DEPARTAMENTO DE COMPUTACIÓN
UNIVERSIDADE DA CORUÑA

Tecnología de la programación
Ingeniería Técnica en Informática de Sistemas. Curso 2005/06
Convocatoria Extraordinaria de Diciembre (15 de Diciembre de 2005)

1. El predicado $(\forall I \in [0 : n] : (\sum J \in [0 : I] : b[J] \bmod 2) = 0)$

(a) significa que todos los elementos de b son múltiplos de 2

(b) equivale a $\neg(\exists I \in [0 : n] : (\forall J \in [0 : I] : (b[J] + b[I]) \bmod 2 \neq 0))$

(c) contiene a la variable n como variable ligada

(d) significa que todas las sumas de elementos consecutivos de b dan como resultado un número par

(a) No necesariamente, ya que la suma de dos elementos impares también cumple el predicado del enunciado.

(b) No. Este predicado no es equivalente al del enunciado ya que sólo contempla la suma de elementos de dos en dos.

(c) No. La variable n aparece como variable libre en la expresión.

(d) Sí.

2. Si tenemos que $\{Q\}S\{j > 0\}$ entonces:

(a) $\{Q\}S\{j \geq 0\}$

(b) $\{Q \wedge j = 0\}S\{j < 0\}$

(c) $\{Q \vee A\}S\{j > 0\}$

(d) $wp(S, j > 0) \Rightarrow wp(S, j \neq 0)$

(a) Sí. Estamos exigiendo una postcondición más débil que la original.

(b) No. S no puede acabar con $j > 0$ y $j < 0$ ciertos.

(c) No. No se puede cambiar una precondición Q por una fórmula más débil $Q \vee A$ que puede contemplar más casos no previstos en Q.

(d) Si. Dado que $j > 0 \Rightarrow j \neq 0$, por una propiedad del *wp* de la implicación obtenemos que: $wp(S, j > 0) \Rightarrow wp(S, j \neq 0)$.

3. Dado el programa $i, j, k := 0, 0, 0;$ **do** $i < 3 \rightarrow i, k := i + 1, k + 1; \mid j < 5 \rightarrow j, k := j + 1, k + 1$ **od**

(a) es equivalente a $i, j := 0, 0;$ **do** $i < 3 \wedge j < 5 \rightarrow i, j, k := i + 1, j + 1, k + 1;$ **od**

(b) es determinista si inicialmente $i \geq 3$ o $j \geq 5$

(c) es equivalente a $i, j, k := 3, 5, 8$

(d) siempre es no determinista

(a) No. Para empezar, en el programa propuesto la k no se inicializa a cero. Pero lo que es más importante, mientras el programa original debe ejecutar las tres iteraciones para la i y las cinco para la j sin un orden preestablecido (esto es, de forma no determinista), en el programa propuesto, en cuanto i alcanza el valor 3, el bucle finaliza. Al constar de una única rama, el bucle propuesto ejecuta exactamente tres iteraciones, con lo que la variable j termina también con valor 3 y la k se incrementa tres veces.

(b) No. El valor inicial de i, j o k es irrelevante, ya que siempre se comienza inicializando esas variables a cero. Después, nada más comenzar el bucle tenemos que tanto $i < 3$ como $j < 5$ son ciertos, con lo que el programa debe ejecutar una de las dos ramas de modo no determinista. Esta elección no determinista se repetirá hasta que una de las dos variables alcance su límite (a partir de ahí, el programa ya sólo puede ejecutar la rama que incrementa la otra variable). De este modo, el programa siempre se enfrenta a alguna elección no determinista a lo largo de su ejecución, sin importar el estado inicial.

(c) Sí. Puesto que el bucle debe ejecutar las 3 iteraciones de i y las 5 de j, k se incrementa en todas ellas, es decir $3 + 5 = 8$ veces. Es necesario destacar que aunque el programa inicial era un bucle y el propuesto es una asignación, para comprobar su equivalencia tan sólo se tiene en cuenta el estado final que alcanzan.

(d) Si. La explicación es la misma que la de la segunda respuesta.

