Algoritmos y Estructuras de Datos

Guía Práctica 3
Verificación de programas (Parte 1)

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

3.1. Precondición más débil en SmallLang

Ejercicio 1. Calcular las siguientes expresiones, donde a, b son variables reales, i una variable entera y A es una secuencia de reales.

Aclaración. Recordar que siempre las variables y los valores (enteros, booleanos, etc.) se encuentran definidos.

a) $\text{def}(a+1) \equiv \text{def}(a) \wedge \text{def}(1) \equiv True$

b) $\text{def}(a/b) \equiv \text{def}(a) \wedge (\text{def}(b) \wedge_L b \neq 0) \equiv b \neq 0$

c) $\text{def}(\sqrt{a/b}) \equiv \text{def}(a/b) \wedge_L (a/b) \geq 0 \equiv b \neq 0 \wedge a/b \geq 0$

d) $\text{def}(A[i]+1) \equiv (\text{def}(A) \wedge \text{def}(i)) \wedge_L 0 \leq i < |A| \equiv 0 \leq i < |A|$

e) $\text{def}(A[i+2]) \equiv (\text{def}(A) \wedge \text{def}(i)) \wedge_L 0 \leq i+2 < |A| \equiv 0 \leq i+2 < |A|$

f) $\text{def}(0 \leq i \leq |A| \ \wedge_L \ A[i] \geq 0) \equiv (\text{def}(A) \wedge \text{def}(i)) \wedge_L 0 \leq i < |A| \equiv 0 \leq i < |A|$

Ejercicio 2. Calcular las siguientes precondiciones más débiles, donde a, b son variables reales, i una variable entera y A es una secuencia de reales.

a) $wp(\textbf{a := a+1; b := a/2}, b \geq 0) \equiv (a-1)/2 \geq 0$

$$\{\text{def}(a) \wedge_L (a-1)/2 \geq 0\} \equiv \{(a-1)/2 \geq 0\}$$
$$S_1 \equiv \textbf{a := a+1}$$
$$\{\text{def}(a) \wedge_L a/2 \geq 0\} \equiv \{a/2 \geq 0\}$$
$$S_2 \equiv \textbf{b := a/2}$$
$$Q \equiv \{b \geq 0\}$$

b) $wp(\textbf{a := A[i] + 1; b := a*a}, b \neq 2) \equiv (0 \leq i < |A|) \wedge_L (A[i] * A[i] + 2 * A[i] + 1 \neq 2)$.

$$\{(\text{def}(A) \wedge_L 0 \leq i < |A|) \wedge_L A[i] * A[i] + 2 * A[i] + 1 \neq 2\} \equiv \{(0 \leq i < |A|) \wedge_L (A[i] * A[i] + 2 * A[i] + 1 \neq 2)\}$$
$$S_1 \equiv \textbf{a := A[i] + 1}$$
$$\{\text{def}(a) \wedge_L a * a \neq 2\} \equiv \{a * a \neq 2\}$$
$$S_2 \equiv \textbf{b := a*a}$$
$$Q \equiv \{b \neq 2\}$$

c) $wp(\textbf{a := A[i] + 1; a := b*b}, a \geq 0) \equiv (0 \leq i < |A|) \wedge (b * b \geq 0)$.

$$\{(\text{def}(A) \wedge_L 0 \leq i < |A|) \wedge b * b \geq 0\} \equiv \{(0 \leq i < |A|) \wedge (b * b \geq 0)\}$$
$$S_1 \equiv \textbf{a := A[i] + 1}$$
$$\{\text{def}(b) \wedge_L b * b \geq 0\} \equiv \{b * b \geq 0\}$$
$$S_2 \equiv \textbf{a := b*b}$$
$$Q \equiv \{a \geq 0\}$$

d) $wp(\textbf{a := a-b; b := a+b}, a \geq 0 \wedge b \geq 0) \equiv a \geq b \wedge a \geq 0$.

$$\{(\text{def}(b) \wedge \text{def}(a)) \wedge_L (a - b \geq 0 \wedge a - b + b \geq 0)\} \equiv \{a \geq b \wedge a \geq 0\}$$
$$S_1 \equiv \textbf{a := a - b}$$
$$\{(\text{def}(b) \wedge \text{def}(a)) \wedge_L (a \geq 0 \wedge a + b \geq 0)\} \equiv \{a \geq 0 \wedge a + b \geq 0\}$$
$$S_2 \equiv \textbf{b := a + b}$$
$$Q \equiv \{a \geq 0 \wedge b \geq 0\}$$

