Precondición más débil de ciclos

Algoritmos y Estructuras de Datos

Recap: Teorema del Invariante

- **Teorema.** Si 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:

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

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

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

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

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

$$I \wedge B \Rightarrow wp(\texttt{S}, I)$$

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

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

$$
\begin{aligned}
& \equiv \quad 0 \le i \le n \wedge s+i = \sum_{k=1}^{i} k \\
& \equiv \quad 0 \le i \le n \wedge s = (\sum_{k=1}^{i} k) - i \\
& \equiv \quad 0 \le i \le n \wedge s = \sum_{k=1}^{i-1} k
\end{aligned}
$$

- Luego de simplificar, nos falta probar que:

$$\underbrace{(1 \le i \le n+1 \wedge s = \sum_{k=1}^{i-1} k}_{I} \wedge \underbrace{i \le n}_{B}) \Rightarrow \underbrace{(0 \le i \le n \wedge s = \sum_{k=1}^{i-1} k)}_{wp(\texttt{S}, I)}$$

- 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:

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

```
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 $fv : \mathbb{V} \to \mathbb{Z}$ tal que

  1. $\{I \wedge B \wedge v_0 = fv\}$ **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 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.

  $$I \equiv 1 \leq i \leq n + 1 \wedge s = \sum_{k=1}^{i-1} k$$
- ¿Cúal 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 $wp(\mathtt{S}, fv < v_0)$.

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

Ejemplo: Suma de índices

- Veamos que se cumplen las dos condiciones del teorema.

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

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

Ejemplo: Suma de índices

Recapitulando, sean

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

Ya habíamos probado que el ciclo es **parcialmente** correcto dado que:

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

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

4. $\{I \wedge B \wedge v_0 = fv\}$ **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

Otro ejemplo: Chequeo de paridad

Sea una secuencia de booleans s, contar la cantidad de posiciones de la secuencia iguales a `true`.

Algunas propiedades de $\#apariciones$:

- $\#apariciones(\langle\rangle, true) = 0$
- $\#apariciones(concat(s, \langle e\rangle), true) =$
  $\#apariciones(s, true) + (\text{if } e = true \text{ then } 1 \text{ else } 0 \text{ fi})$

Otro ejemplo: Chequeo de paridad

- ¿Es válida la siguiente tripla de Hoare?
- $\{i = 0 \land c = 0\}$