$\boxed{\mathbf{4.}}$ Determinar cuales de las siguientes opciones se corresponden con la especificación de un algoritmo S que calcula cuántos elementos de un array $v[1:n] : integer$ son múltiplos de un cierto número k:

(a) $\{Q : k < n\}\ S\ \{R : r = (\mathbf{N}I \in [1:n] : v[k] \text{ mod } I = 0)\}$

(b) $\{Q : k > 0\}\ S\ \{R : r = (\mathbf{N}I \in [1:n] : v[I] \text{ mod } k = 0)\}$

(c) $\{Q : T\}\ S\ \{R : r = (\mathbf{N}I \in [1:n] : v[I] \text{ div } k = 0)\}$

(d) Ninguna de las anteriores.

(a) No. En este caso la postcondición indica cuantos elementos del array son múltiplos de la posición que ocupan dentro del mismo array.

(b) Sí.

(c) No. En este caso la postcondición indica cuantos elementos del array son menores que cierto número k

(d) No. La segunda respuesta es correcta.

$\boxed{\mathbf{5.}}$ (2 puntos) Calcula y simplifica: $wp(\text{"}b[i] := b[j]; b[j] := b[i]\text{"}, b[b[i]] = b[j])$

$$wp(\text{"}b[i] := b[j]; b[j] := b[i]\text{"}, b[b[i]] = b[j]) \equiv wp(\text{"}b[i] := b[j]\text{"}, wp(\text{"}b[j] = b[i]\text{"}, b[b[i]] = b[j]))$$

Primero calcularemos:

$$wp(\text{"}b[j] := b[i]\text{"}, b[b[i]] = b[j]) = dominio(b[i]) \text{ cand } enrango(b,i) \text{ cand } enrango(b,j) \text{ cand } (b[b[i]] = b[j])^b_{(b;j:b[i])}$$

Desarrollando el último término obtenemos:

$$(b[b[i]] = b[j])^b_{(b;j:b[i])}$$

$$\equiv (b;j:b[i])[(b;j:b[i])[i]] = (b;j:b[i])[j])$$

$$\equiv (b;j:b[i])[(b;j:b[i])[i]] = b[i]$$

$$\equiv (j = i \wedge (b;j:b[i])[b[i]] = b[i]) \vee (j \neq i \wedge (b;j:b[i])[b[i]] = b[i])$$

Sacando factor común el segundo término de ambas conjunciones tenemos:

$\equiv (j = i \vee j \neq i) \wedge ((b; j : b[i])[b[i]] = b[i])$

$\equiv (b; j : b[i])[b[i]] = b[i]$

$\equiv (j = b[i] \wedge b[i] = b[i]) \vee (j \neq b[i] \wedge b[b[i]] = b[i])$

$\equiv j = b[i] \vee (j \neq b[i] \wedge b[b[i]] = b[i])$

$\equiv j = b[i] \vee b[b[i]] = b[i]$

A continuación calculamos:

$\equiv wp(\text{"}b[i] := b[j]\text{"}, dominio(b[i]) \text{ cand } enrango(b,i) \text{ cand } enrango(b,j) \text{ cand } (j = b[i] \vee b[b[i]] = b[i])$

$\equiv dominio(b[i]) \text{ cand } enrango(b,i) \text{ cand } enrango(b,j) \text{ cand}$
$dominio(b[j]) \text{ cand } enrango(i,(b;i:b[j])) \text{ cand } enrango(j,(b;i:b[j])) \text{ cand } (j = b[i] \vee b[b[i]] = b[i])^{b}_{(b;i:b[j])}$

$\equiv dominio(b[i]) \text{ cand } enrango(b,i) \text{ cand } enrango(b,j) \text{ cand } dominio(b[j]) \text{ cand}$
$enrango(b,i) \text{ cand } enrango(b,j) \text{ cand } (j = b[i] \vee b[b[i]] = b[i])^{b}_{(b;i:b[j])}$

$\equiv dominio(b[i]) \text{ cand } dominio(b[j]) \text{ cand } enrango(b,i) \text{ cand } enrango(b,j) \text{ cand } (j = b[i] \vee b[b[i]] = b[i])^{b}_{(b;i:b[j])}$

Desarrollando el último término obtenemos:

$(j = b[i] \vee b[b[i]] = b[i])^{b}_{(b;i:b[j])}$

$\equiv (j = (b; i : b[j])[i] \vee (b; i : b[j])[(b; i : b[j])[i]] = (b; i : b[j])[i])$

$\equiv (j = b[j] \vee (b; i : b[j])[b[j]] = b[j])$

$\equiv (j = b[j] \vee (i = b[j] \wedge b[j] = b[j]) \vee (i \neq b[j] \wedge b[b[j]] = b[j]))$

