $P_C \equiv n \geq 0 \wedge i = 1 \wedge s = 0$

- $Q_C \equiv s = \sum_{k=1}^n k$

- $B \equiv i \leq n$

- $I \equiv 1 \leq i \leq n + 1 \wedge s = \sum_{k=1}^{i-1} k$



## Repaso: $P_C \Rightarrow I$

$$\begin{aligned}P_C &\equiv n \geq 0 \wedge i = 1 \wedge s = 0 \\&\Rightarrow 1 \leq i \leq n + 1 \wedge s = 0 \\&\Rightarrow 1 \leq i \leq n + 1 \wedge s = \sum_{k=1}^0 k \\&\Rightarrow 1 \leq i \leq n + 1 \wedge s = \sum_{k=1}^{i-1} k \\&\equiv I \checkmark\end{aligned}$$

## Repaso: $I \wedge \neg B \Rightarrow Q_C$

$$\begin{aligned} I \wedge \neg B &\equiv 1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{i-1} k \wedge \neg(i \leq n) \\ &\equiv 1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{i-1} k \wedge i > n \\ &\Rightarrow 1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{i-1} k \wedge i = n+1 \\ &\Rightarrow 1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{n+1-1} k \wedge i = n+1 \\ &\Rightarrow s = \sum_{k=1}^n k \equiv Q_C \checkmark \end{aligned}$$

$$\{I \wedge B\} S \{I\}$$

Para demostrar  $\{I \wedge B\} S \{I\}$  tenemos que probar que:

$$I \wedge B \Rightarrow wp(S, I)$$

$$\begin{aligned}
 & wp(s:=s+i; i:=i+1, 1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{i-1} k) \\
 \equiv & wp(s:=s+i, wp(i:=i+1, 1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{i-1} k)) \\
 \equiv & wp(s:=s+i, def(i+1) \wedge_L (1 \leq i+1 \leq n+1 \wedge s = \sum_{k=1}^{i+1-1} k)) \\
 \equiv & wp(s:=s+i, 1 \leq i+1 \leq n+1 \wedge s = \sum_{k=1}^{i+1-1} k) \\
 \equiv & def(s+i) \wedge_L (1 \leq i+1 \leq n+1 \wedge s+i = \sum_{k=1}^{i+1-1} k)
 \end{aligned}$$

$$\{I \wedge B\} S \{I\}$$

$$\equiv 0 \leq i \leq n \wedge s + i = \sum_{k=1}^i k$$

$$\equiv 0 \leq i \leq n \wedge s = \left(\sum_{k=1}^i k\right) - i$$

$$\equiv 0 \leq i \leq n \wedge s = \sum_{k=1}^{i-1} k$$

- Luego de simplificar, nos falta probar que:

$$\underbrace{\left(1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{i-1} k\right)}_I \wedge \underbrace{i \leq n}_B \Rightarrow \underbrace{\left(0 \leq i \leq n \wedge s = \sum_{k=1}^{i-1} k\right)}_{wp(S,I)}$$

- Lo cual es trivialmente cierto.
- Por lo tanto podemos concluir que  $\{I \wedge B\} S \{I\}$  es una tripla de Hoare válida (i.e., verdadera)

## Ejemplo: suma de índices

- Habiendo probado las hipótesis del Teorema del Invariante podemos decir que **si el ciclo siempre termina**, entonces la siguiente tripla de Hoare es válida:

$$\{n \geq 0 \wedge i = 1 \wedge s = 0\}$$

while (i <= n) do

    s = s + i;

    i = i + 1;

endwhile

$$\{s = \sum_{k=1}^n k\}$$

- **Pero** ..., ¡todavía no probamos que el ciclo siempre termina!
- ¿Cómo podemos probar si dada una precondición, un ciclo siempre termina?
  - Para eso tenemos el **Teorema de terminación**.

# Teorema de terminación de un ciclo

- **Teorema.** Sea  $\mathbb{V}$  el producto cartesiano de los dominios de las variables del programa y sea  $I$  un invariante del ciclo **while B do S endwhile**. Si existe una función  $fv : \mathbb{V} \rightarrow \mathbb{Z}$  tal que

1.  $\{I \wedge B \wedge v_0 = fv\} \text{ S } \{fv < v_0\},$
2.  $I \wedge fv \leq 0 \Rightarrow \neg B,$

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

- La función  $fv$  se llama **función variante** del ciclo.
- El Teorema de terminación nos permite demostrar que si un ciclo termina (i.e. no se cuelga).