```
while( i < |s| ) do
   if s[i]=true then
     c := c + 1
   else
     skip
   endif;
   i := i +1
endwhile
```

$\{c = \#apariciones(s, true)\}$

Otro ejemplo: Chequeo de Paridad

- Para probar que se cumplen las condiciones del Teorema del Invariante tenemos que demostrar formalmente que se cumple:
  1. $P_C \Rightarrow I$
  2. $\{I \land B\}$ S $\{I\}$
  3. $I \land \neg B \Rightarrow Q_C$
- ¿Cuál es el predicado P_C, Q_C, I y B?
  - $P_C \equiv i = 0 \land c = 0$
  - $Q_C \equiv c = \#apariciones(s, true)$
  - $B \equiv i < |s|$
  - $I \equiv 0 \leq i \leq |s| \land_L c = \#apariciones(subseq(s, 0, i), true)$

Otro ejemplo: Chequeo de Paridad

1. $P_C \Rightarrow I$?

$$
\begin{aligned}
P_C &\equiv i = 0 \land c = 0 \\
&\Rightarrow 0 \leq i \leq |s| \land c = 0 \\
&\Rightarrow 0 \leq i \leq |s| \land c = \#apariciones(\langle\rangle, true) \\
&\Rightarrow 0 \leq i \leq |s| \land c = \#apariciones(subseq(s, 0, 0), true) \\
&\Rightarrow 0 \leq i \leq |s| \land c = \#apariciones(subseq(s, 0, i), true) \\
&\equiv I \checkmark
\end{aligned}
$$

Otro ejemplo: Chequeo de Paridad

3. $I \land \neg B \Rightarrow Q_C$?

$$
\begin{aligned}
I \land \neg B &\equiv 0 \leq i \leq |s| \land c = \#apariciones(subseq(s, 0, i), true) \\
&\quad \land \neg(i < |s|) \\
&\Rightarrow i = |s| \land c = \#apariciones(subseq(s, 0, i), true) \\
&\Rightarrow c = \#apariciones(subseq(s, 0, |s|), true) \\
&\Rightarrow c = \#apariciones(s, true) \\
&\equiv Q_C \checkmark
\end{aligned}
$$

Otro ejemplo: Chequeo de Paridad

2. Finalmente, tenemos que demostrar que $\{I \land B\}$ S $\{I\}$, para lo cual debemos probar que:

$$I \land B \Rightarrow_L wp(\mathtt{S}, I).$$

- Calculamos:

$$\begin{aligned} & wp(\mathtt{if...endif;i:=i+1}, I) \\ \equiv\ & wp(\mathtt{if...endif}, wp(\mathtt{i:=i+1}, I)) \\ \equiv\ & wp(\mathtt{if...endif}, I^i_{i+1}) \\ \equiv\ & (s[i] = true \land wp(\mathtt{c:=c+1}, I^i_{i+1})) \lor (s[i] = false \land wp(\mathtt{skip}, I^i_{i+1})) \\ \equiv\ & (s[i] = true \land (I^i_{i+1})^c_{c+1})) \lor (s[i] = false \land I^i_{i+1}) \end{aligned}$$

- Para probar que esto es verdadero, separemos en 2 casos: $s[i] = true$ y $s[i] = false$

Otro ejemplo: Chequeo de Paridad

- Si $s[i] = true$, entonces podemos simplificar el predicado:

$$\begin{aligned} & (s[i] = true \land (I^i_{i+1})^c_{c+1}) \lor (s[i] = false \land I^i_{i+1}) \\ \equiv\ & (s[i] = true \land (I^i_{i+1})^c_{c+1}) \\ \equiv\ & (I^i_{i+1})^c_{c+1} \\ \equiv\ & 0 \leq i+1 \leq |s| \land_L c+1 = \#apariciones(subseq(s, 0, i+1), true) \end{aligned}$$

- Ahora probemos que este predicado es verdadero:
  - Por hipótesis, $0 \leq i \leq |s|$ y $i < |s|$
  - Por lo tanto,

$$\begin{aligned} & 0 \leq i < |s| \\ \Rightarrow\ & 0 \leq i+1 \leq |s| \qquad (1) \end{aligned}$$

Otro ejemplo: Chequeo de Paridad

Por hipótesis:

$$c = \#apariciones(subseq(s, 0, i), true)$$

Por lo tanto,

$$\begin{aligned} c+1 &= \#apariciones(subseq(s, 0, i), true) + 1 \\ &= \#apariciones(subseq(s, 0, i), true) + (\text{if } true = true \text{ then } 1 \text{ else } 0 \text{ fi}) \\ &= \#apariciones(subseq(s, 0, i), true) + (\text{if } s[i] = true \text{ then } 1 \text{ else } 0 \text{ fi}) \\ &= \#apariciones(subseq(s, 0, i+1), true) \qquad (2) \end{aligned}$$

Finalmente, por (1) y (2) demostramos que $I \land B \Rightarrow wp(\mathtt{S}, I)$ para el caso que $s[i] = true$ (pero aún falta probarlo para $s[i] = false$)

Otro ejemplo: Chequeo de Paridad