Ejercicio 3. Sea $Q \equiv (\forall j : \mathbb{Z})(0 \leq j < |A| \rightarrow_L A[j] \geq 0)$. Calcular las siguientes precondiciones más débiles, donde i es una variable entera y A es una secuencia de reales.

a)

$$\begin{aligned}
wp(\textbf{A[i] := 0}, Q) &\equiv wp(setAt(A, i, 0), Q) \\
&\equiv ((\text{def}(A) \wedge \text{def}(i)) \wedge_L (0 \leq i < |A|)) \wedge_L Q^A_{setAt(A,i,0)} \\
&\equiv (0 \leq i < |A|) \wedge_L Q^A_{setAt(A,i,0)} \\
&\equiv (0 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |setAt(A, i, 0)| \rightarrow_L setAt(A, i, 0)[j] \geq 0) \quad (1) \\
&\equiv (0 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |A| \rightarrow_L ((i = j \rightarrow 0 \geq 0) \wedge (i \neq j \rightarrow A[j] \geq 0))) \quad (2) \\
&\equiv (0 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |A| \wedge i \neq j) \rightarrow_L A[j] \geq 0)
\end{aligned}$$

Como la operación $setAt$ no modifica la longitud de la secuencia, se tiene que $|setAt(A, i, 0)| = |A|$ en (1).

En (2) se puede utilizar la siguiente definición,

$$setAt(A, i, 0)[j] = \begin{cases} 0 & \text{si } i = j \\ A[j] & \text{si } i \neq j \end{cases}$$

y como $0 \geq 0 \equiv True$ se puede ver que $i = j \rightarrow 0 \geq 0$ es una tautología, por lo que el valor de verdad del consecuente del cuantificador universal sólo depende del segundo término de la conjunción, es decir, de $i \neq j \rightarrow A[j] \geq 0$.

b)

$$\begin{aligned}
wp(\textbf{A[i+2] := 0}, Q) &\equiv wp(setAt(A, i+2, 0), Q) \\
&\equiv ((\text{def}(A) \wedge \text{def}(i)) \wedge_L (0 \leq i+2 < |A|)) \wedge_L Q^A_{setAt(A,i+2,0)} \\
&\equiv (0 \leq i+2 < |A|) \wedge_L Q^A_{setAt(A,i,0)} \\
&\equiv (0 \leq i+2 < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |setAt(A, i+2, 0)| \rightarrow_L setAt(A, i+2, 0)[j] \geq 0) \\
&\equiv (0 \leq i+2 < |A|) \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |A| \wedge i+2 \neq j) \rightarrow_L A[j] \geq 0)
\end{aligned}$$

c)

$$\begin{aligned}
wp(\textbf{A[i+2] := -1}, Q) &\equiv wp(setAt(A, i+2, -1), Q) \\
&\equiv ((\text{def}(A) \wedge \text{def}(i)) \wedge_L (0 \leq i+2 < |A|)) \wedge_L Q^A_{setAt(A,i+2,-1)} \\
&\equiv (0 \leq i+2 < |A|) \wedge_L Q^A_{setAt(A,i,-1)} \\
&\equiv (0 \leq i+2 < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |setAt(A, i+2, -1)| \rightarrow_L setAt(A, i+2, -1)[j] \geq 0) \\
&\equiv (0 \leq i+2 < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |A| \rightarrow_L ((i+2 = j \rightarrow -1 \geq 0) \wedge (i+2 \neq j \rightarrow A[j] \geq 0))) \quad (3) \\
&\equiv (0 \leq i+2 < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |A| \rightarrow_L (False \wedge (i+2 \neq j \rightarrow A[j] \geq 0))) \\
&\equiv (0 \leq i+2 < |A|) \wedge_L |A| = 0) \\
&\equiv (0 \leq i+2 < |A|) \wedge_L |A| = 0) \quad (4) \\
&\equiv False
\end{aligned}$$

En (3) se puede ver que cuando $i+2 = j$ no se va a poder cumplir el consecuente. Esto sucede, intuitivamente, porque Q nos pide que todos los valores de la secuencia sean cero o positivos, pero estamos fijando un valor en -1. Claramente no vamos a poder cumplir con Q a menos que la secuencia esté vacía.

