Apellidos, Nombre:. . . . . **GRUPO:**. . . . .

Tecnología de la Programación. ITIS. Curso 2007/2008

Grupos Cooperativos. Seminario III. 18-21 de Enero de 2008

---

1. Escribir un programa que determine si un array $b[0:n-1]: integer$ con $n \geq 0$ tiene todos sus elementos a cero.

   a) Establecer una *precondición* y una *postcondición* adecuadas para el programa.

   $$\{Q:\ n \geq 0 \wedge b[0:n-1]: integer \wedge b = B\}$$
   $$\{R:\ z = (\forall J: 0 \leq J < n: b[J] = 0) \wedge b = B\}$$

   b) Establecer la *invariante* y *función cota* para el programa.

   $$\{P:\ (0 \leq i \leq n) \wedge z = (\forall J: 0 \leq J < i: b[J] = 0) \wedge b = B\}$$
   $$\{t:\ n - i\}$$

   c) Desarrollar el *programa* y *anotarlo*.

   $\{Q\}$
   $S_0: i, z := 0, T;$
   $\{P\}$
   $\{t\}$
   do $B_1: i \neq n$ cand $b[i] = 0 \rightarrow S_1: i := i + 1;$
   $\mid B_2: i \neq n$ cand $b[i] \neq 0 \rightarrow S_2: z, i := F, i + 1;$
   od
   $\{P \wedge \neg BB\}$
   $\{R\}$

   d) Realizar la prueba de *corrección total*.

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

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

   $\equiv n \geq 0 \wedge b = B \Rightarrow 0 \leq 0 \leq n \wedge T = (\forall J: 0 \leq J < 0: b[J] = 0) \wedge b = B$

   El cuantificador universal de un rango vacío es T.

   $\equiv n \geq 0 \wedge b = B \Rightarrow 0 \leq n \wedge T = T \wedge b = B$

   $\equiv n \geq 0 \wedge b = B \Rightarrow 0 \leq n \wedge b = B$ que es cierto.

   2) $\{P \wedge B_i\}S_i\{P\}$ para $1 \leq i \leq n$. O lo que es lo mismo $P \wedge B_i \Rightarrow wp(S_i, P)$ lo que nos lleva a demostrar:

   2.1 $P \wedge B_1 \Rightarrow wp(S_1, P) \equiv P \wedge B_1 \Rightarrow wp(\text{"}i := i + 1;\text{"}, P)$

$\equiv 0 \leq i \leq n \wedge z = (\forall J : 0 \leq J < i : b[J] = 0) \wedge b = B \wedge i \neq n \text{ cand } b[i] = 0$

$\Rightarrow 0 \leq i+1 \leq n \wedge \overbrace{z = (\forall J : 0 \leq J < i+1 : b[J] = 0)}^{\alpha} \wedge b = B$

$b = B$ es cierto por que también lo es en P.
La primera expresión de la conjunción se obtiene directamente de P y B_1:

$$\left.\begin{array}{l} P \Rightarrow 0 \leq i \leq n \\ B_1 \Rightarrow i \neq n \end{array}\right\} 0 \leq i < n \Rightarrow 0 \leq i+1 \leq n$$

En α se aísla el caso $J = i$ de la forma: $z = (\forall J : 0 \leq J < i : b[J] = 0) \wedge b[i] = 0$
predicado que es cierto por P y B_1.

2.2 $P \wedge B_2 \Rightarrow wp(S_2, P) \equiv P \wedge B_2 \Rightarrow wp(\text{"}z, i := F, i+1\text{"}, P)$

$\equiv 0 \leq i \leq n \wedge z = (\forall J : 0 \leq J < i : b[J] = 0) \wedge b = B \wedge i \neq n \text{ cand } b[i] \neq 0$

$\Rightarrow 0 \leq i+1 \leq n \wedge \overbrace{F = (\forall J : 0 \leq J < i+1 : b[J] = 0)}^{\alpha} \wedge b = B$

$b = B$ es cierto por que también lo es en P.
La primera expresión de la conjunción se obtiene directamente de P y B_2:

$$\left.\begin{array}{l} P \Rightarrow 0 \leq i \leq n \\ B_1 \Rightarrow i \neq n \end{array}\right\} 0 \leq i < n \Rightarrow 0 \leq i+1 \leq n$$

En α se aísla el caso $J = i$ de la forma: $F = (\forall J : 0 \leq J < i : b[J] = 0) \wedge b[i] = 0$

$b[i] = 0$ es falso por B_2, por lo tanto: $F = (\forall J : 0 \leq J < i : b[J] = 0) \wedge F \equiv F = F$ que es cierto.

