Procedimientos,Funciones y Recursión

Diego Mosquera y Luis Astorga
Departamento de Computación y Tecnología de la Información
Universidad Simón Bolívar
Septiembre-Diciembre 2008

December 2, 2008

1 Procedimientos

1.1 Procedimientos no recursivos

1.1.1 Sintaxis

$$\textbf{proc } p \quad \left(\begin{array}{c} \text{in } x_1 : \text{ Tipo-}x_1 \text{ ; } \ldots \text{; in } x_n : \text{ Tipo-}x_n \text{ ;} \\ \text{in-out } y_1 : \text{ Tipo-}y_1 \text{ ; } \ldots \text{; in-out } y_m : \text{ Tipo-}y_m \text{ ;} \\ \text{out } z_1 : \text{ Tipo-}z_1 \text{ ; } \ldots \text{; out } z_l : \text{ Tipo-}z_l \text{ ;} \end{array} \right)$$

{Pre}
{Post}
|[
Cuerpo
]|
.

Si los tipos de los parámetros formales que comparten el mismo atributo son iguales se abrevia la notación. Para simplificar el desarrollo posterior asumimos la notación vectorial

proc $p\,(\text{in } \bar{x} : \text{ Tipo-}x \text{ ; in-out } \bar{y} : \text{ Tipo-}y \text{ ; out } \bar{z} : \text{ Tipo-}z)$

1.1.2 Verificación de la correctitud de los procedimientos

Teorema 1 *Probar la correctitud de un procedimiento equivale a verificar que se cumple la tripleta*

$$\{Pre \ \wedge \bar{y} = \bar{y}'\}\ Cuerpo\,\{Post\} \tag{1}$$

que, en términos de la pre-condición más débil se traduce en probar la tautología

$$[Pre \Rightarrow (wp_{Cuerpo}\,(Post))\,(\bar{y}' := \bar{y})] \tag{2}$$

donde $\bar{y}$ es una lista de parametros in-out y $\bar{y}'$ denota la lista de valores iniciales o de entrada de $\bar{y}$ antes de ser procesados.

Ejemplo

$$
\begin{array}{rl}
\textbf{proc}\ \mathit{swap} & (\text{in-out } y_1, y_2 : \text{ int}) \\
\text{Pre:} & \{V\} \\
\text{Post:} & \{y_1 = y_2' \wedge y_2 = y_1'\} \\
|[& \\
& y_1 := y_1 + y_2; \\
& y_2 := y_1 - y_2; \\
& y_1 := y_1 - y_2 \\
]| &
\end{array}
$$

Se busca verificar que se cumple la tripleta

$$\{V \wedge y_1 = y_1' \wedge y_2 = y_2'\}\, y_1 := y_1 + y_2; y_2 := y_1 - y_2; y_1 := y_1 - y_2\, \{y_1 = y_2' \wedge y_2 = y_1'\}$$

Calculemos primero la pre-condición más débil

$$
\begin{array}{rl}
& wp_{y_1:=y_1+y_2;y_2:=y_1-y_2;y_1:=y_1-y_2}\,(y_1 = y_2' \wedge y_2 = y_1') \\
\equiv & wp_{y_1:=y_1+y_2}\,(wp_{y_2:=y_1-y_2}\,(wp_{y_1:=y_1-y_2}\,(y_1 = y_2' \wedge y_2 = y_1'))) \\
\equiv & wp_{y_1:=y_1+y_2}\,(wp_{y_2:=y_1-y_2}\,((y_1 = y_2' \wedge y_2 = y_1')\,(y_1 := y_1 - y_2))) \\
\equiv & wp_{y_1:=y_1+y_2}\,(wp_{y_2:=y_1-y_2}\,(y_1 - y_2 = y_2' \wedge y_2 = y_1')) \\
\equiv & wp_{y_1:=y_1+y_2}\,((y_1 - y_2 = y_2' \wedge y_2 = y_1')\,(y_2 := y_1 - y_2)) \\
\equiv & wp_{y_1:=y_1+y_2}\,(y_2 = y_2' \wedge y_1 - y_2 = y_1') \\
\equiv & (y_2 = y_2' \wedge y_1 - y_2 = y_1')\,(y_1 := y_1 + y_2) \\
\equiv & y_2 = y_2' \wedge y_1 = y_1' \\
\equiv & y_1 = y_1' \wedge y_2 = y_2'
\end{array}
$$

Ahora, obsérvese que, por (1),

$$[\text{Pre } \wedge \bar{y} = \bar{y}' \Rightarrow wp_{\text{Cuerpo}}\,(\text{Post})]$$

que en este caso es

$$[V \wedge y_1 = y_1' \wedge y_2 = y_2' \Rightarrow y_1 = y_1' \wedge y_2 = y_2']$$

lo cual es trivialmente cierto. Por (2)

$$[V \Rightarrow (y_1 = y_1' \wedge y_2 = y_2')\,(y_1', y_2' := y_1, y_2)]$$

lo cual también es trivialmente cierto.

Ejercicio Dado el procedimiento *interseg* que intercambia dos segmentos no solapados de igual longitud dentro de un arreglo *arr*, donde *izq* (por izquierdo) es la posición de la primera entrada del primer segmento, *der* (por derecho) la primera entrada del segundo segmento, y *long* es la longitud de los segmentos a intercambiar, la condición $izq + long - 1 < der$ garantiza que los segmentos no se solapan. Aplicando la técnica de sustitución de la constante *long* por la variable

k en la post-condición para determinar el invariante, y tomando la función de cota $long - k$ se tiene el procedimiento con anotaciones:

$$
\begin{array}{rl}
\textbf{proc}\ \mathit{interseg} & (\text{in}\ izq, der, long:\ \text{int}\ ;\ \text{in-out}\ arr:\ \text{arreglo de int}) \\
\text{Pre:} & \{long > 0 \wedge izq < der \wedge izq + long - 1 < der\} \\
\text{Post:} & \left\{arr = arr'_{([izq..izq+long),[der..der+long);arr'[der..der+long),arr'[izq..izq+long))}\right\} \\
|[& \text{var}\ k : \text{int}; \\
 & k := 0; \\
\text{Inv:} & \left\{ \begin{array}{c} 0 < k \leq long \wedge \\ arr = arr'_{([izq..izq+k),[der..der+k);arr'[der..der+k),arr'[izq..izq+k))} \end{array} \right\} \\
\text{cota:} & \{long - k\} \\
 & \text{do}\ (k \neq long) \longrightarrow arr[k], arr[k+long], k := arr[k+long], a[k], k+1\ \text{od} \\
]| &
\end{array}
$$

Para verificar que el procedimiento es correcto debemos, según el teorema 1, establecer el cumplimiento de la secuencia

$$
\begin{array}{l}
\{\text{Pre} \wedge (arr = arr')\} \\
k := 0; \\
\{\text{Inv}\} \\
\text{do}\ (k \neq long) \longrightarrow arr[k], arr[k+long], k := arr[k+long], arr[k], k+1\ \text{od} \\
\{\text{Post}\}
\end{array}
$$

1.1.3 Regla de la correctitud de las llamadas a los procedimientos

La llamada a un procedimiento p es una instrucción $p(\bar{a};\bar{b};\bar{c})$ donde $\bar{a}$ es una lista de argumentos de entrada (in), $\bar{b}$ es una lista de argumentos de entrada-salida (in-out), y $\bar{c}$ es una lista de argumentos de salida (out).