$\equiv (j = b[j] \vee i = b[j] \vee (i \neq b[j] \wedge b[b[j]] = b[j]))$

$\equiv (j = b[j] \vee i = b[j] \vee b[b[j]] = b[j])$

Con lo que nos queda como resultado final:

$\boxed{dominio(b[i]) \text{ cand } dominio(b[j]) \text{ cand } enrango(b,i) \text{ cand } enrango(b,j) \text{ cand } (j = b[j] \vee i = b[j] \vee b[b[j]] = b[j])}$

6. (4 puntos) Un array es un **monte** si su primera mitad es creciente, la segunda decreciente y el vector completo es un palíndromo. Dado un array $b[0 : n-1] : integer$ con $n > 0$, se desea escribir un programa que determine si dicho array es un monte. Se pide:

6.1 Establecer una precondición y una postcondición adecuadas

Estudiando el enunciado del problema vemos que parte de la descripción de la propiedad *ser monte* es redundante, ya que se nos dice que el vector debe ser palíndromo, esto es, que aquellos elementos que se encuentren a la misa distancia del centro han de ser iguales, y que la primera mitad ha de ser creciente y la segunda decreciente. Pues bien, si se cumplen las dos primeras condiciones la tercera se satisfará necesariamente, por lo cual, simplificaremos la descripción de la postcondición al eliminarla. La precondición y postcondición resultantes son:

$$\{Q: \ b[0:n-1] : integer \wedge n > 0 \wedge b = B\}$$
$$\{R: \ monte = (\forall J \in [0 : \tfrac{n}{2} - 1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])\}$$

6.2 Fijar una invariante y una función cota

Una de las maneras de resolver el problema pasa por debilitar R cambiando la constante $\frac{n}{2}$ por una variable i. De esta forma, invariante y cota quedan como sigue:

$$\{P: \ 0 \leq i \leq \tfrac{n}{2} \wedge monte = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])\}$$
$$\{t: \ \tfrac{n}{2} - i\}$$

6.3 Escribir el programa

$$
\begin{array}{l}
\{Q\} \\
S_0 : i, monte := 0, T; \\
\{P\} \\
\{t\} \\
\text{do } B : i \neq \frac{n}{2} \rightarrow \ \text{if } B_1 : b[i] \leq b[i+1] \wedge b[i] = b[n-1-i] \rightarrow S_1 : skip; \\
\qquad\qquad\qquad \| \ B_2 : b[i] > b[i+1] \vee b[i] \neq b[n-1-i] \rightarrow S_2 : monte := F; \\
\qquad\qquad\qquad \text{fi} \\
\qquad\qquad\qquad S_3 : i := i+1; \\
\text{od} \\
\{R\}
\end{array}
$$

6.4 Demostrar su corrección *total*

a) $Q \Rightarrow wp(S_0, P)$

$$
\begin{array}{l}
b[0:n-1] : integer \wedge n > 0 \wedge b = B \Rightarrow wp(\text{"}i, monte := 0, T\text{"}, P) \\
\equiv n > 0 \Rightarrow 0 \leq 0 \leq \frac{n}{2} \wedge T = (\forall J \in [0:-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J]) \\
\text{El cuantificador universal de un rango vacío es } T \\
\equiv n > 0 \Rightarrow 0 \leq \frac{n}{2} \wedge T = T \\
\equiv n > 0 \Rightarrow 0 \leq \frac{n}{2} \\
T
\end{array}
$$

b) $\{P \wedge B_i\}S_i\{P\}$ para $1 \leq i \leq n$. O lo que es lo mismo $P \wedge B_i \Rightarrow wp(S_i, P)$. En este caso tenemos $P \wedge B \Rightarrow wp(\text{"}IF; S_3\text{"}, P)$ lo que nos lleva a demostrar:

1) $P \wedge B \wedge B_1 \Rightarrow wp(S_1; S_3, P) \equiv P \wedge B \wedge B_1 \Rightarrow wp(\text{"}skip; i := i+1\text{"}, P)$

$$\equiv 0 \leq i \leq \tfrac{n}{2} \wedge monte = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])$$

$$\wedge i \neq \tfrac{n}{2} \wedge b[i] \leq b[i+1] \wedge b[i] = b[n-1-i]$$

$$\Rightarrow \overbrace{0 \leq i+1 \leq \frac{n}{2}} \wedge \underbrace{monte = (\forall J \in [0 : i] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])}$$

$$\left.\begin{array}{l} P \Rightarrow i \leq \frac{n}{2} \\ B \Rightarrow i \neq \frac{n}{2} \end{array}\right\} \Rightarrow i < \tfrac{n}{2} \Rightarrow i+1 \leq \tfrac{n}{2} \qquad P \Rightarrow 0 \leq i \ \} \Rightarrow 0 \leq i+1$$

$$\left.\begin{array}{l} P \Rightarrow monte = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J]) \\ B_1 \Rightarrow b[i] \leq b[i+1] \wedge b[i] = b[n-1-i] \end{array}\right\} \Rightarrow$$

$$monte = (\forall J \in [0 : i] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])$$