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

Primero calculamos $\neg BB \equiv \neg((i \neq n \text{ cand } b[i] = 0) \vee (i \neq n \text{ cand } b[i] \neq 0))$

$\equiv \neg((i \neq n \text{ cand } (b[i] = 0 \vee b[i] \neq 0)) \equiv \neg(i \neq n) \equiv (i = n)$

En la expresión de partida ($P \wedge \neg BB \Rightarrow R$) tenemos entonces:

$0 \leq i \leq n \wedge z = (\forall J : 0 \leq J < i : b[J] = 0) \wedge b = B \wedge i = n \Rightarrow z = (\forall J : 0 \leq J < n : b[J] = 0) \wedge b = B$

que es cierto por que P se ha obtenido a partir de R cambiando n por i.

4) $P \wedge B \Rightarrow (t > 0)$
Tendríamos que demostrar que la cota es mayor que cero teniendo en cuenta las dos ramas de la estructura iterativa, es decir: $P \wedge B_1 \Rightarrow (t > 0)$ y $P \wedge B_2 \Rightarrow (t > 0)$.

a) $P \wedge B_1 \Rightarrow (n - i) > 0$

$P \wedge (i \neq n \text{ cand } b[i] = 0) \Rightarrow (n - i) > 0$
$0 \leq i \leq n \wedge i \neq n \Rightarrow (n - i) > 0$
$i < n \Rightarrow n > i$
Que es cierto.

b) $P \wedge B_2 \Rightarrow (n-i) > 0$

$P \wedge (i \neq n \text{ cand } b[i] \neq 0) \Rightarrow (n-i) > 0$
$0 \leq i \leq n \wedge i \neq n \Rightarrow (n-i) > 0$
$i < n \Rightarrow n > i$
Que es cierto.

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

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

$P \wedge B_1 \Rightarrow ((n-i < t_1)^{i}_{i+1}))^{t_1}_{n-i}$
$P \wedge B_1 \Rightarrow (n-(i+1) < t_1)^{t_1}_{n-i}$
$P \wedge B_1 \Rightarrow n-(i+1) < n-i$
$P \wedge B_1 \Rightarrow -1 < 0$
Que es cierto.

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

$P \wedge B_2 \Rightarrow ((n-i < t_1)^{z,i}_{F,i+1})^{t_1}_{n-i}$
$P \wedge B_2 \Rightarrow (n-(i+1) < t_1)^{t_1}_{n-i}$
$P \wedge B_2 \Rightarrow n-(i+1) < n-i$
$P \wedge B_2 \Rightarrow -1 < 0$
Que es cierto.

2. Escribir un programa que dado un array de valores enteros $b[0:n-1]$ con $n > 0$, deje en una variable r el valor $'s'$ si el array es simétrico o el valor $'n'$ en cualquier otro caso.

a) Establecer una *precondición* y una *postcondición* adecuadas para el programa.

$\{Q: \ n > 0 \wedge b[0:n-1] : integer \wedge b = B\}$
$\{R: \ ((r =' s' \wedge (\forall J : 0 \leq J < n/2 : b[J] = b[n-1-J])) \vee (r =' n' \wedge (\exists J : 0 \leq J < n/2 : b[J] \neq b[n-1-J])))$
$\wedge\, b = B\}$

La división utilizada en la operación $n/2$ es la división entera.

b) Establecer la *invariante* y *función cota* para el programa.

$\{P: \ (0 \leq i \leq n/2) \wedge ((r =' s' \wedge (\forall J : 0 \leq J < i : b[J] = b[n-1-J]))$
$\vee (r =' n' \wedge (\exists J : 0 \leq J < i : b[J] \neq b[n-1-J]))) \wedge b = B\}$
$\{t: \ n/2 - i\}$

c) Desarrollar el *programa* y *anotarlo*.