Teorema 2 *Dada la especificación del procedimiento*

proc p	***proc*** p *(in* $\bar{x}$ *: Tipo-x ; in-out* $\bar{y}$ *: Tipo-y ; out* $\bar{z}$ *: Tipo-z)*
	$\{Pre\}$
	$\{Post\}$

y la llamada al mismo

$$\{P\}\, p(\bar{a};\bar{b};\bar{c})\, \{Q\}$$

entonces,

1) La pre-condición más débil de la llamada viene dada por el predicado

$$
\begin{array}{rcl}
wp_{p(\bar{a};\bar{b};\bar{c})}(Q) & \equiv & Pre\left(\bar{x},\bar{y} := \bar{a},\bar{b}\right) \wedge \\
 & & \left(Post\left(\bar{x},\bar{y}',\bar{y},\bar{z} := \bar{a},\bar{b},\bar{B},\bar{C}\right) \Rightarrow Q\left(\bar{b},\bar{c} := \bar{B},\bar{C}\right)\right)
\end{array}
$$

2) Se tiene la tautología

$$\left[P \Rightarrow wp_{p(\bar{a};\bar{b};\bar{c})}(Q)\right]$$

donde $\bar{B}$ *y* $\bar{C}$ *son símbolos que representan los valores de salida de las listas de parametros* $\bar{y}$ *y* $\bar{z}$ *respectivamente, después de la ejecución del procedimiento.*

Ejemplo Dado el procedimiento

proc *swap*	(in-out y_1, y_2 : int)
Pre:	$\{V\}$
Post:	$\{y_1 = y_2' \wedge y_2 = y_1'\}$

se quiere verificar la correctitud de la llamada

$$\{a > 0 \wedge b < 0\}\, swap\,(a, b)\, \{a < 0 \wedge b > 0\}$$

.

Primero,

$$\begin{array}{ll}
 & wp_{swap(a,b)}\,(a < 0 \wedge b > 0) \\
\equiv & \text{por teorema 2 (1)} \\
 & V\,(y_1, y_2 := a, b) \wedge \\
 & \left(\begin{array}{l} (y_1 = y_2' \wedge y_2 = y_1')\,(y_1', y_2', y_1, y_2 := a, b, A, B) \\ \qquad \Rightarrow (a < 0 \wedge b > 0)\,(a, b := A, B) \end{array} \right) \\
\equiv & \text{por sustitución, } [V\,(x := E) \equiv V] \text{ e identidad de } \wedge \\
 & (A = b \wedge B = a) \Rightarrow (A < 0 \wedge B > 0)
\end{array}$$

Luego, por teorema 2 (2), se tiene que verificar la tautología

$$[(a > 0 \wedge b < 0) \Rightarrow ((A = b \wedge B = a) \Rightarrow (A < 0 \wedge B > 0))]$$

En efecto,

$$\begin{array}{ll}
 & (a > 0 \wedge b < 0) \Rightarrow ((A = b \wedge B = a) \Rightarrow (A < 0 \wedge B > 0)) \\
\equiv & \text{porque } [(p \Rightarrow (q \Rightarrow r)) \equiv (p \wedge q \Rightarrow r)] \\
 & (a > 0 \wedge b < 0 \wedge A = b \wedge B = a) \Rightarrow (A < 0 \wedge B > 0) \\
\equiv & \text{por simplificación} \\
 & (a > 0 \wedge b < 0 \wedge A = b \wedge B = a) \Rightarrow (b < 0 \wedge a > 0) \\
\equiv & \text{porque } [p \wedge q \Rightarrow p] \\
 & V
\end{array}$$

Ejercicio Dado el procedimiento

proc *interseg*	(in $izq, der, long$: int ; in-out arr : arreglo de int)
Pre:	$\{long > 0 \wedge izq < der \wedge izq + long - 1 < der\}$
Post:	$\left\{arr = arr'_{([izq..izq+long),[der..der+long);arr'[der..der+long),arr'[izq..izq+long))}\right\}$

Se construye otro procedimiento que hace la rotación de dos segmentos contiguos (no necesariamente de la misma longitud) en un arreglo b, usando el procedimiento anterior, donde: a es la posición inicial de los segmentos, z la posición inicial del complemento y m el pivote de rotación

$$
\begin{array}{rl}
\textbf{proc } \mathit{rotaseg} & (\text{in } a, m, z: \text{ int };\ \text{in-out } b: \text{ arreglo de int}) \\
\text{Pre:} & \{a < k < z\} \\
\text{Post:} & \left\{ b = b'_{([a..a+z-k),[a+z-k..z);b'[k..z),b'[a..k))} \right\} \\
|[& \text{var } i, d : \text{int}; \\
 & i, d := m - a, z - m \\
\text{Inv:} & \left\{ \begin{array}{c} 0 < i \le m - a \wedge 0 < d \le z - m \wedge \\ \left(\begin{array}{c} b = b'_{([m-i..m),[m-i+d..m+d);b'[m-i+d..m+d),b'[m-i..m))} \vee \\ b = b'_{([m-i..m-i+d),[m..m+d);b'[m..m+d),b'[m-i..m-i+d))} \end{array} \right) \end{array} \right\} \\
\text{cota:} & \{\max(i, d)\} \\
 & \quad \text{do } (i < d) \longrightarrow \quad \mathit{interseg}\,(m - i, m + d - i, i; b)\,; \\
 & \qquad\qquad\qquad\qquad d := d - i \\
 & \quad\ \ [] \ (i > d) \longrightarrow \quad \mathit{interseg}\,(m - i, m, d; b)\,; \\
 & \qquad\qquad\qquad\qquad i := i - d \\
 & \quad \text{od}; \\
 & \mathit{interseg}\,(m - i, m, i; b) \\
]| &
\end{array}
$$