Pero en (4) se nos requiere que $0 \leq i+2 < |A|$, es decir, $0 < |A|$ y $0 = |A|$. Esto resulta en una contradicción.

d)

$$\begin{aligned}
wp(\textbf{A[i] := 2 * A[i]}, Q) &\equiv wp(setAt(A, i, 2 * A[i]), Q) \\
&\equiv ((\text{def}(A) \wedge \text{def}(i)) \wedge_L (0 \leq i < |A|)) \wedge_L Q^A_{setAt(A,i,2*A[i])} \\
&\equiv (0 \leq i < |A|) \wedge_L Q^A_{setAt(A,i,2*A[i])} \\
&\equiv (0 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |setAt(A, i, 2 * A[i])| \rightarrow_L setAt(A, i, 2 * A[i])[j] \geq 0) \\
&\equiv (0 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |A|) \rightarrow_L setAt(A, i, 2 * A[i])[j] \geq 0) \\
&\equiv (0 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |A|) \rightarrow_L A[j] \geq 0)
\end{aligned}$$

Intuitivamente, como en cada posición duplico el valor que tenía previamente, es claro ver que necesito tener un valor que previamente hubiera sido cero o positivo, pues $2 * k \geq 0 \leftrightarrow k \geq 0/2 \leftrightarrow k \geq 0$.

e)

$$
\begin{aligned}
wp(\textbf{A[i] := A[i-1]}, Q) &\equiv wp(setAt(A, i, A[i-1]), Q) \\
&\equiv ((\text{def}(A) \wedge \text{def}(i) \wedge \text{def}(A[i-1])) \wedge_L (0 \leq i < |A|)) \wedge_L Q^A_{setAt(A,i,A[i-1])} \\
&\equiv ((\text{def}(A) \wedge \text{def}(i) \wedge (\text{def}(A) \wedge_L 0 \leq i-1 < |A|)) \wedge_L (0 \leq i < |A|)) \wedge_L Q^A_{setAt(A,i,A[i-1])} \\
&\equiv (1 \leq i < |A|) \wedge_L Q^A_{setAt(A,i,A[i-1])} \\
&\equiv (1 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |setAt(A, i, A[i-1])| \rightarrow_L setAt(A, i, A[i-1])[j] \geq 0) \\
&\equiv (1 \leq i < |A|) \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |A| \wedge j \neq i) \rightarrow_L A[j] \geq 0)
\end{aligned}
$$

Ejercicio 4. Calcular $wp(S, Q)$, para los siguientes pares de programas S y postcondiciones Q.
Sabemos que $wp(\textbf{if B then S1 else S2 endif}, Q) \equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q)))$

a) $\textbf{B} \equiv a < 0$, $\textbf{S1} \equiv b := a$, $\textbf{S2} \equiv b := -a$, $\textbf{Q} \equiv (b = -|a|)$

$$
\begin{aligned}
wp(\textbf{if B then S1 else S2 endif}, Q) &\equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q))) \\
&\equiv \text{def}(a) \wedge_L (\\
&\qquad (a < 0 \wedge (\text{def}(a) \wedge_L a = -|a|)) \vee \\
&\qquad (a \geq 0 \wedge (\text{def}(a) \wedge_L -a = -|a|)) \\
&\quad) \\
&\equiv ((a < 0 \wedge a = -|a|) \vee (a \geq 0 \wedge a = |a|)) \\
&\equiv True
\end{aligned}
$$

b) $\textbf{B} \equiv a < 0$, $\textbf{S1} \equiv b := a$, $\textbf{S2} \equiv b := -a$, $\textbf{Q} \equiv (b = |a|)$

$$
\begin{aligned}
wp(\textbf{if B then S1 else S2 endif}, Q) &\equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q))) \\
&\equiv \text{def}(a) \wedge_L (\\
&\qquad (a < 0 \wedge (\text{def}(a) \wedge_L a = |a|)) \vee \\
&\qquad (a \geq 0 \wedge (\text{def}(a) \wedge_L -a = |a|)) \\
&\quad) \\
&\equiv ((a < 0 \wedge a = |a|) \vee (a \geq 0 \wedge -a = |a|)) \\
&\equiv False
\end{aligned}
$$

c) $\textbf{B} \equiv i > 0$, $\textbf{S1} \equiv s[i] := 0$, $\textbf{S2} \equiv s[0] := 0$, $\textbf{Q} \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \geq 0)$