$\{Q\}$
$S_0 : i, r := 0,' s';$
$\{P\}$
$\{t\}$
do $B : i \neq n/2 \rightarrow \{P \wedge B\}$ if $B_1 : b[i] = b[n-1-i] \rightarrow \{P \wedge B \wedge B1\} S_1 : i := i+1; \{P\}$
$\mid B_2 : b[i] \neq b[n-1-i] \rightarrow S_2 : i, r := i+1,' n'; \{P\}$
fi
od
$\{P \wedge \neg B\}$
$\{R\}$

d) Realizar la prueba de *corrección total.*

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

$n > 0 \wedge b[0:n-1] : integer \wedge b = B \Rightarrow wp("i, r := 0, s", P)$

$\equiv n > 0 \wedge b = B \Rightarrow \ (0 \leq 0 \leq n/2) \wedge (('s' =' s' \wedge (\forall J : 0 \leq J < 0 : b[J] = b[n-1-J]))$
$\vee('s' =' n' \wedge (\exists J : 0 \leq J < 0 : b[J] \neq b[n-1-J]))) \wedge b = B$

El cuantificador universal de un rango vacío es T y el cuantificador existencial de un rango vacío es F.

$\equiv n > 0 \wedge b = B \Rightarrow 0 \leq n/2 \wedge (T \vee F) \wedge b = B$

$\equiv n > 0 \wedge b = B \Rightarrow 0 \leq n/2 \wedge b = B$ que es cierto

2) $\{P \wedge B\}IF\{P\}$. Demostrar este paso equivale a demostrar la corrección de la estructura alternativa.

$\{Q^{'} \equiv P \wedge B\}$
if $B_1 : b[i] = b[n-1-i] \rightarrow S_1 : i := i+1;$
$\mid B_2 : b[i] \neq b[n-1-i] \rightarrow S_2 : i, r := i+1,' n';$
fi
$\{R^{'} \equiv P\}$

2.1 $P \wedge B \Rightarrow BB$

$P \wedge B \Rightarrow (b[i] = b[n-1-i] \vee b[i] \neq b[n-1-i])$

$P \wedge B \Rightarrow T$, que es cierto.

2.2 $P \wedge B \wedge B_1 \Rightarrow wp(S_1, P) \equiv P \wedge B \wedge B_1 \Rightarrow wp("i := i+1;", P)$

$$(0 \leq i \leq n/2) \wedge (\overbrace{(r =' s' \wedge (\forall J : 0 \leq J < i : b[J] = b[n-1-J]))}^{\alpha}$$
$$\vee \underbrace{(r =' n' \wedge (\exists J : 0 \leq J < i : b[J] \neq b[n-1-J]))}_{\beta}) \wedge b = B \wedge i \neq n/2 \wedge b[i] = b[n-1-i] \Rightarrow$$
$$(0 \leq i+1 \leq n/2) \wedge ((r =' s' \wedge (\forall J : 0 \leq J < i+1 : b[J] = b[n-1-J]))$$
$$\vee(r =' n' \wedge (\exists J : 0 \leq J < i+1 : b[J] \neq b[n-1-J]))) \wedge b = B$$

$b = B$ es cierto por que también lo es en P.
La primera expresión de la conjunción se obtiene directamente de P y B:

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

En P se pueden dar dos situaciones: que α sea cierto o que β sea cierto.

(1) Si α es cierto entonces por P y B_1 en la parte derecha de la implicación es cierto que:

$(r =' s' \wedge (\forall J : 0 \leq J < i : b[J] = b[n-1-J]) \wedge b[i] = b[n-1-i])$ (hemos separado el caso $J = i$)

y por ello toda la implicación es cierta.