2) $P \wedge B \wedge B_2 \Rightarrow wp(S_2; S_3, P) \equiv P \wedge B \wedge B_2 \Rightarrow wp(\text{"}monte := F; i := i+1\text{"}, P)$

$$\equiv 0 \leq i \leq \tfrac{n}{2} \wedge monte = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])$$

$$\wedge i \neq \tfrac{n}{2} \wedge b[i] > b[i+1] \vee b[i] \neq b[n-1-i]$$

$$\Rightarrow \overbrace{0 \leq i+1 \leq \frac{n}{2}} \wedge \underbrace{F = (\forall J \in [0 : i] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])}$$

$$\left.\begin{array}{l} P \Rightarrow i \leq \frac{n}{2} \\ B \Rightarrow i \neq \frac{n}{2} \end{array}\right\} i < \frac{n}{2} \Rightarrow i+1 \leq \frac{n}{2} \qquad P \Rightarrow 0 \leq i \;\Big\} 0 \leq i+1$$

En la parte derecha de la implicación (B_2) se pueden dar dos casos: $b[i] > b[i+1]$ o $b[i] \neq b[n-1-i]$

a) $b[i] > b[i+1]$

Separamos el cuantificador de la derecha de la forma siguiente:

$F = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J]) \wedge b[i] \leq b[i+1] \wedge b[i] = b[n-1-i]$

Como $b[i] > b[i+1]$ entonces,

$F = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J]) \wedge F \wedge b[i] = b[n-1-i]$

$F = F$ que es cierto

b) $b[i] \neq b[n-1-i]$

Igual que en el caso anterior, separamos el cuantificador de la derecha de la forma siguiente:

$F = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J]) \wedge b[i] \leq b[i+1] \wedge b[i] = b[n-1-i]$

Como $b[i] \neq b[n-1-i]$ entonces,

$F = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J]) \wedge b[i] \leq b[i+1] \wedge F$

$F = F$ que es cierto

c) $P \wedge \neg BB \Rightarrow R$

$\equiv 0 \leq i \leq \frac{n}{2} \wedge monte = (\forall J \in [0 : i-1] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J]) \wedge i = \frac{n}{2}$

$\Rightarrow monte = (\forall J \in [0 : \frac{n}{2}] : b[J] \leq b[J+1] \wedge b[J] \leq b[n-1-J])$

Que es cierto ya que si sustituimos i por $\frac{n}{2}$ tenemos directamente R

d) $P \wedge B \Rightarrow (t > 0)$
Tendríamos que demostrar que la cota es mayor que cero teniendo en cuenta las dos ramas de la estructura alternativa, es decir: $P \wedge B \wedge B_1 \Rightarrow (t > 0)$ y $P \wedge B \wedge B_2 \Rightarrow (t > 0)$. Sin embargo, como con P y B somos capaces de demostrar la parte derecha de la implicación, resumimos de la forma siguiente:

$$\left.\begin{array}{l} P \Rightarrow i \leq \frac{n}{2} \\ B \Rightarrow i \neq \frac{n}{2} \end{array}\right\} i < \frac{n}{2} \Rightarrow 0 < \frac{n}{2} - i$$

e) $\{P \wedge B_i\} t_1 := t; S_i \{t < t_1\}$ para $1 \leq i \leq n$. En este último paso tenemos que distinguir dos casos:

a) $P \wedge B \wedge B_1 \Rightarrow wp(\text{"}t_1 := t; skip; i := i+1\text{"}, t < t_1)$