- Si $s[i] = false$, entonces podemos nuevamente simplificar el predicado:

$$\begin{aligned} & (s[i] = true \land (I^i_{i+1})^c_{c+1}) \lor (s[i] = false \land I^i_{i+1}) \\ \equiv\ & (s[i] = false \land I^i_{i+1}) \\ \equiv\ & I^i_{i+1} \\ \equiv\ & 0 \leq i+1 \leq |s| \land_L c = \#apariciones(subseq(s, 0, i+1), true) \end{aligned}$$

- Ahora probemos que este predicado es verdadero:
  - Por hipótesis, $0 \leq i \leq |s|$ y $i < |s|$
  - Análogo al caso $s[i] = true$, podemos probar que:

$$0 \leq i+1 \leq |s| \qquad (3)$$

Otro ejemplo: Chequeo de Paridad

Por hipótesis:

$$c = \#apariciones(subseq(s, 0, i), true)$$

Por lo tanto,

$$\begin{aligned} c &= \#apariciones(subseq(s, 0, i), true) + 0 \\ &= \#apariciones(subseq(s, 0, i), true) + (\text{ if } false = true \text{ then } 1 \text{ else } 0 \text{ fi}) \\ &= \#apariciones(subseq(s, 0, i), true) + (\text{ if } s[i] = true \text{ then } 1 \text{ else } 0 \text{ fi}) \\ &= \#apariciones(subseq(s, 0, i+1), true) \qquad (4) \end{aligned}$$

Finalmente, por (3) y (4) demostramos que $I \wedge B \Rightarrow wp(\mathtt{S}, I)$ para el caso que $s[i] = false$. Y como ya probamos lo mismo para $s[i] = true$, podemos concluir que:

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

Otro ejemplo: Chequeo de Paridad

- Ya que probamos
  - $P_C \Rightarrow I$
  - $\{I \wedge B\}S\{I\}$
  - $I \wedge \neg B \Rightarrow Q_C$
- usando el teorema del invariante pudimos probar que (si el ciclo termina), se cumple Q_C.
- Ya probamos que $I \equiv 0 \leq i \leq |s| \wedge_L c = \#apariciones(s, true)$ es un invariante del ciclo.
- ¡Pero no probamos todavía que la ejecución del ciclo termina!

Otro ejemplo: Chequeo de Paridad

- La función variante representa una cantidad que se va reduciendo.
- Pero... ¿Cuál la condición para que se detenga el ciclo?
  - $B \equiv i < |s|$
  - Necesitamos que $fv \leq 0$ implique $\neg(i < |s|)$
- Por lo que proponemos entonces:

$$fv = |s| - i$$

Otro ejemplo: Chequeo de Paridad

- Sea la siguiente función candidato a función variante:

$$fv = |s| - i$$

- Veamos como evoluciona con los valores para $|s| = 4$

Iteración	$\|s\|$	i	$fv = \|s\| - i$
0	4	0	4-0=4
1	4	1	4-1=3
2	4	2	4-2=2
3	4	3	4-3=1
4	4	4	4-4=0

Otro ejemplo: Chequeo de Paridad

- Con esta definición de fv, veamos si se cumplen las dos condiciones del Teorema de Terminación:
  1. $\{I \wedge B \wedge fv = v_0\}$ S $\{fv < v_0\}$
  2. $I \wedge fv \leq 0 \Rightarrow \neg B$

Otro ejemplo: Chequeo de Paridad

- $\{I \wedge B \wedge fv = v_0\}$ S $\{fv < v_0\}$?

  Para demostrarlo tenemos que probar que:

$$I \wedge B \wedge fv = v_0 \Rightarrow wp(\mathtt{S}, fv < v_0)$$

Otro ejemplo: Chequeo de Paridad

- Comenzamos con la definición de la wp:

$$\begin{aligned}
& wp(\mathtt{if\ldots endif;i:=i+1}, |s| - i < v_0) \\
\equiv\ & wp(\mathtt{if\ldots endif}, wp(\mathtt{i:=i+1}, |s| - i < v_0)) \\
\equiv\ & wp(\mathtt{if\ldots endif}, (|s| - i < v_0)^{i}_{i+1}) \\
\equiv\ & wp(\mathtt{if\ldots endif}, |s| - (i+1) < v_0) \\
\equiv\ & (s[i] = true \wedge wp(\mathtt{c:=c+1}, |s| - (i+1) < v_0)) \\
& \vee (s[i] = false \wedge wp(\mathtt{skip}, |s| - (i+1) < v_0)) \\
\equiv\ & (s[i] = true \wedge (|s| - (i+1) < v_0)^{c}_{c+1}) \\
& \vee (s[i] = false \wedge |s| - (i+1) < v_0) \\
\equiv\ & (s[i] = true \wedge |s| - (i+1) < v_0) \\
& \vee (s[i] = false \wedge |s| - (i+1) < v_0) \\
\equiv\ & (s[i] = true \vee s[i] = false) \wedge |s| - (i+1) < v_0 \\