(2) Si β es cierto entonces por P en la parte derecha de la implicación es cierto que:

$(s =' n' \wedge ((\exists J : 0 \leq J < i : b[J] \neq b[n-1-J]) \vee (b[i] \neq b[n-1-i])))$ (hemos separado el caso $J = i$)

por que es cierta una de las dos partes del $\vee$. Por ello toda la implicación es cierta.

$$2.3\;\; P \wedge B \wedge B_2 \Rightarrow wp(S_2, P) \equiv P \wedge B \wedge B_2 \Rightarrow wp(''i, r := i+1, 'n''', P)$$

$$\begin{array}{l}
(0 \leq i \leq n/2) \wedge (\overbrace{(r =' s' \wedge (\forall J : 0 \leq J < i : b[J] = b[n-1-J]))}^{\alpha} \\
\vee \underbrace{(r =' n' \wedge (\exists J : 0 \leq J < i : b[J] \neq b[n-1-J])))}_{\beta} \wedge b = B \wedge i \neq n/2 \wedge b[i] \neq b[n-1-i] \Rightarrow \\
(0 \leq i+1 \leq n/2) \wedge (('n' =' s' \wedge (\forall J : 0 \leq J < i+1 : b[J] = b[n-1-J])) \\
\vee('n' =' n' \wedge (\exists J : 0 \leq J < i+1 : b[J] \neq b[n-1-J]))) \wedge b = B
\end{array}$$

$b = B$ es cierto por que también lo es en P.
La primera expresión de la conjunción se obtiene directamente de P y B:

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

En P se pueden dar dos situaciones: que α sea cierto o que β sea cierto.

(1) α es cierto. Sin embargo, por B_2 en la parte derecha de la implicación es cierto que:

$('n' =' n' \wedge ((\exists J : 0 \leq J < i : b[J] \neq b[n-1-J]) \vee (b[i] \neq b[n-1-i])))$ (hemos separado el caso $J = i$)

y por ello toda la implicación es cierta.

(2) Si β es cierto entonces por P en la parte derecha de la implicación es cierto que:

$('n' =' n' \wedge ((\exists J : 0 \leq J < i : b[J] \neq b[n-1-J]) \vee (b[i] \neq b[n-1-i])))$ (hemos separado el caso $J = i$)

Por ello toda la implicación es cierta.

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

$$\begin{array}{l}
(0 \leq i \leq n/2) \wedge ((r =' s' \wedge (\forall J : 0 \leq J < i : b[J] = b[n-1-J])) \\
\vee(r =' n' \wedge (\exists J : 0 \leq J < i : b[J] \neq b[n-1-J]))) \wedge b = B \wedge i = n/2 \Rightarrow \\
((r =' s' \wedge (\forall J : 0 \leq J < n/2 : b[J] = b[n-1-J])) \\
\vee(r =' n' \wedge (\exists J : 0 \leq J < n/2 : b[J] \neq b[n-1-J]))) \wedge b = B
\end{array}$$

que es cierto por que P se ha obtenido a partir de R cambiando $n/2$ por i.

4) $P \wedge B \Rightarrow (t > 0)$

Tendríamos que demostrar que la cota es mayor que cero teniendo en cuenta las dos ramas de la estructura alternativa. Sin embargo, como ambas ramas modifican de la misma forma la función cota llega con demostrar que $P \wedge B \Rightarrow (n/2 - i > 0)$.

$P \wedge i \neq n/2 \Rightarrow (n/2 - i) > 0$
$0 \leq i \leq n/2 \wedge i \neq n/2 \Rightarrow (n/2 - i) > 0$
$i < n/2 \Rightarrow n/2 > i$
Que es cierto.

5) $\{P \wedge B \wedge B_i\}t_1 := t; S_i\{t < t_1\}$ para $1 \leq i \leq n$. Al igual que en el apartado anterior, en este último paso tendríamos que distinguir dos casos, pero como ambas ramas de la estructura alternativa modifican de la misma forma la función cota, llega con demostrar:

$P \wedge B \Rightarrow ((n/2 - i < t_1)^i_{i+1}))^{t_1}_{n/2-i}$
$P \wedge B \Rightarrow (n/2 - (i+1) < t_1)^{t_1}_{n/2-i}$
$P \wedge B \Rightarrow n/2 - (i+1) < n/2 - i$
$P \wedge B \Rightarrow -1 < 0$
Que es cierto.

3. Escribir un programa que invierta un array de enteros $b[0 : n-1]$ con $n > 0$.

   a) Establecer una *precondición* y una *postcondición* adecuadas para el programa.

   $\{Q :\ n > 0 \wedge b[0 : n-1] : integer \wedge b = B\}$
   $\{R :\ (\forall K : 0 \leq K < n/2 : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])\}$

   b) Establecer la *invariante* y *función cota* para el programa.