$$
\begin{aligned}
wp(\textbf{if B then S1 else S2 endif}, Q) &\equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q))) \\
&\equiv \text{def}(i) \wedge_L (\\
&\qquad (i > 0 \wedge (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L Q^s_{setAt(s,i,0)})) \vee \\
&\qquad (i \leq 0 \wedge (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L Q^s_{setAt(s,0,0)})) \\
&\quad) \\
&\equiv ((0 < i < |s| \wedge_L Q^s_{setAt(s,i,0)}) \vee (i = 0 \wedge_L Q^s_{setAt(s,0,0)})) \\
&\equiv (0 \leq i < |s| \wedge_L Q^s_{setAt(s,i,0)}) \\
&\equiv (0 \leq i < |s| \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] \geq 0))
\end{aligned}
$$

d) $\mathbf{B} \equiv i > 1$, $\mathbf{S1} \equiv s[i] := s[i-1]$, $\mathbf{S2} \equiv s[i] := 0$, $\mathbf{Q} \equiv (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L s[j] = s[j-1])$

$$
\begin{aligned}
wp(\textbf{if B then S1 else S2 endif}, Q) &\equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q))) \\
&\equiv \text{def}(i) \wedge_L (\\
&\qquad (i > 1 \wedge (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L Q^s_{setAt(s,i,s[i-1])})) \vee \\
&\qquad (i \leq 1 \wedge (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L Q^s_{setAt(s,i,0)})) \\
&\quad) \\
&\equiv ((1 < i < |s| \wedge_L Q^s_{setAt(s,i,s[i-1])}) \vee (0 \leq i \leq 1 \wedge_L Q^s_{setAt(s,i,0)})) \\
&\equiv (\\
&\qquad (1 < i < |s| \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L setAt(s, i, s[i-1])[j] = setAt(s, i, s[i-1])[j-1] \\
&\qquad (0 \leq i \leq 1 \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L setAt(s, i, 0)[j] = setAt(s, i, 0)[j-1])) \\
&\quad) \\
&\equiv (\\
&\qquad (1 < i < |s| \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \wedge i \neq j \rightarrow_L s[j] = s[j-1])) \vee \\
&\qquad (i = 0 \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L setAt(s, i, s[i-1])[j] = setAt(s, i, s[i-1])[j-1])) \\
&\qquad (i = 1 \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L setAt(s, i, s[i-1])[j] = setAt(s, i, s[i-1])[j-1])) \\
&\quad) \\
&\equiv (\\
&\qquad (1 < i < |s| \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \wedge i \neq j \rightarrow_L s[j] = s[j-1])) \vee \\
&\qquad (i = 0 \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L s[j] = s[j-1]) \\
&\qquad (i = 1 \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \rightarrow_L (i = j \rightarrow s[0] = 0 \wedge i \neq j \rightarrow s[j] = s[j-1]))) \\
&\quad) \\
&\equiv (\\
&\qquad (1 < i < |s| \wedge_L (\forall j : \mathbb{Z})(1 \leq j < |s| \wedge i \neq j \rightarrow_L s[j] = s[j-1])) \vee \\
&\qquad (0 \leq i \leq 1 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \wedge i \neq j \rightarrow_L (s[j] = 0))) \\
&\quad)
\end{aligned}
$$

e) $\mathbf{B} \equiv s[i] < 0$, $\mathbf{S1} \equiv s[i] := -s[i-1]$, $\mathbf{S2} \equiv skip$, $\mathbf{Q} \equiv (0 \leq i < |s| \ \wedge_L \ s[i] \geq 0)$