## Ejemplo: Suma de índices

- Sea la siguiente tripla de Hoare:

$$\{n \geq 0 \wedge i = 1 \wedge s = 0\}$$

while (i <= n) do

    s = s + i;

    i = i + 1;

endwhile

$$\{s = \sum_{k=1}^n k\}$$

- Ya probamos que el siguiente predicado es un invariante de este ciclo.

$$I \equiv 1 \leq i \leq n + 1 \wedge s = \sum_{k=1}^{i-1} k$$

- ¿Cuál sería una buena función variante para este ciclo?

- Ejecutemos el ciclo con  $n = 6$ .

Iteración	i	s	n	$n+1-i$
0	1	0	6	6
1	2	1	6	5
2	3	3	6	4
3	4	6	6	3
4	5	10	6	2
5	6	15	6	1
6	7	21	6	0

- Una función variante representa una **cantidad que se va reduciendo** a lo largo de las iteraciones. En este caso es la cantidad de índices que falta sumar.
- Proponemos entonces  $fv = n + 1 - i$



## Ejemplo: Suma de índices

► Veamos que se cumplen las dos condiciones del teorema.

1. Para verificar que  $\{I \wedge B \wedge fv = v_0\} S \{fv < v_0\}$  para todo  $v_0$ , calculamos  $wp(S, fv < v_0)$ .

$$\begin{aligned} & wp(s:=s+1; i:=i+1, fv < v_0) \\ & \equiv wp(s:=s+1; i:=i+1, (n+1-i) < v_0) \\ & \equiv wp(s:=s+1, wp(i:=i+1, (n+1-i) < v_0)) \\ & \equiv wp(s:=s+1, def(i+1) \wedge_L (n+1-(i+1)) < v_0)) \\ & \equiv wp(s:=s+1, (n+1-(i+1)) < v_0)) \\ & \equiv def(s+1) \wedge_L n-i < v_0 \\ & \equiv n-i < v_0 \\ & \equiv n-i < n+1-i \\ & \equiv n-i < n-i+1 \checkmark \end{aligned}$$

## Ejemplo: Suma de índices

- Veamos que se cumplen las dos condiciones del teorema.

2. Verifiquemos que  $I \wedge fv \leq 0 \Rightarrow \neg B$

$$\begin{aligned} I \wedge fv \leq 0 &\equiv \overbrace{1 \leq i \leq n+1}^I \wedge s = \sum_{k=1}^{i-1} k \wedge \overbrace{n+1-i \leq 0}^{fv \leq 0} \\ &\Rightarrow i \leq n+1 \wedge n+1-i \leq 0 \\ &\Rightarrow i \leq n+1 \wedge n+1 \leq i \\ &\Rightarrow i = n+1 \\ &\Rightarrow \neg(i \leq n) \\ &\Rightarrow \neg B \checkmark \end{aligned}$$

## Ejemplo: Suma de índices

Recapitulando, sean

$$\blacktriangleright I \equiv 1 \leq i \leq n+1 \wedge s = \sum_{k=1}^{i-1} k$$

$$\blacktriangleright fv = n+1-i$$

En la teórica pasada ya habíamos probado que el ciclo es **parcialmente** correcto dado que:

1.  $P_C \Rightarrow I$
2.  $\{I \wedge B\} \text{ S } \{I\}$
3.  $I \wedge \neg B \Rightarrow Q_C$

Ahora acabamos de probar que el ciclo siempre termina ya que:

4.  $\{I \wedge B \wedge v_0 = fv\} \text{ S } \{fv < v_0\},$
5.  $I \wedge fv \leq 0 \Rightarrow \neg B,$

Por lo tanto, por (1)-(5) tenemos (finalmente) que ...

## Ejemplo: Suma de índices

- Que la siguiente tripla de Hoare:

$$\{P_C : n \geq 0 \wedge i = 1 \wedge s = 0\}$$

while (i <= n) do

    s = s + i;

    i = i + 1;

endwhile

$$\{Q_C : s = \sum_{k=1}^n k\}$$

es una tripla de Hoare **válida!**

- Esto significa que:
  1. Si el ciclo comienza en un estado que cumple  $P_C$
  2. ... entonces termina luego de un número finito de pasos
  3. y además en un estado que cumple  $Q_C$

## Recap #1: Teorema del invariante

► **Teorema.** Si  $\text{def}(B)$  y existe un predicado  $I$  tal que

1.  $P_C \Rightarrow I$ ,
2.  $\{I \wedge B\} S \{I\}$ ,
3.  $I \wedge \neg B \Rightarrow Q_C$ ,