$P \wedge B \wedge B_1 \Rightarrow (((\frac{n}{2} - i < t_1)^{i}_{i+1})^{skip})^{t_1}_{t}$
$P \wedge B \wedge B_1 \Rightarrow (\frac{n}{2} - (i+1) < t_1)^{t_1}_{t}$
$P \wedge B \wedge B_1 \Rightarrow (\frac{n}{2} - (i+1) < \frac{n}{2} - i)$
$P \wedge B \wedge B_1 \Rightarrow \frac{n}{2} - (i+1) < \frac{n}{2} - i$
$P \wedge B \wedge B_1 \Rightarrow -1 < 0$
Que es cierto

b) $P \wedge B \wedge B_2 \Rightarrow wp(\text{"}t_1 := t; monte := F; i := i+1\text{"}, t < t_1)$

$P \wedge B \wedge B_2 \Rightarrow (((\frac{n}{2} - i < t_1)^i_{i+1})^{monte}_F)^{t_1}_t$
$P \wedge B \wedge B_2 \Rightarrow (\frac{n}{2} - (i+1) < t_1)^{t_1}_t$
$P \wedge B \wedge B_2 \Rightarrow (\frac{n}{2} - (i+1) < \frac{n}{2} - i)$
$P \wedge B \wedge B_2 \Rightarrow \frac{n}{2} - (i+1) < \frac{n}{2} - i$
$P \wedge B \wedge B_2 \Rightarrow -1 < 0$
Que es cierto

$\boxed{7.}$ (1 punto) Se dice que un array de enteros presenta **predominancia positiva** cuando el número de sus componentes estrictamente positivos es estrictamente superior al número de sus componentes estrictamente negativos. Dado un array $b[0:n-1] : integer$ se desea saber si dicho array tiene predominancia positiva. Establecer la precondición y la postcondición, la invariante, el programa y demostrar la corrección *parcial* (**equivale a la práctica optativa**).

7.1 Establecer una precondición y una postcondición adecuadas

$$\{Q: \ b[0:n-1] : integer \wedge n \geq 0 \wedge b = B\}$$
$$\{R: \ s = (\mathbf{N}J \in [0:n-1] : b[J] > 0) - (\mathbf{N}J \in [0:n-1] : b[J] < 0)\}$$

Escribiendo la postcondición de esta forma, diremos que el vector tiene predominancia positiva si s es mayor que cero y no la tiene en cualquier otro caso.

7.2 Fijar una invariante

Una de las maneras de resolver el problema pasa por debilitar R cambiando la constante n por una variable i. De esta forma, la invariante queda como sigue:

$$\{P: \ 0 \leq i \leq n \wedge s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0)\}$$

7.3 Escribir el programa

$$
\begin{array}{l}
\{Q\} \\
S_0 : i, s := 0, 0; \\
\text{do } B : i \neq n \rightarrow \ \text{if } B_1 : b[i] > 0 \rightarrow S_1 : s := s + 1; \\
\qquad\qquad\qquad\ \ \| \ B_2 : b[i] < 0 \rightarrow S_2 : s := s - 1; \\
\qquad\qquad\qquad\ \ \| \ B_3 : b[i] = 0 \rightarrow S_3 : skip; \\
\qquad\qquad\qquad\ \ \text{fi} \\
\qquad\qquad\qquad\ \ S_4 : i := i + 1; \\
\text{od} \\
\{R\}
\end{array}
$$

7.4 Demostrar su corrección *parcial*

a) $Q \Rightarrow wp(S_0, P)$

$b[0:n-1] : integer \wedge n \geq 0 \wedge b = B \Rightarrow wp(\text{"}i, s := 0, 0\text{"}, P)$

$\equiv n \geq 0 \Rightarrow 0 \leq 0 \leq n \wedge 0 = (\mathbf{N}J \in [0:-1] : b[J] > 0) - (\mathbf{N}J \in [0:-1] : b[J] < 0)$

El cuantificador numérico de un rango vacío es 0

$\equiv n \geq 0 \Rightarrow 0 \leq n \wedge 0 = 0$

$\equiv T$

b) $\{P \wedge B_i\} S_i \{P\}$ para $1 \leq i \leq n$. O lo que es lo mismo $P \wedge B_i \Rightarrow wp(S_i, P)$ En este caso tenemos $P \wedge B \Rightarrow wp(\text{"}IF; S_4\text{"}, P)$ lo que nos lleva a demostrar:

1) $P \wedge B \wedge B_1 \Rightarrow wp(\text{"}S_1; S_4, P) \equiv P \wedge B \wedge B_1 \Rightarrow wp(\text{"}s := s+1; i := i+1\text{"}, P)$

$$\equiv 0 \le i \le n \wedge s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) \wedge i \ne n \wedge b[i] > 0$$

$$\Rightarrow \overbrace{0 \le i+1 \le n} \wedge \underbrace{s+1 = (\mathbf{N}J \in [0:i] : b[J] > 0) - (\mathbf{N}J \in [0:i] : b[J] < 0)}$$

$$\left.\begin{array}{l} P \Rightarrow 0 \le i \le n \\ B \Rightarrow i \ne n \end{array}\right\} 0 \le i < n \Rightarrow 0 \le i+1 \le n$$

La segunda expresión de la conjunción puede escribirse de la forma siguiente:

$s+1 = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) + \alpha(i)$

en donde $\alpha(i)$ puede tomar los siguientes valores:

$$\alpha(i) \left\{ \begin{array}{l} 1 \text{ si } b[i] > 0 \\ 0 \text{ si } b[i] = 0 \\ -1 \text{ si } b[i] < 0 \end{array}\right.$$

$$\left.\begin{array}{l} P \Rightarrow s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) \\ B_1 \Rightarrow b[i] > 0 \Rightarrow \alpha(i) = 1 \end{array}\right\} s+1 = s+1$$

2) $P \wedge B \wedge B_2 \Rightarrow wp(\text{"}S_2; S_4, P) \equiv P \wedge B \wedge B_2 \Rightarrow wp(\text{"}s := s-1; i := i+1\text{"}, P)$

$$\equiv 0 \le i \le n \wedge s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) \wedge i \ne n \wedge b[i] < 0$$

$$\Rightarrow \overbrace{0 \le i+1 \le n} \wedge \underbrace{s-1 = (\mathbf{N}J \in [0:i] : b[J] > 0) - (\mathbf{N}J \in [0:i] : b[J] < 0)}$$

$$\left.\begin{array}{l} P \Rightarrow 0 \le i \le n \\ B \Rightarrow i \ne n \end{array}\right\} 0 \le i < n \Rightarrow 0 \le i+1 \le n$$

Utilizando $\alpha(i)$ como en el apartado anterior,

$s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) + \alpha(i)$

$$\left.\begin{array}{l} P \Rightarrow s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) \\ B_2 \Rightarrow b[i] < 0 \Rightarrow \alpha(i) = -1 \end{array}\right\} s-1 = s-1$$

3) $P \wedge B \wedge B_3 \Rightarrow wp(\text{"}S_3; S_4, P) \equiv P \wedge B \wedge B_3 \Rightarrow wp(\text{"}skip; i := i+1\text{"}, P)$

$$\equiv 0 \le i \le n \wedge s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) \wedge i \ne n \wedge b[i] = 0$$

$$\Rightarrow \overbrace{0 \le i+1 \le n} \wedge \underbrace{s = (\mathbf{N}J \in [0:i] : b[J] > 0) - (\mathbf{N}J \in [0:i] : b[J] < 0)}$$

$$\left.\begin{array}{l} P \Rightarrow 0 \le i \le n \\ B \Rightarrow i \ne n \end{array}\right\} 0 \le i < n \Rightarrow 0 \le i+1 \le n$$

Utilizando $\alpha(i)$ como en el apartado anterior,

$s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) + \alpha(i)$

$$\left.\begin{array}{l} P \Rightarrow s = (\mathbf{N}J \in [0:i-1] : b[J] > 0) - (\mathbf{N}J \in [0:i-1] : b[J] < 0) \\ B_3 \Rightarrow b[i] = 0 \Rightarrow \alpha(i) = 0 \end{array}\right\} s = s$$

4) $P \wedge \neg BB \Rightarrow R$

$\equiv 0 \leq i \leq n \wedge s = (\mathbf{N}J \in [0 : i-1] : b[J] > 0) - (\mathbf{N}J \in [0 : i-1] : b[J] < 0) \wedge \neg(i \neq n)$

$\Rightarrow s = (\mathbf{N}J \in [0 : n-1] : b[J] > 0) - (\mathbf{N}J \in [0 : n-1] : b[J] < 0)$

$\equiv 0 \leq i \leq n \wedge s = (\mathbf{N}J \in [0 : i-1] : b[J] > 0) - (\mathbf{N}J \in [0 : i-1] : b[J] < 0) \wedge i = n$

$\Rightarrow s = (\mathbf{N}J \in [0 : n-1] : b[J] > 0) - (\mathbf{N}J \in [0 : n-1] : b[J] < 0)$

Que es cierto ya que si sustituimos i por n tenemos directamente R