$$
\begin{aligned}
wp(\textbf{if B then S1 else S2 endif}, Q) &\equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q))) \\
&\equiv \text{def}(i) \wedge_L (\\
&\qquad (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L s[i] < 0 \wedge Q^s_{setAt(s,i,-s[i-1])}) \vee \\
&\qquad (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L s[i] \geq 0 \wedge Q) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L ((s[i] < 0 \wedge_L Q^s_{setAt(s,i,-s[i-1])}) \vee (s[i] \geq 0 \wedge_L Q)) \\
&\equiv (0 \leq i < |s|) \wedge_L ((s[i] < 0 \wedge_L (0 \leq i < |s| \wedge setAt(s, i, -s[i-1])[i] \geq 0) \vee \\
&\quad (s[i] \geq 0 \wedge_L (0 \leq i < |s| \wedge s[i] \geq 0))) \\
&\equiv (0 \leq i < |s|) \wedge_L ((s[i] < 0 \wedge_L -s[i-1] \geq 0) \vee (s[i] \geq 0 \wedge_L s[i] \geq 0)) \\
&\equiv (0 \leq i < |s|) \wedge_L ((s[i] < 0 \wedge_L s[i-1] \leq 0) \vee s[i] \geq 0) \\
&\equiv (0 \leq i < |s|) \wedge_L (s[i-1] < 0 \vee s[i] \geq 0) \\
&\equiv (0 \leq i < |s|) \wedge_L True \\
&\equiv (0 \leq i < |s|)
\end{aligned}
$$

f) $\mathbf{B} \equiv s[i] > 0$, $\mathbf{S1} \equiv s[i] := -s[i]$, $\mathbf{S2} \equiv skip$, $\mathbf{Q} \equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \ \rightarrow_L \ s[j] \geq 0)$

$$
\begin{aligned}
wp(\textbf{if B then S1 else S2 endif}, Q) &\equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q))) \\
&\equiv \text{def}(i) \wedge_L (\\
&\qquad (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L s[i] > 0 \wedge Q^s_{setAt(s,i,-s[i])}) \vee \\
&\qquad (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L s[i] \leq 0 \wedge Q) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L ((s[i] > 0 \wedge_L Q^s_{setAt(s,i,-s[i])}) \vee (s[i] \leq 0 \wedge_L Q)) \\
&\equiv (0 \leq i < |s|) \wedge_L ((s[i] > 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L setAt(s, i, -s[i])[j] \geq 0)) \vee \\
&\quad (s[i] \leq 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \geq 0))) \\
&\equiv (0 \leq i < |s|) \wedge_L (\\
&\quad (s[i] > 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L (i = j \rightarrow s[j] = 0 \wedge i \neq j \rightarrow s[j] \geq 0]))) \vee \\
&\quad (s[i] \leq 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L (i = j \rightarrow s[j] = 0 \wedge i \neq j \rightarrow s[j] \geq 0])))) \\
&\equiv (0 \leq i < |s|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L (i = j \rightarrow s[j] = 0 \wedge i \neq j \rightarrow s[j] \geq 0]))
\end{aligned}
$$

Ejercicio 5. Para las siguientes especificaciones:

- Poner nombre al problema que resuelven
- Escribir un programa S sencillo en SmallLang, sin ciclos, que lo resuelva
- Dar la precondición más débil del programa escrito con respecto a la postcondición de su especificación

a) `proc agregaI-EsimoALaSuma` (in s: $seq\langle\mathbb{Z}\rangle$, in i: $\mathbb{Z}$, inout a: $\mathbb{Z}$)

`requiere` $\{0 \leq i < |s| \ \wedge_L \ a = \sum_{j=0}^{i-1} s[j]\}$

`asegura` $\{a = \sum_{j=0}^{i} s[j]\}$

```
1  func agregaI-EsimoALaSuma(s <int>, i int, a int) int {
2      a := a + s[i];
3      return a;
4  }
```

b) `proc hastaI-EsimoSonPositivos` (in s: $seq\langle\mathbb{Z}\rangle$, in i: $\mathbb{Z}$) : Bool

`requiere` $\{0 \leq i < |s| \ \wedge_L \ (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow_L s[j] \geq 0)\}$

`asegura` $\{res = true \leftrightarrow (\forall j : \mathbb{Z})(0 \leq j \leq i \rightarrow_L s[j] \geq 0)\}$

```
1  func hastaI-EsimoSonPositivos(s <int>, i int) bool {
2      return s[i] >= 0;
3  }
```

c) `proc sucesionDeFibonacciHastaI` (inout s: $seq\langle\mathbb{Z}\rangle$, in i: $\mathbb{Z}$)

`requiere` $\{(0 \leq i < |s|) \wedge_L (\forall j : \mathbb{Z})(0 \leq j < i \rightarrow s[j] = fibonacci(j))\}$