Para verificar la correctitud de *rotaseg*, por teorema 1, debemos probar que se cumple la tripleta

$$
\begin{array}{l}
\{\text{Pre} \wedge (\forall j : a \le j < z : b[j] = b'[j])\} \\
i, d := m - a, z - m; \\
\{\text{Inv}\} \\
\quad \text{do } (i < d) \longrightarrow \quad \mathit{interseg}\,(m - i, m + d - i, i; b)\,; \\
\qquad\qquad\qquad\qquad d := d - i \\
\quad\ \ [] \ (i > d) \longrightarrow \quad \mathit{interseg}\,(m - i, m, d; b)\,; \\
\qquad\qquad\qquad\qquad i := i - d \\
\quad \text{od}; \\
\{\text{Inv} \wedge i = d\} \\
\mathit{interseg}\,(m - i, m, i; b) \\
\{\text{Post}\}
\end{array}
$$

Cuando se verifique la correctitud del ciclo, deberán probarse las llamadas al procedimiento *interseg*:

1) Que se cumple la tripleta

$$\{\text{Inv} \wedge i < d\}\, \mathit{interseg}\,(m - i, m + d - i, i; b)\, \{\text{Inv}\,(d := d - i)\}$$

2) Que se cumple la tripleta

$$\{\text{Inv} \wedge i > d\}\, \mathit{interseg}\,(m - i, m, d; b)\,;\ \{\text{Inv}\,(i := i - d)\}$$

3) Que se cumple la tripleta

$$\{\text{Inv} \wedge i = d\}\, \mathit{interseg}\,(m - i, m, i; b)\, \{\text{Post}\}$$

1.2 Procedimientos recursivos

1.2.1 Sintaxis

Dada la especificación e implementación recursiva del procedimiento

$$
\begin{array}{ll}
\textbf{proc}\ p & \textbf{proc}\ p\,(\text{in } \bar{x}:\ \text{Tipo-}x\ ;\ \text{in-out } \bar{y}:\ \text{Tipo-}y\ ;\ \text{out } \bar{z}:\ \text{Tipo-}z) \\
 & \{\text{Pre}\} \\
 & \{\text{Post}\} \\
 & \{\text{cota}\} \\
|[& \\
 & ...\ \{P\}\, p(f(\bar{x}); g(\bar{y}); h(\bar{z}))\, \{Q\}\ ... \\
]| &
\end{array}
$$

verificar la correctitud del procedimiento p requiere primero verificar la correctitud de las llamadas recursivas del tipo

$$\{P\}\, p(f(\bar{x}); g(\bar{y}); h(\bar{z}))\, \{Q\}$$

donde $f(\bar{x}), g(\bar{y}), h(\bar{z})$ son modificaciones de los parametros formales $\bar{x}, \bar{y}, \bar{z}$.

1.2.2 Regla de la correctitud de las llamadas recursivas

Teorema 3 *La tripleta de la llamada recursiva en un procedimiento se cumple si y sólo si:*

1) La pre-condición más débil de la llamada viene dada por el predicado

$$
\begin{array}{rcl}
wp_{p(f(\bar{x});g(\bar{y});h(\bar{z}))}\,(Q) & \equiv & \mathit{Pre}\,(\bar{x},\bar{y} := f(\bar{x}); g(\bar{y})) \wedge \\
0 & \leq & (\mathit{cota})\,(\bar{x},\bar{y} := f(\bar{x}); g(\bar{y})) < \mathit{cota} \wedge \\
 & & \left(\mathit{Post}\left(\bar{x},\bar{y}',\bar{y},\bar{z} := f(\bar{x}), g(\bar{y}), \bar{B}, \bar{C}\right) \Rightarrow Q\left(g(\bar{y}), h(\bar{z}) := \bar{B}, \bar{C}\right)\right)
\end{array}
$$

2) Se tiene la tautología

$$\left[P \Rightarrow wp_{p(f(\bar{x});g(\bar{y});h(\bar{z}))}\,(Q)\right]$$

donde $\bar{B}$ y $\bar{C}$ son símbolos que representan los valores de salida de las listas de parametros $\bar{y}$ y $\bar{z}$ respectivamente, después de la ejecución del procedimiento.

Ejemplo Dada la función recursiva de Fibonacci

$$
\begin{array}{rl}
F: & \mathbb{N} \longrightarrow \mathbb{N} \\
F\,(n) = & \begin{cases} 0 & ;\text{si } n = 0 \\ 1 & ;\text{si } n = 1 \\ F\,(n-1) + F\,(n-2) & ;\text{si } n \geq 2 \end{cases}
\end{array}
$$

se construye un procedimiento recursivo que implementa la función F

$$
\begin{array}{rll}
\textbf{proc } fib & (\text{in } n : \text{int} \ ; \text{out } r : \text{int}) & \\
\text{Pre:} & \{n \geq 0\} & \\
\text{Post:} & \{r = F(n)\} & \\
\text{cota} & \{n\} & \\
|[& \text{var } r_1, r_2 : \text{int} \ ; & \\
 & r_1, r_2 := 0, 0 \ ; & \\
 & \quad \text{if } (n = 0) \longrightarrow & \{n = 0\} \\
 & & r := 0 \\
 & & \text{Post} \\
 & \quad [] \ (n = 1) \longrightarrow & \{n = 1\} \\
 & & r := 1 \\
 & & \text{Post} \\
 & \quad [] \ (n \geq 2) \longrightarrow & \{n \geq 2\} \\
 & & fib\,(n-1, r_1)\,; \\
 & & \{n \geq 2 \wedge r_1 = F(n-1)\} \\
 & & fib\,(n-2, r_2)\,; \\
 & & \{n \geq 2 \wedge r_1 = F(n-1) \wedge r_2 = F(n-2)\} \\
 & & r := r_1 + r_2 \\
 & & \text{Post} \\
 & \quad \text{fi} & \\
]| & &
\end{array}
$$

Para verificar la correctitud del procedimiento recursivo fib debe probarse previamente que:

1) Se cumple la tripleta

$$\{n \geq 2 \wedge r_1 = F(n-1) \wedge r_2 = F(n-2)\}\, r := r_1 + r_2\, \{r = F(n)\}$$

.

$$
\begin{array}{ll}
 & (r = F(n))\,(r := r_1 + r_2) \\
\equiv & \text{por sustitución} \\
 & r_1 + r_2 = F(n) \\
\Leftarrow & \text{por definición de } F \\
 & r_1 + r_2 = F(n-1) + F(n-2) \wedge n \geq 2 \\
\Leftarrow & \text{por álgebra} \\
 & r_1 = F(n-1) \wedge r_2 = F(n-2) \wedge n \geq 2
\end{array}
$$

.