$\{P :\ (0 \leq i \leq n/2) \wedge (\forall K : 0 \leq K < i : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$
$\wedge (\forall J : i \leq J \leq (n-1) - i : b[J] = B[J])\}$
$\{t :\ n/2 - i\}$

   c) Desarrollar el *programa* y *anotarlo.*

$\{Q\}$
$S_0 : i := 0;$
$\{P\}$
$\{t\}$
do $B : i \neq n/2 \rightarrow S_1 : b[i], b[n-1-i], i := b[n-1-i], b[i], i+1;$
od
$\{P \wedge \neg B\}$
$\{R\}$

   d) Realizar la prueba de *corrección total.*

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

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

$\equiv n > 0 \wedge b = B \Rightarrow (0 \leq 0 \leq n/2) \wedge (\forall K : 0 \leq K < 0 : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$
$\wedge(\forall J : 0 \leq J \leq n-1 : b[J] = B[J])$

El cuantificador universal de un rango vacío es T.

$\equiv n > 0 \wedge b = B \Rightarrow 0 \leq n/2 \wedge T \wedge b = B$ que es cierto.

b) $\{P \wedge B\}S_1\{P\}$, o lo que es lo mismo $P \wedge B \Rightarrow wp(S_1, P)$:

Calculamos primero el wp aislando el caso $K = i$ para poder aplicar S_1:

$$wp(S_1, P) = (0 \le i+1 \le n/2) \wedge (\forall K : 0 \le K < i : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$$
$$\wedge (b[i] = B[n-1-i] \wedge b[n-1-ii] = B[i])^{b[i], b[n-1-i]}_{b[n-1-i], b[i]}$$
$$\wedge (\forall J : i+1 \le J \le (n-1)-(i+1) : b[J] = B[J])$$

$$wp(S_1, P) = (0 \le i+1 \le n/2) \wedge (\forall K : 0 \le K < i : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$$
$$\wedge b[n-1-i] = B[n-1-i] \wedge b[i] = B[i]$$
$$\wedge (\forall J : i+1 \le J \le (n-1)-(i+1) : b[J] = B[J])$$

Y ahora sustituimos en la expresión de partida, $P \wedge B \Rightarrow wp(S_1, P)$:

$$\equiv (0 \le i \le n/2) \wedge (\forall K : 0 \le K < i : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$$
$$\wedge (\forall J : i \le J \le (n-1)-i : b[J] = B[J]) \wedge i \ne n/2 \Rightarrow$$
$$(0 \le i+1 \le n/2) \wedge \overbrace{(\forall K : 0 \le K < i : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])}^{\alpha}$$
$$\wedge \underbrace{b[n-1-i] = B[n-1-i] \wedge b[i] = B[i]}_{\beta} \wedge \overbrace{(\forall J : i+1 \le J \le (n-1)-(i+1) : b[J] = B[J])}^{\gamma}$$

La primera expresión de la conjunción se obtiene directamente de P y B:

$$\left.\begin{array}{l} P \Rightarrow 0 \le i \le n/2 \\ B \Rightarrow i \ne n/2 \end{array}\right\} 0 \le i < n/2 \Rightarrow 0 \le i+1 \le n/2$$

En la parte derecha de la implicación α es cierto por P y γ también es cierto por P (es un $\forall$ con un rango menor que el que aparece en P). β es cierto también por P ya que el elemento $J = i$ aparece en el predicado de P: $(\forall J : i \le J \le (n-1)-i : b[J] = B[J])$

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

$$\equiv (0 \le i \le n/2) \wedge (\forall K : 0 \le K < i : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$$
$$\wedge (\forall J : i \le J \le (n-1)-i : b[J] = B[J]) \wedge i = n/2 \Rightarrow$$
$$(\forall K : 0 \le K < n/2 : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$$

$$\equiv (\forall K : 0 \le K < n/2 : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$$
$$\wedge \overbrace{(\forall J : n/2 \le J \le (n-1)-n/2 : b[J] = B[J])}^{\alpha} \Rightarrow$$
$$(\forall K : 0 \le K < n/2 : b[K] = B[n-1-K] \wedge b[n-1-K] = B[K])$$

que es cierto ya que α es un cuantificador con rango vacío por lo que en P tenemos todos los elementos intercambiados.