`asegura` $\{(\forall j : \mathbb{Z})(0 \leq j \leq i \rightarrow s[j] = fibonacci(j))\}$

```
1  func sucesionDeFibonacciHastaI(s <int>, i int) <int> {
2      if i >= 2 {
3          s[i] := s[i-1] + s[i-2];
4      } else {
5          // Como s[0] = 0 y s[1] = 1...
6          s[i] = i;
7      }
8
9      return s;
10 }
```

Ejercicio 6. Dada la poscondición $Q \equiv \{(\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \ mod \ 2 = 0)\}$ y el siguiente código

```
1       if (i mod 3 = 0)
2           s[i] := s[i] + 6;
3       else
4           s[i] := i;
5       endif
```

a) Demostrar que las siguientes WPs son incorrectas dando un contraejemplo

   I) $P \equiv \{0 \leq i \leq |s| \wedge_L i \ mod \ 3 = 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \ mod \ 2 = 0)\}$
   Si tomo un i que no sea múltiplo de 3 pero que sea par, al entrar a la rama *else* se le asignará el valor de la posición al elemento de la secuencia. De esta manera, la posición seguirá teniendo un valor par. Por lo tanto, encontramos un valor que hace verdadera la poscondición pero no está contemplada en la WP presentada.
   Contraejemplo. $i = 2, s = [0, 4, 20, 10, 8, 6]$

   II) $P \equiv \{0 \leq i < |s| \wedge_L i \ mod \ 3 \neq 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \ mod \ 2 = 0)\}$
   Si tomo un i que sea múltiplo de 3, al entrar a la rama *then* se le sumará seis al valor de esa posición. De esta manera, la posición seguirá teniendo un valor par ya que por precondición tenía un valor par y el seis es un número par. Por lo tanto, encontramos un valor que hace verdadera la poscondición pero no está contemplada en la WP presentada.
   Contraejemplo. $i = 3, s = [0, 4, 20, 10, 8, 6]$

   III) $P \equiv \{i \ mod \ 3 = 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \ mod \ 2 = 0)\}$
   La precondición presentada permite recibir valores de i que no estén en rango. Esto provocará un error al intentar escribir en esa posición de la secuencia.
   Contraejemplo. $i = -3, s = [0, 4, 20, 10, 8, 6]$

   IV) $P \equiv \{0 \leq i < |s|/2 \wedge_L i \ mod \ 3 = 0 \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \ mod \ 2 = 0)\}$
   Si tomo un i que se encuentre entre $|s|/2$ y $|s|$, al entrar en cualquiera de las dos ramas se obtendrá un valor par en esa posición según lo explicado en los ítems I y II. Por lo tanto, encontramos un valor que hace verdadera la poscondición pero no está contemplada en la WP presentada.
   Contraejemplo. $i = 4, s = [0, 4, 20, 10, 8, 6]$

b) La siguiente WP es incorrecta pero no se puede dar un contraejemplo para demostrarlo. ¿Por qué sucede esto?

   $P \equiv \{0 \leq i < |s| \wedge_L (i \ mod \ 3 = 0 \vee i \ mod \ 2 = 0) \wedge (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L s[j] \ mod \ 2 = 0)\}$

   Para probar que no es correcta debería poder elegir un valor de i que haga verdadera la poscondición pero que no esté contemplada en la WP propuesta. Vemos que i no puede ser múltiplo de 3, por lo que no ingresará a la rama *then*. Tampoco puede ser un número par, por lo que al ingresar a la rama *else* no se asignará un valor par.

   Sin embargo, podemos proponer una precondición más débil que P que también garantice que se cumpla la poscondición. Por lo tanto, P es una precondición válida pero no es la más débil, es por eso que no podemos pensar un contraejemplo.

3.1.1. Ejercicios de parcial

Ejercicio 7. Dado el siguiente condicional determinar la precondición más débil que permite hacer valer la poscondición (Q) propuesta. Se pide:

- Describir en palabras la WP esperada
- Derivarla formalmente a partir de los axiomas de precondición más débil. Para obtener el puntaje máximo deberá simplificarla lo más posible.

a) La precondición más débil debería requerir que i esté en rango para que no se indefina el código, que el valor en la posición i-ésima sea negativo y que las demás posiciones del arreglo tengan valores positivo.

   Se puede ver que si el valor de la posición i-ésima no fuera negativo, entonces el código S pondría un cero en esa posición y no se cumpliría la poscondición Q.

   Ahora vamos a calcularla **formalmente**.