2) Se cumple la tripleta

$$\{n \geq 2 \wedge r_1 = F(n-1)\}\, fib\,(n-2, r_2)\, \{n \geq 2 \wedge r_1 = F(n-1) \wedge r_2 = F(n-2)\}$$

Por teorema 3,

$wp_{fib(n-2,r_2)}\,(n \geq 2 \wedge r_1 = F\,(n-1) \wedge r_2 = F\,(n-2))$

$\equiv$ teorema

$(n \geq 0)\,(n := n-2) \wedge 0 \leq (n)\,(n := n-2) < n$

$((r = F(n))\,(n, r := n-2, R) \Rightarrow (n \geq 2 \wedge r_1 = F\,(n-1) \wedge r_2 = F\,(n-2))\,(r_2 := R))$

$\equiv$ por sustitución

$n \geq 2 \wedge n \geq 2 \wedge V \wedge (R = F(n-2) \Rightarrow n \geq 2 \wedge r_1 = F\,(n-1) \wedge R = F\,(n-2))$

$\Leftarrow$ porque $[p \Rightarrow (q \Rightarrow p \wedge q)]$, idempotencia e identidad de $\wedge$

$n \geq 2 \wedge r_1 = F\,(n-1)$

.

3) Se cumple la tripleta

$$\{n \geq 2\}\,fib\,(n-1, r_1)\,\{n \geq 2 \wedge r_1 = F\,(n-1)\}$$

Por teorema 3,

$wp_{fib(n-1,r_1)}\,(n \geq 2 \wedge r_1 = F\,(n-1))$

$\equiv$ teorema

$(n \geq 0)\,(n := n-1) \wedge 0 \leq (n)\,(n := n-1) < n$

$((r = F(n))\,(n, r := n-1, R) \Rightarrow (n \geq 2 \wedge r_1 = F\,(n-1))\,(r_1 := R))$

$\equiv$ por sustitución,

$n \geq 1 \wedge n \geq 1 \wedge V \wedge (R = F(n-1) \Rightarrow n \geq 2 \wedge R = F\,(n-1))$

$\Leftarrow$ porque $[p \Rightarrow (q \Rightarrow p \wedge q)]$, idempotencia e identidad de $\wedge$

$n \geq 1 \wedge n \geq 2$

$\equiv$ por álgebra

$n \geq 2$

.

4) Se cumple la tripleta

$$\{n = 1\}\,r := 1\,\{r = F(n)\}$$

.

$(r = F(n))\,(r := 1)$

$\equiv$ por sustitución

$1 = F(n)$

$\Leftarrow$ por definición de F

$n = 1$

.

5) Se cumple la tripleta

$$\{n = 0\}\,r := 0\,\{r = F(n)\}$$

.

$(r = F(n))\,(r := 0)$

$\equiv$ por sustitución

$0 = F(n)$

$\Leftarrow$ por definición de F

$n = 0$

.

Ejercicio Dado el procedimiento

proc *interseg*	(in $izq, der, long:$ int ; in-out $arr:$ arreglo de int)
Pre:	$\{long > 0 \wedge izq < der \wedge izq + long - 1 < der\}$
Post:	$\left\{arr = arr'_{([izq..izq+long),[der..der+long);arr'[der..der+long),arr'[izq..izq+long))}\right\}$

Se construye un procedimiento recursivo que hace de nuevo la rotación de dos segmentos contiguos (no necesariamente de la misma longitud) en un arreglo b, usando el procedimiento anterior, donde: a es la posición inicial de los segmentos, z la posición inicial del complemento y m el pivote de rotación

proc *rotaseg* (in $a, m, z:$ int ; in-out $b:$ arreglo de int)
Pre: $\{a < m < z\}$
Post: $\left\{b = b'_{([a..a+z-m),[a+z-m..z);b'[m..z),b'[a..m))}\right\}$
cota: $\{\max(m-a, z-m)\}$

$$
\begin{array}{l}
|[\\
\quad \text{if} \quad z - m = m - a \longrightarrow \quad \{z - m = m - a\} \\
\qquad\qquad\qquad\qquad\qquad\quad interseg\,(a, m, z - m; b) \\
\qquad\qquad\qquad\qquad\qquad\quad \{\text{Post}\} \\
\qquad\quad z - m < m - a \longrightarrow \quad \{z - m < m - a\} \\
\qquad\qquad\qquad\qquad\qquad\quad interseg\,(a, m, z - m; b)\,; \\
\qquad\qquad\qquad\qquad\qquad\quad wp_{rotaseg(a+z-m,m,z;b)}\,(\text{Post}) \\
\qquad\qquad\qquad\qquad\qquad\quad rotaseg\,(a + z - m, m, z; b) \\
\qquad\qquad\qquad\qquad\qquad\quad \{\text{Post}\} \\
\qquad\quad z - m > m - a \longrightarrow \quad \{z - m > m - a\} \\
\qquad\qquad\qquad\qquad\qquad\quad interseg\,(a, a + z - m, m - a; b)\,; \\
\qquad\qquad\qquad\qquad\qquad\quad wp_{rotaseg(a,m,a+z-m;b)}\,(\text{Post}) \\
\qquad\qquad\qquad\qquad\qquad\quad rotaseg\,(a, m, a + z - m; b) \\
\qquad\qquad\qquad\qquad\qquad\quad \{\text{Post}\} \\
\quad \text{fi} \\
]|
\end{array}
$$

Antes de que se verifique la correctitud del procedimiento, deberán probarse las llamadas recursivas y las llamadas al procedimiento *interseg*:

1) Que se cumple la tripleta

$$\{z - m = m - a\}\, interseg\,(a, m, z - m; b)\,\{\text{Post}\}$$

2) Que se cumple la secuencia

$$\{z - m < m - a\}\, interseg\,(a, m, z - m; b)\,; rotaseg\,(a + z - m, m, z; b)\;\;\{\text{Post}\}$$

3) Que se cumple la secuencia

$$\{z - m > m - a\}\, interseg\,(a, a + z - m, m - a; b)\,; rotaseg\,(a, m, a + z - m; b)\,\{\text{Post}\}$$

2 Funciones

2.1 Funciones no recursivas

2.1.1 Sintaxis

fun f $(x_1 : \text{Tipo-1}; \ldots\ x_n : \text{Tipo-}n) \longrightarrow \text{Tipo-}f$
{Pre}
{Post}
|[
Cuerpo
$>> E$
]|