\equiv\ & |s| - (i+1) < v_0
\end{aligned}$$

Otro ejemplo: Chequeo de Paridad

- ... como $fv = v_0$ equivale a $|s| - i$, reemplazamos v_0 con esa expresión

$$\begin{aligned}
\equiv\ & |s| - (i+1) < |s| - i \\
\equiv\ & -(i+1) < -i \\
\equiv\ & (i+1) > i
\end{aligned}$$

- Lo cual es verdadero.
- Por lo tanto, demostramos que

$$I \wedge B \wedge fv = v_0 \Rightarrow wp(\mathtt{S}, fv < v_0) \checkmark$$

Otro ejemplo: Chequeo de Paridad

2. $I \land fv \leq 0 \Rightarrow \neg B$?

$$\begin{aligned} fv \leq 0 &\equiv |s| - i \leq 0 \\ &\equiv |s| \leq i \\ &\Rightarrow \neg(i < |s|) \\ &\equiv \neg B \checkmark \end{aligned}$$

- Por lo tanto, probamos que $I \land fv \leq 0 \Rightarrow \neg B$
- Ya que se cumplen sus hipótesis, por el teorema de terminación podemos concluir que el ciclo siempre termina.

Otro ejemplo: Chequeo de Paridad

- Finalmente, probamos que:
  1. $P_C \Rightarrow I$
  2. $\{I \land B\}S\{I\}$
  3. $I \land \neg B \Rightarrow Q_C$
  4. $\{I \land v_0 = fv\}S\{fv < v_0\}$
  5. $I \land fv \leq 0 \Rightarrow \neg B$
- Entonces, por (1)-(5) , se cumplen las hipótesis de ambos teoremas (teorema del invariante + teorema de terminación).
- Por lo tanto, la tripla de Hoare es válida (i.e., dada P_C, el ciclo siempre termina y vale Q_C)

Recap #1: Teorema del invariante

- **Teorema.** Si def(B) y existe un predicado I tal que
  1. $P_C \Rightarrow I$,
  2. $\{I \land B\}\ S\ \{I\}$,
  3. $I \land \neg B \Rightarrow Q_C$,

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

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

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 $fv : \mathbb{V} \to \mathbb{Z}$ tal que
  1. $\{I \land B \land v_0 = fv\}$ **S** $\{fv < v_0\}$,
  2. $I \land fv \leq 0 \Rightarrow \neg B$,

  ... 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 $fv : \mathbb{V} \to \mathbb{Z}$ (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:

  $\{P_C\}$ **while B do S 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:

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

  ... entonces probamos que:

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

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

  $wp($while B do S endwhile$, Q_C)$

Programas con ciclos

- En general, no se puede definir un mecanismo efectivo para obtener una fórmula cerrada que represente la precondición 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 $\{Pre\}\texttt{S1;while...;S3}\{Post\}$ es válida?

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

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

Recap: SmallLang

- Para las demostraciones de corrección, introdujimos un lenguaje sencillo y con menos opciones (mucho más simple que Java). 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.

Java → SmallLang

Pero dado un programa en Java podemos traducirlo a SmallLang preservando su semántica (comportamiento).
Por ejemplo:

Versión Java

```
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.

Corrección de programas en Java

Para demostrar la corrección de un programa en Java con respecto a una especificación, podemos:

1. Traducir el programa Java a SmallLang preservando su comportamiento.
2. Demostrar la corrección del programa en SmallLang con respecto a la especificación.
3. Entonces, probamos la corrección del comportamiento del programa original.

Bibliografía

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