d) $P \wedge B \Rightarrow (t > 0)$

$$P \wedge i \ne n/2 \Rightarrow (n/2 - i) > 0$$
$$0 \le i \le n/2 \wedge i \ne n/2 \Rightarrow (n/2 - i) > 0$$
$$i < n/2 \Rightarrow n/2 > i$$
Que es cierto.

e) $\{P \wedge B \wedge B_i\} t_1 := t; S_i \{t < t_1\}$ para $1 \leq i \leq n$.

$$P \wedge B \Rightarrow ((n/2 - i < t_1)^i_{i+1}))^{t_1}_{n/2-i}$$
$$P \wedge B \Rightarrow (n/2 - (i+1) < t_1)^{t_1}_{n/2-i}$$
$$P \wedge B \Rightarrow n/2 - (i+1) < n/2 - i$$
$$P \wedge B \Rightarrow -1 < 0$$

Que es cierto.

4. Escribir un programa que deje en la variable *max* el mayor valor absoluto de un array $b[0 : n-1] : integer$ con $n > 0$.

a) Establecer una *precondición* y una *postcondición* adecuadas para el programa.

$\{Q: \ n > 0 \wedge b[0 : n-1] : integer \wedge b = B\}$
$\{R: \ (\exists K : 0 \leq K < n : max = |b[K]|) \wedge (\forall J : 0 \leq J < n : max \geq |b[J]|) \wedge b = B\}$

b) Establecer la *invariante* y *función cota* para el programa.

$\{P: \ (0 \leq i \leq n) \wedge (\exists K : 0 \leq K < i : max = |b[K]|) \wedge (\forall J : 0 \leq J < i : max \geq |b[J]|) \wedge b = B\}$

$\{t: \ n - i\}$

c) Desarrollar el *programa* y *anotarlo*.

$\{Q\}$
$S_0 : i, r := 1, |b[0]|;$
$\{P\}$
$\{t\}$
do $B : i \neq n \rightarrow \{P \wedge B\}$ if $B_1 : |b[i]| > max \rightarrow \{P \wedge B \wedge B1\} S_1 : max := |b[i]|; \{P\}$
$\quad | \ B_2 : max \leq |b[i]| \rightarrow S_2 : skip; \{P\}$
$\quad$ fi
$\quad S_3 : i := i + 1;$
od
$\{P \wedge \neg B\}$
$\{R\}$

d) Realizar la prueba de *corrección total*.

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

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

$\equiv n > 0 \wedge b = B \Rightarrow \ (\underbrace{0 \leq 1}_{T} \leq n) \wedge (\exists K : 0 \leq K < 1 : |b[0]| = |b[K]|) \wedge (\forall J : 0 \leq J < 1 : |b[0]| \geq |b[J]|) \wedge b = B$

$\equiv n > 0 \wedge b = B \Rightarrow 1 \leq n \wedge (|b[0]| = |b[0]| \wedge |b[0]| = |b[0]|) \wedge b = B$

$\equiv n > 0 \wedge b = B \Rightarrow 1 \leq n \wedge b = B$ que es cierto.

2) $\{P \wedge B\} IF \{P'\} i := i+1; \{P\}$. Demostrar este paso equivale a demostrar la corrección de la estructura alternativa sobre $\{P'\}$ que se obtiene a través de $wp(\text{"}i := i+1\text{"}, P)$.

$\{Q^{'} \equiv P \wedge B\}$
if $B_1 : |b[i]| > max \rightarrow S_1 : max := |b[i]|$;
$| \; B_2 : |b[i]| \leq |b[i]| \rightarrow S_2 : skip$;
fi
$\{R^{'} \equiv P' \equiv wp(\text{"}i := i+1\text{"}, P)\}$

$P' \equiv (0 \leq i+1 \leq n \wedge (\exists K : 0 \leq K < i+1 : max = |b[K]|) \wedge (\forall J : 0 \leq J < i+1 : max \geq |b[J]|))$

2.1 $P \wedge B \Rightarrow dominio(BB)$

$P \wedge i \neq n \Rightarrow dominio(|b[i]|)$

$P \wedge i \neq n \Rightarrow enrango(b, i)$ cand $dominio(|b[i]|)$

$P : 0 \leq i \leq n \wedge B : i \neq n \Rightarrow enrango(b, i)$

$P : b = B \Rightarrow dominio(|b[i]|)$

Y por lo tanto la implicación inicial es cierta.