2.1.2 Correctitud de las funciones

Cuando la función tiene cuerpo, debe cumplirse

$$\{\text{Pre}\}\,\text{Cuerpo}\,\{\text{Post}\,(f := E)\}$$

Cuando la función no tiene cuerpo

$$[\text{Pre}\ \Rightarrow\ \text{Post}\,(f := E)]$$

2.1.3 Correctitud de las llamadas a funciones

Los siguientes casos no consideran llamadas anidadas:

Primer caso: Una o varias llamadas similares

Teorema 4 *Se quiere verificar la correctitud de la llamada*

$$\{P\}\,Ins\,\{Q\}$$

donde Ins es una instrucción que contiene una llamada a función $f\,(\bar{a})$ o varias llamadas iguales (con los mismos argumentos)

1)

$$wp_{Ins}\,(Q) \equiv\ Pre\,(\bar{x} := \bar{a}) \wedge \big(Post\,(\bar{x}, f := \bar{a}, E) \Rightarrow wp_{\underline{Ins}}Q\big)$$

donde $\underline{Ins}$ es la instrucción Ins sustituyendo cada ocurrencia de $f\,(\bar{a})$ por el símbolo E que representa el valor de la función f ya evaluada en $\bar{a}$.

2) Se verifica que

$$[P \Rightarrow wp_{Ins}\,(Q)]$$

Ejemplo: Sea la especificación de función no recursiva

$$\boxed{\begin{array}{rl} \textbf{fun } f & (x : \text{int}) \longrightarrow \text{real} \\ \text{Pre:} & \{x \geq 0\} \\ \text{Post:} & \{f = \sqrt{x}\} \end{array}}$$

Se quiere verificar la correctitud de la instrucción de asignación con dos llamadas iguales a una misma función

$$\{n = 5\}\, r_0, r_1 := (1 + f(n))\,/2, (1 - f(n))\,/2 \left\{ r_0 = \frac{1+\sqrt{5}}{2} \wedge r_1 = \frac{1-\sqrt{5}}{2} \right\}$$

Paso 1: Primero consideramos la precondición más débil para la instrucción auxiliar

$$r_0, r_1 := (1 + E)\,/2, (1 - E)\,/2$$

donde el símbolo E representa el valor de la función f evaluada en el argumento n :

.

$wp_{r_0,r_1:=(1+E)/2,(1-E)/2}\left(r_0 = \frac{1+\sqrt{5}}{2} \wedge r_1 = \frac{1-\sqrt{5}}{2}\right)$

$\equiv$ por definición de asignación

$\left(r_0 = \frac{1+\sqrt{5}}{2} \wedge r_1 = \frac{1-\sqrt{5}}{2}\right)(r_0, r_1 := (1 + E)\,/2, (1 - E)\,/2)$

$\equiv$ por sustitución

$\frac{1+E}{2} = \frac{1+\sqrt{5}}{2} \wedge \frac{1-E}{2} = \frac{1-\sqrt{5}}{2}$

$\equiv$ por aritmética

$E = \sqrt{5}$

.

Ahora se busca la precondición más débil para la instrucción de asignación en cuestión

.

$wp_{r_0,r_1:=(1+f(n))/2,(1-f(n))/2}\left(r_0 = \frac{1+\sqrt{5}}{2} \wedge r_1 = \frac{1-\sqrt{5}}{2}\right)$

$\equiv$ por teorema 4

$(x \geq 0)\,(x := n) \wedge \left((f = \sqrt{x})\,(x, f := n, E) \Rightarrow E = \sqrt{5}\right)$

$\equiv$ por sustitución

$n \geq 0 \wedge \left((E = \sqrt{n}) \Rightarrow E = \sqrt{5}\right)$

Paso 2: Se verifica que

$$\left[(n = 5) \Rightarrow \left(n \geq 0 \wedge \left((E = \sqrt{n}) \Rightarrow E = \sqrt{5}\right)\right)\right]$$

Se asume el antecedente cierto, entonces

$n \geq 0 \wedge \left((E = \sqrt{n}) \Rightarrow E = \sqrt{5}\right)$

$\equiv$ porque $n = 5$

$V \wedge V$

$\equiv$ por idempotencia

V

.

Segundo caso: Distintas llamadas de la misma función

Teorema 5 *Se quiere verificar la correctitud de la llamada*

$$\{P\}\, Ins\, \{Q\}$$

donde Ins es una instrucción que contiene n distintas llamada a una misma función

$$f\left(\bar{a}_0\right), \ldots, f\left(\bar{a}_{n-1}\right)$$

con $\bar{a}_i \neq \bar{a}_j$ *para todos* $0 \leq i < j < n$.

1)

$$\begin{array}{lll} wp_{Ins}\left(Q\right) & \equiv & \left(\forall i : 0 \leq i < n : \ Pre\left(\bar{x} := \bar{a}_i\right)\right) \wedge \\ & & \big(\left(\forall i : 0 \leq i < n : Post\left(\bar{x}, f := \bar{a}_i, E_i\right)\right) \Rightarrow wp_{\underline{Ins}}Q\big) \end{array}$$

donde $\underline{Ins}$ *es la instrucción Ins sustituyendo cada ocurrencia de* $f\left(\bar{a}_i\right)$ *por el símbolo* E_i *que representa el valor de la función* f *ya evaluada en* $\bar{a}_i$.

2) Se verifica que

$$\left[P \Rightarrow wp_{Ins}\left(Q\right)\right]$$

Ejemplo: Sea la especificación de función

fun f	$(x : \text{int}) \longrightarrow \text{real}$
Pre:	$\{x \geq 0\}$
Post:	$\{f = \log_2 x\}$

Se quiere verificar la correctitud de la instrucción de asignación con dos llamadas distintas a una misma función

$$\{n \neq m \wedge n = 2 \wedge m = 1\}\, z := \left(f(n) - f(m)\right) / \left(n - m\right) \{z = 1\}$$

Paso 1: Primero consideramos la precondición más débil para la instrucción auxiliar $z := (E - F) / (n - m)$ donde los símbolos E y F representan los valores de la función f evaluada en los argumentos distintos n y m :

.

$$\begin{array}{ll} & wp_{z:=(E-F)/(n-m)}\left(z = 1\right) \\ \equiv & \text{por definición de asignación} \\ & \left(z = 1\right)\left(z := \left(E - F\right) / \left(n - m\right)\right) \\ \equiv & \text{por sustitución} \\ & \frac{E-F}{n-m} = 1 \end{array}$$

.