   B $\equiv s[i] < 0$, **S1** $\equiv s[i] := -s[i]$, **S2** $\equiv s[i] := 0$, **Q** $\equiv (\forall j : \mathbb{Z})(0 \leq j < |s| \ \rightarrow_L \ s[j] > 0)$

$$
\begin{aligned}
wp(\textbf{if B then S1 else S2 endif}, Q) &\equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q))) \\
&\equiv ((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L (\\
&\quad (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L s[i] < 0 \wedge Q^s_{setAt(s,i,-s[i])}) \vee \\
&\quad (((\text{def}(s) \wedge \text{def}(i)) \wedge_L 0 \leq i < |s|) \wedge_L s[i] \geq 0 \wedge Q^s_{setAt(s,i,0)}) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L (\\
&\quad (s[i] < 0 \wedge_L Q^s_{setAt(s,i,-s[i])}) \vee \\
&\quad (s[i] \geq 0 \wedge_L Q^s_{setAt(s,i,0)}) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L (\\
&\quad (s[i] < 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L setAt(s, i, -s[i])[j] > 0)) \vee \\
&\quad (s[i] \geq 0 \wedge_L (\forall j : \mathbb{Z})(0 \leq j < |s| \rightarrow_L setAt(s, i, 0)[j] > 0)) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L (\\
&\quad (s[i] < 0 \wedge_L ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge -s[i] > 0)) \vee \\
&\quad (s[i] \geq 0 \wedge_L ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge 0 > 0)) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L (\\
&\quad (s[i] < 0 \wedge_L ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge True)) \vee \\
&\quad (s[i] \geq 0 \wedge_L ((\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0) \wedge False)) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L (\\
&\quad (s[i] < 0 \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0)) \vee \\
&\quad (s[i] \geq 0 \wedge_L False) \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L (\\
&\quad (s[i] < 0 \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0)) \vee \\
&\quad False \\
&\quad) \\
&\equiv (0 \leq i < |s|) \wedge_L (s[i] < 0 \wedge_L (\forall j : \mathbb{Z})((0 \leq j < |s| \wedge i \neq j) \rightarrow_L s[j] > 0))
\end{aligned}
$$

Aclaración. Se puede ver que $-s[i] > 0$ es una tautología (evalúa siempre a $True$) porque justamente B nos está diciendo que vale que $s[i] < 0$. En el otro caso, es evidente por qué $0 > 0$ es una contradicción, lo que termina cancelando todo el término correspondiente a la rama *else* del condicional.

b) se sabe que al finalizar el programa, a tiene que ser un *cuadrado perfecto*. Si se cumple la guarda (o sea, si a es par), entonces a pasa a ser $a * a$, o sea, un cuadrado perfecto. Por lo tanto se va a cumplir la postcondición no importa cuál sea el valor de a. En cambio si no se cumple la guarda (si a es impar), a se va a transformar en un número negativo. Como un número negativo nunca puede ser un cuadrado perfecto, nunca se va a poder cumplir la ppostcondición. En resumen, para que se cumpla la postcondición, *a tiene que ser par*.

Veamos formalmente: $\mathbf{B} \equiv a \mod 2 = 0$, $\mathbf{S1} \equiv a := a * a$, $\mathbf{S2} \equiv a := -|a|$, $\mathbf{Q} \equiv (\exists j : \mathbb{Z})(j \geq 0 \wedge j^2 = a)$

$wp(\textbf{if B then S1 else S2 endif}, Q) \equiv \text{def}(B) \wedge_L ((B \wedge wp(S1, Q)) \vee (\neg B \wedge wp(S2, Q)))$

$wp(S1, Q) \equiv def(a * a) \wedge_L Q^a_{a*a} \equiv (\exists j : \mathbb{Z})(j \geq 0 \wedge j^2 = a * a)$
esto es *verdadero* pues para $j = a$ se cumple que $j^2 = a * a$

$wp(S2, Q) \equiv def(-|a|) \wedge_L Q^a_{-|a|} \equiv (\exists j : \mathbb{Z})(j \geq 0 \wedge j^2 = -|a|)$
esto es *falso* pues no existe ningún número que elevado al cuadrado de negativo.

Uniendo todo:

$wp(\textbf{if B then S1 else S2 endif}, Q) \equiv ((a \mod 2 = 0) \wedge true) \vee ((a \mod 2 = 0) \wedge false) \equiv a \mod 2 = 0$

que es lo que esperábamos.