2.2 $P \wedge B \Rightarrow BB$

$P \wedge i \neq n \Rightarrow (|b[i]| > max \vee |b[i]| \leq max)$

$P \wedge i \neq n \Rightarrow T$, que es cierto.

2.3 $P \wedge B \wedge B_1 \Rightarrow wp(S_1, P) \equiv P \wedge B \wedge B_1 \Rightarrow wp(\text{"}max := |b[i]|\text{"}, P)$

Calculamos el wp

$$\underbrace{(0 \leq i+1 \leq n)}_{\alpha} \wedge (\overbrace{(\exists K : 0 \leq K < i+1 : |b[i]| = |b[K]|)}^{\beta} \wedge \underbrace{(\forall J : 0 \leq J < i+1 : |b[i]| \geq |b[J]|)}_{\delta} \wedge b = B$$

$b = B$ es cierto por que también lo es en P. α se obtiene directamente de P y B:

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

β es cierto ya que el cuantificador existencial es cierto para $K = i$.

En δ aislamos primero el caso $K = i$:

$$\delta \equiv (\forall J : 0 \leq J < i : |b[i]| \geq |b[J]|) \wedge \overbrace{(|b[i]| \geq |b[i]|)}^{T}$$

a primera parte de la conjunción es cierta por P y la segunda lo es directamente.

Por ello toda la implicación es cierta.

2.4 $P \wedge B \wedge B_2 \Rightarrow wp(S_2, P) \equiv P \wedge B \wedge B_2 \Rightarrow wp(skip, P)$

$$P \wedge i \neq n \wedge |b[i]| \leq max \Rightarrow (0 \leq i+1 \leq n \wedge \underbrace{(\exists K : 0 \leq K < i+1 : max = |b[K]|)}_{\alpha}$$

$$\wedge \underbrace{(\forall J : 0 \leq J < i+1 : max \geq |b[J]|)}_{\beta})$$

$b = B$ es cierto por que también lo es en P.
La primera expresión de la conjunción se obtiene directamente de P y B:

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

En α separamos el caso $K = i$ en el cuantificador existencial.

$\alpha \equiv (\exists K : 0 \leq K < i : max = |b[K]|) \vee max = |b[i]|$

La primera parte de la disyunción es cierta por P y por lo tanto α es cierto.

En β separamos también el caso $J = i$.

$$\beta \equiv \overbrace{(\forall J : 0 \leq J < i : max \geq |b[J]|)}^{T \text{ por } P} \wedge \overbrace{max \geq |b[i]|}^{T \text{ por } B_2}$$

Por ello β es cierto y toda la implicación es cierta.

3) $P \wedge \neg B \Rightarrow R$

$P \wedge i = n \Rightarrow R$

que es cierto por que P se ha obtenido a partir de R cambiando n por i.

4) $P \wedge B \Rightarrow (t > 0)$

$P \wedge i \neq n \Rightarrow (n-i) > 0$
$0 \leq i \leq n \wedge i \neq n \Rightarrow (n-i) > 0$
$i < n \Rightarrow n > i$
Que es cierto.

5) $\{P \wedge B \wedge B_i\} t_1 := t; S_i; S3; \{t < t_1\}$ para $1 \leq i \leq 2$. En este último paso tendríamos que distinguir dos casos, pero como ninguna de las dos ramas de la estructura alternativa modifica la función cota, llega con demostrar :

$P \wedge B \Rightarrow wp(\text{"}t_1 := n-i; i := i+1\text{"}, n-i < t_1)$
$P \wedge B \Rightarrow (n-i < t_1)^{i}_{i+1}))^{t_1}_{n-i}$
$P \wedge B \Rightarrow (n-(i+1) < t_1)^{t_1}_{n-i}$
$P \wedge B \Rightarrow n-(i+1) < n-i$
$P \wedge B \Rightarrow -1 < 0$
Que es cierto.