Ahora se busca la precondición más débil para la instrucción de asignación en cuestión

.

$$
\begin{array}{cl}
 & wp_{z:=(f(n)-f(m))/(n-m)}\,(z=1) \\
\equiv & \text{por teorema} \\
 & \left(\begin{array}{l} \qquad\qquad (x \geq 0)\,(x := n) \wedge (x \geq 0)\,(x := m) \wedge \\ (f = \log_2 x)\,(x, f := n, E) \wedge (f = \log_2 x)\,(x, f := m, F) \\ \Rightarrow \\ (E - F)\,/\,(n - m) = 1 \end{array}\right) \\
\equiv & \text{por sustitución} \\
 & n \geq 0 \wedge m \geq 0 \wedge \left(E = \log_2 n \wedge F = \log_2 m \Rightarrow \frac{E-F}{n-m} = 1\right)
\end{array}
$$

Paso 2: Se verifica que

$$
\left[(n \neq m \wedge n = 2 \wedge m = 1) \Rightarrow n \geq 0 \wedge m \geq 0 \wedge \left(E = \log_2 n \wedge F = \log_2 m \Rightarrow \frac{E-F}{n-m} = 1\right)\right]
$$

Se asume el antecedente cierto, entonces

$$
\begin{array}{cl}
 & n \geq 0 \wedge m \geq 0 \wedge \left(E = \log_2 n \wedge F = \log_2 m \Rightarrow \frac{E-F}{n-m} = 1\right) \\
\equiv & \text{porque } n = 2 \wedge m = 1 \\
 & V \wedge V \wedge (E = 1 \wedge F = 0 \Rightarrow E - F = 1) \\
\equiv & \text{por simplicación} \\
 & V \wedge V \wedge (E = 1 \wedge F = 0 \Rightarrow V) \\
\equiv & \text{por implicación} \\
 & V \wedge V \wedge V \\
\equiv & \text{por idempotencia} \\
 & V
\end{array}
$$

.

Tercer caso: Distintas llamadas de distintas funciones

Teorema 6 *Se quiere verificar la correctitud de la llamada*

$$\{P\}\,Ins\,\{Q\}$$

donde Ins es una instrucción que contiene n_1 distintas llamada a la función f_1

$$f_1\,(\bar{a}_{1,0})\,, \ldots, f_1\,(\bar{a}_{1,n_1-1})$$

n_2 distintas llamada a la función f_2

$$f_2\,(\bar{a}_{2,0})\,, \ldots, f_2\,(\bar{a}_{2,n_2-1})$$

$\ldots$ n_k distintas llamada a la función f_k

$$f_k\,(\bar{a}_{k,0})\,, \ldots, f_k\,(\bar{a}_{k,n_k-1})$$

con $\bar{a}_{t,i} \neq \bar{a}_{t,j}$ para todos $0 \leq i < j < n_t$; para todo $1 \leq t \leq k$. Entonces,
1)

$$
\begin{array}{rcl}
wp_{Ins}\,(Q) & \equiv & (\forall t : 1 \leq t \leq k : (\forall i : 0 \leq i < n : \; Pre\,(\bar{x}_t := \bar{a}_{t,i}))) \wedge \\
 & & \big((\forall t : 1 \leq t \leq k : (\forall i : 0 \leq i < n : Post\,(\bar{x}_t, f_t := \bar{a}_{t,i}, E_{t,i}))) \Rightarrow wp_{\underline{Ins}}Q\big)
\end{array}
$$

donde $\underline{Ins}$ es la instrucción Ins sustituyendo cada ocurrencia de $f_t\left(\bar{a}_{t,i}\right)$ por el símbolo $E_{t,i}$ que representa el valor de la función f_t ya evaluada en $\bar{a}_{t,i}$.

2) Se verifica que

$$\left[P \Rightarrow wp_{Ins}\left(Q\right)\right]$$

Ejemplo: Sean las especificaciones de funciones

$$\boxed{\begin{array}{rl} \textbf{fun } f & (x : \text{real}) \longrightarrow \text{real} \\ \text{Pre:} & \{V\} \\ \text{Post:} & \left\{f = x^2 - 2\right\} \end{array}}$$

y

$$\boxed{\begin{array}{rl} \textbf{fun } f' & (x : \text{real}) \longrightarrow \text{real} \\ \text{Pre:} & \{V\} \\ \text{Post:} & \{f' = 2 * x\} \end{array}}$$

Se quiere verificar la correctitud de la instrucción de asignación con dos llamadas de dos funciones distintas

$$\{z = 3\}\; z := z - \left(f(z)/f'(z)\right) \left\{z = \frac{11}{6}\right\}$$

Paso 1: Primero consideramos la precondición más débil para la instrucción auxiliar

$$z := z - (E/F)$$

donde los símbolos E y F representan los valores de las funciones f y f'evaluadas en el mismo argumento z :

$$\begin{array}{ll} . & \\ & wp_{z:=z-(E/F)}\left(z = \frac{11}{6}\right) \\ \equiv & \text{definición de asignación} \\ & \left(z = \frac{11}{6}\right)\left(z := z - (E/F)\right) \\ \equiv & \text{por sustitución} \\ & z - \frac{E}{F} = \frac{11}{6} \\ . & \end{array}$$

Ahora se busca la precondición más débil para la instrucción de asignación en cuestión

$$\begin{array}{ll} . & \\ & wp_{z:=z-(f(z)/f'(z))}\left(z = \frac{11}{6}\right) \\ \equiv & \text{por teorema} \\ & \left(\begin{array}{l} \qquad\qquad V\left(x := z\right) \wedge V\left(x := z\right) \wedge \\ \left(f = x^2 - 2\right)\left(x, f := z, E\right) \wedge \left(f = 2 * x\right)\left(x, f := z, F\right) \\ \Rightarrow \\ z - \frac{E}{F} = \frac{11}{6} \end{array}\right) \\ \equiv & \text{por sustitución} \\ & E = z^2 - 2 \wedge F = 2 * z \Rightarrow z - \frac{E}{F} = \frac{11}{6} \\ . & \end{array}$$

Paso 2: Se verifica que

$$\left[z = 3 \Rightarrow \left(E = z^2 - 2 \wedge F = 2 * z \Rightarrow z - \frac{E}{F} = \frac{11}{6}\right)\right]$$

Se asume el antecedente cierto, entonces

$$
\begin{array}{ll}
 & E = z^2 - 2 \wedge F = 2 * z \Rightarrow z - \frac{E}{F} = \frac{11}{6} \\
\equiv & \text{porque } z = 3 \\
 & E = 7 \wedge F = 6 \Rightarrow V \\
\equiv & \text{por implicación} \\
 & V
\end{array}
$$

3 Funciones recursivas

3.0.4 Sintaxis

fun f $(x_1 : \text{Tipo-1}; \ldots\ x_n : \text{Tipo-}n) \longrightarrow \text{Tipo-}f$

{Pre}

{Post}

{cota}

|[

Cuerpo

$>> E$

]|

3.0.5 Correctitud de las funciones recursivas

Para probar la correctitud de las funciones recursivas debe probarse primero la correctitud de las llamadas recursivas dentro del cuerpo de la función.