... y **el ciclo termina**, entonces la siguiente tripla de Hoare es válida:

$$\{P_C\} \text{ while } B \text{ do } S \text{ endwhile } \{Q_C\}$$

## Recap #2: Teorema de terminación de un ciclo

- **Teorema.** Sea  $\mathbb{V}$  el producto cartesiano de los dominios de las variables del programa y sea  $I$  un invariante del ciclo **while B do S endwhile**. Si existe una función  $fv : \mathbb{V} \rightarrow \mathbb{Z}$  tal que

1.  $\{I \wedge B \wedge v_0 = fv\} \text{ S } \{fv < v_0\},$
2.  $I \wedge fv \leq 0 \Rightarrow \neg B,$

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

- La función  $fv$  se llama **función variante** del ciclo.

# Teorema de corrección de un ciclo

- **Teorema.** Sean un predicado  $I$  y una función  $fv : \mathbb{V} \rightarrow \mathbb{Z}$  (donde  $\mathbb{V}$  es el producto cartesiano de los dominios de las variables del programa), y supongamos que  $I \Rightarrow \text{def}(B)$ . Si

1.  $P_C \Rightarrow I$ ,
2.  $\{I \wedge B\} S \{I\}$ ,
3.  $I \wedge \neg B \Rightarrow Q_C$ ,
4.  $\{I \wedge B \wedge v_0 = fv\} S \{fv < v_0\}$ ,
5.  $I \wedge fv \leq 0 \Rightarrow \neg B$ ,

... entonces la siguiente tripla de Hoare es válida:

$$\{P_C\} \text{ while } B \text{ do } S \text{ endwhile } \{Q_C\}$$

# Teorema de corrección de un ciclo

- ▶ El **teorema de corrección de un ciclo** nos permite demostrar la validez de una tripla de Hoare cuando el programa es un ciclo.
- ▶ Por definición, si probamos que:

$$\{P_C\} \text{ while } B \text{ do } S \text{ endwhile } \{Q_C\}$$

... entonces probamos que:

$$P_C \Rightarrow wp(\text{while } B \text{ do } S \text{ endwhile}, Q_C)$$

- ▶ **¡Cuidado!** Probar lo anterior no significa haber obtenido un **predicado** que caracteriza a la **precondición más débil** del ciclo:

$$wp(\text{while } B \text{ do } S \text{ endwhile}, Q_C)$$



# Programas con ciclos

- ▶ En general, no se puede definir un **mecanismo efectivo** para obtener una fórmula cerrada que represente la precondition más débil de un ciclo.
- ▶ Entonces, ¿cómo hacemos para probar la corrección y terminación de un programa que **incluye** ciclos intercalados con otras instrucciones?

# Guía para demostrar programas con ciclos

¿Qué tenemos que hacer para probar que  $\{Pre\}S1;while\dots;S3\{Post\}$  es válida?

1.  $Pre \Rightarrow_L wp(S1, P_C)$
2.  $P_C \Rightarrow_L wp(while\dots, Q_C)$
3.  $Q_C \Rightarrow_L wp(S3, Post)$

Por monotonía, esto nos permite demostrar que  $Pre \Rightarrow_L wp(S1;while\dots;S3, Post)$  es verdadera.

## Recap: SmallLang

- ▶ Para las demostraciones de corrección, introdujimos un **lenguaje sencillo** y con menos opciones (mucho más simple que C++). Llamemos **SmallLang** a este lenguaje.
- ▶ SmallLang tiene únicamente:
  - ▶ Nada: `skip`
  - ▶ Asignación: `x := E`
  - ▶ Secuencia: `S1;S2`
  - ▶ Condicional: `if B then S1 else S2 endif`
  - ▶ Ciclo: `while B do S endwhile`
- ▶ No posee memoria dinámica (punteros), aliasing, llamados a función, estructura `for`, etc.

## C++ → SmallLang

Pero dado un programa en C++ podemos traducirlo a SmallLang preservando su semántica (comportamiento).

Por ejemplo:

Versión C++

```
for (int i =0; i < s.size(); i++) {  
    if (s[i]==0) {  
        s[i]++;  
    }  
}
```

Versión SmallLang

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

Ambos programas tienen el mismo comportamiento.

# Bibliografía

- ▶ David Gries - The Science of Programming
  - ▶ Part II - The Semantics of a Small Language
    - ▶ Chapter 11 - The Iterative Command