Primer caso: Una o varias llamadas similares

Teorema 7 *Dada la post-condición Q de la instrucción Ins donde se halla una llamada a función $f(e(\bar{x}))$ (o varias llamadas iguales), la pre-condición más débil viene dada por el predicado*

$$
\begin{array}{rcl}
wp_{Ins}(Q) & \equiv & Pre(\bar{x} := e(\bar{x})) \wedge \\
0 & \leq & (cota)(\bar{x} := e(\bar{x})) < cota\ \wedge \\
 & & \left(Post(\bar{x}, f := e(\bar{x}), E) \Rightarrow wp_{\underline{Ins}} Q\right)
\end{array}
$$

donde $e(\bar{x})$ es una expresión que depende del parámetro formal $\bar{x}$, y $\underline{Ins}$ es la instrucción Ins sustituyendo cada ocurrencia de $f(e(\bar{x}))$ por el símbolo E que representa el valor de la función f ya evaluada en $e(\bar{x})$.

Ejemplo: Sea la especificación e implementación de función recursiva

$$\begin{array}{|rl|}\hline \textbf{fun}\ fac & (n : \text{int}) \longrightarrow \text{int} \\ \text{Pre:} & \{n \geq 0\} \\ \text{Post:} & \{fac = (\Pi i : 1 \leq i \leq n : i)\} \\ \text{cota:} & \{n\} \\ |[& \text{if } (n = 0) \longrightarrow fac := 1 \\ & [\,]\ (n \geq 1) \longrightarrow fac := fac\,(n-1) * n \\ & \text{fi} \\ & >>\ fac \\ ]| & \\ \hline \end{array}$$

Se quiere verificar la correctitud de la función $fac\,(n)$ verifcando primero la correctitud de la instrucción

$$fac := fac\,(n-1) * n$$

que contiene la llamada recursiva $fac\,(n-1)$. Antes de aplicar el teorema anterior calculamos la pre-condición más débil para la instrucción auxiliar

$$fac := E * n$$

donde E representa el valor de la función f ya evaluada en $n-1$;

$$\begin{array}{ll} . & \\ & wp_{fac:=E*n}\,(fac = (\Pi i : 1 \leq i \leq n : i)) \\ \equiv & \text{por definición de asignación} \\ & (fac = (\Pi i : 1 \leq i \leq n : i))\,(fac := E * n) \\ \equiv & \text{por sustitución} \\ & E * n = (\Pi i : 1 \leq i \leq n : i) \\ \equiv & \text{por definición inductiva de } \Pi \text{ (separación de rango)} \\ & E * n = (\Pi i : 1 \leq i \leq n-1 : i) * n \\ \equiv & \text{por aritmética} \\ & E = (\Pi i : 1 \leq i \leq n-1 : i) \\ . & \end{array}$$

y por lo tanto,

$$\begin{array}{ll} . & \\ & wp_{fac:=n*fac(n-1)}\,(fac = (\Pi i : 1 \leq i \leq n : i)) \\ \equiv & \text{por teorema} \\ & \qquad (n \geq 0)\,(n := n-1) \wedge (0 \leq (n)\,(n := n-1) < n) \wedge \\ & ((fac = (\Pi i : 1 \leq i \leq n : i))\,(n, fac := n-1, E) \Rightarrow E = (\Pi i : 1 \leq i \leq n-1 : i)) \\ \equiv & \text{por sustitución} \\ & \qquad n \geq 1 \wedge 1 \leq n \wedge n-1 < n \wedge \\ & (E = (\Pi i : 1 \leq i \leq n-1 : i) \Rightarrow E = (\Pi i : 1 \leq i \leq n-1 : i)) \\ \equiv & \text{por idempotencia, identidad de } \wedge \text{, implicación y aritmética} \\ & n \geq 1 \\ . & \end{array}$$

Ahora para verificar la correctitud de la función fac debe probarse la tripleta

$$\begin{array}{l} \{n \geq 0\} \\ \text{if } (n = 0) \longrightarrow \{n = 0\}\, fac := 1 \\ [\,]\ (n \geq 1) \longrightarrow \{n \geq 1\}\, fac := fac\,(n-1) * n \\ \text{fi} \\ \{fac = (\Pi i : 1 \leq i \leq n : i)\} \end{array}$$

para la cual se verifica que

$$[n \geq 0 \Rightarrow wp_{\text{if-fi}}\,(fac = (\Pi i : 1 \leq i \leq n : i))]$$

Segundo caso: Distintas llamadas de la misma función

Teorema 8 *Dada la post-condición Q de la instrucción Ins donde se hallan las k llamadas distintas*

$$f\,(e_1\,(\bar{x}))\,, , f\,(e_k\,(\bar{x}))$$

la pre-condición más débil viene dada por el predicado

$$\begin{array}{rcl} wp_{Ins}\,(Q) & \equiv & (\forall i : 1 \leq i \leq k : Pre\,(\bar{x} := e_i\,(\bar{x})) \wedge 0 \leq (cota)\,(\bar{x} := e_i\,(\bar{x})) < cota) \wedge \\ & & \big((\forall i : 1 \leq i \leq k : Post\,(\bar{x}, f := e_i\,(\bar{x})\,, E_i)) \Rightarrow wp_{\underline{Ins}} Q\big) \end{array}$$

donde, para todo $1 \leq i \leq k$, $e_i\,(\bar{x})$ es una expresión que depende del parámetro formal $\bar{x}$, y $\underline{Ins}$ es la instrucción Ins sustituyendo cada ocurrencia de $f\,(e_i\,(\bar{x}))$ por el símbolo E_i que representa el valor de la función f ya evaluada en $e_i\,(\bar{x})$.

Ejemplo: Dada la especificación algebraica de la función de Fibonacci

$$\begin{array}{rl} F: & \mathbb{N} \longrightarrow \mathbb{N} \\ F\,(n) = & \begin{cases} 0 & ;\text{ si } n = 0 \\ 1 & ;\text{ si } n = 1 \\ F\,(n-1) + F\,(n-2) & ;\text{ si } n \geq 2 \end{cases} \end{array}$$

y la especificación e implementación recursiva de la función

```
fun fib   (n : int) ⟶ int
   Pre:   {n ≥ 0}
  Post:   {fib = F (n)}
     |[   if (n = 0) ⟶ fib := 0
          [] (n = 1) ⟶ fib := 1
          [] (n ≥ 2) ⟶ fib := fib (n − 1) + fib (n − 2)
          fi
          >> fib
     ]|
```

se quiere verificar la correctitud de la función $fib\,(n)$ verifcando primero la correctitud de la instrucción

$$fib := fib\,(n-1) + fib\,(n-2)$$

que contiene las llamadas recursivas distintas $fib\,(n-1)$ y $fib\,(n-2)$. Antes de aplicar el teorema anterior calculamos la pre-condición más débil para la instrucción auxiliar

$$fib := E_1 + E_2$$

donde E_1 y E_2 representan los valor respectivos de la función f ya evaluada en $n-1$ y $n-2$,

.

$$\begin{array}{ll}
& wp_{fib:=E_1+E_2}\,(fib = F\,(n)) \\
\equiv & \text{por definición de asignación} \\
& (fib = F\,(n))\,(fib := E_1 + E_2) \\
\equiv & \text{por sustitución} \\
& E_1 + E_2 = F\,(n) \\
\Leftarrow & \text{por definición inductiva de } F \\
& E_1 + E_2 = F\,(n-1) + F\,(n-2) \wedge n \geq 2 \\
\Leftarrow & \text{por álgebra} \\
& E_1 = F\,(n-1) \wedge E_2 = F\,(n-2) \wedge n \geq 2
\end{array}$$

.

y por lo tanto,

.

$$\begin{array}{ll}
& wp_{fib:=fib(n-1)+fib(n-2)}\,(fib = F\,(n)) \\
\equiv & \text{por teorema} \\
& \left(\begin{array}{c}
(n \geq 0)\,(n := n-1) \wedge (n \geq 0)\,(n := n-2) \wedge \\
0 \leq (n)\,(n := n-1) < n \wedge 0 \leq (n)\,(n := n-2) < n \wedge \\
(fib = F\,(n))\,(n, fib := n-1, E_1) \wedge (fib = F\,(n))\,(n, fib := n-2, E_2) \\
\Rightarrow E_1 = F\,(n-1) \wedge E_2 = F\,(n-2) \wedge n \geq 2
\end{array}\right) \\
\equiv & \text{por sustitución} \\
& \left(\begin{array}{c}
n \geq 1 \wedge n \geq 2 \wedge \\
1 \leq n \wedge n-1 < n \wedge 2 \leq n \wedge n-2 < n \wedge \\
E_1 = F\,(n-1) \wedge E_2 = F\,(n-2) \\
\Rightarrow E_1 = F\,(n-1) \wedge E_2 = F\,(n-2) \wedge n \geq 2
\end{array}\right) \\
\equiv & \text{por aritmética, identidad de } \wedge \text{ e intersección de extensiones} \\
& \left(\begin{array}{c}
n \geq 2 \wedge \\
E_1 = F\,(n-1) \wedge E_2 = F\,(n-2) \\
\Rightarrow E_1 = F\,(n-1) \wedge E_2 = F\,(n-2) \wedge n \geq 2
\end{array}\right) \\
\Leftarrow & \text{porque } [p \Rightarrow (q \Rightarrow q \wedge p)] \text{ e idempotencia de } \wedge \\
& n \geq 2
\end{array}$$

Ahora, para verificar la correctitud de la función fib debe probarse la tripleta

$$\begin{array}{l}
\{n \geq 0\} \\
\text{if } (n = 0) \longrightarrow \{n = 0\}\, fib := 0 \\
[]\ (n = 1) \longrightarrow \{n = 1\}\, fib := 1 \\
[]\ (n \geq 2) \longrightarrow \{n \geq 2\}\, fib := fib\,(n-1) + fib\,(n-2) \\
\text{fi} \\
\{fib = F(n)\}
\end{array}$$

para la cual se verifica que

$$[n \geq 0 \Rightarrow wp_{\text{if-fi}}\,(fib = F(n))]$$

Tercer caso: Distintas llamadas de distintas funciones

Teorema 9 *Dentro del cuerpo de la función f_1, dada la post-condición Q de la instrucción Ins donde se hallan k_1 distintas llamadas recursivas a la función f_1*

$$f_1\left(e_{1,1}\left(\bar{x}\right)\right),\ldots,f_1\left(e_{1,k_1}\left(\bar{x}\right)\right)$$

junto con k_2 distintas llamadas a la función f_2

$$f_2\left(e_{2,1}\left(\bar{x}\right)\right),\ldots,f_2\left(e_{2,k_2}\left(\bar{x}\right)\right)$$

... k_m distintas llamadas a la función f_m

$$f_m\left(e_{m,1}\left(\bar{x}\right)\right),\ldots,f_m\left(e_{m,k_m}\left(\bar{x}\right)\right)$$

con $e_{t,i}\left(\bar{x}\right) \neq e_{t,j}\left(\bar{x}\right)$ para todos $0 \leq i < j < k_t$; para todo $1 \leq t \leq m$, la pre-condición más débil viene dada por el predicado

$$\begin{aligned} wp_{Ins}\left(Q\right) \quad \equiv \quad & \left(\forall t : 1 \leq t \leq m : \left(\forall i : 1 \leq i \leq k_t : Pre_t\left(\bar{x} := e_{t,i}\left(\bar{x}\right)\right)\right)\right) \wedge \\ & \left(\forall i : 1 \leq i \leq k_1 : 0 \leq \left(cota_1\right)\left(\bar{x} := e_{1,i}\left(\bar{x}\right)\right) < cota_1\right) \wedge \\ & \big(\left(\forall t : 1 \leq t \leq m : \left(\forall i : 1 \leq i \leq k_t : Post\left(\bar{x}, f_t := e_{t,i}\left(\bar{x}\right), E_{t,i}\right)\right)\right) \Rightarrow wp_{\underline{Ins}}Q\big) \end{aligned}$$

donde, para todos $1 \leq t \leq m$ y $1 \leq i \leq k_t$, $e_{t,i}\left(\bar{x}\right)$ es una expresión que depende del parámetro formal $\bar{x}$, y $\underline{Ins}$ es la instrucción Ins sustituyendo cada ocurrencia de $f_t\left(e_{t,i}\left(\bar{x}\right)\right)$ por el símbolo $E_{t,i}$ que representa el valor de la función f ya evaluada en $e_{t,i}\left(\bar{x}\right)$.