DEPARTAMENTO DE COMPUTACIÓN
UNIVERSIDADE DA CORUÑA

Tecnología de la programación

Ingeniería Técnica en Informática de Sistemas. Curso 2005/06

Convocatoria Ordinaria de Febrero (13 de Febrero de 2006)

Nombre : ....

Apellidos : ....

D.N.I. : ....

Instrucciones: En las preguntas 1 a 4 marcar todas las respuestas correctas. Cada pregunta correcta se valora con +0,5 puntos, incorrecta con -0,25, y no contestada con 0. Una puntuación total negativa en las preguntas 1 a 4 es valorada como 0 de cara al resto del examen.

1. El predicado $(\forall I : 0 \leq I < n : (\exists J : 0 \leq J < n : b[J] = b[I] \Leftrightarrow b[I] > 0))$

(a) es equivalente a $(\forall I : 0 \leq I < n : (\exists J : 0 \leq J < n : (b[J] \neq b[I] \vee b[I] > 0) \wedge (b[I] \leq 0 \vee b[J] = b[I])))$

(b) significa que todo elemento de b tiene otro igual en el array si dicho elemento es positivo

(c) es equivalente a $(\exists J : 0 \leq J < n : (\forall I : 0 \leq I < n : b[I] > 0 \wedge b[J] = b[I]))$

(d) es equivalente a $(\forall J : 0 \leq J < n : (\exists I : 0 \leq I < n : (b[J] = b[I] \Rightarrow b[J] > 0) \wedge (b[J] > 0 \Rightarrow b[J] = b[I])))$

(a) **Sí**, este predicado resulta de la aplicación de la ley de la implicación sobre ambos lados de la doble implicación .

(b) **No**, el predicado del enunciado dice *si y sólo sí*.

(c) **No**, este predicado dice que todos los elementos positivos son iguales y no dice nada sobre los elementos negativos. El predicado del enunciado dice que para todos los elementos de b si el elemento es positivo entonces tiene al menos otro igual a él, pero si el elemento es negativo no existe otro elemento igual a ese elemento negativo.

(d) **Sí**, este predicado resulta de la transformación de la doble implicación en dos implicaciones simples.

2. Dada la expresión $\alpha : 0 \leq h \leq k \leq n \wedge (\forall i \in [0, h-1] : b[i] \neq 0) \wedge (\forall i \in [k+1, n] : b[i] \neq 0)$ se cumple que:

(a) la variable k aparece ligada en α

(b) las variables h y k aparecen libres en α

(c) la sustitución α^i_j es igual a $0 \leq h \leq k \leq n \wedge (\forall j \in [0, h-1] : b[j] \neq 0) \wedge (\forall j \in [k+1, n] : b[j] \neq 0)$

(d) la sustitución α^k_m es igual a $0 \leq h \leq m \leq n \wedge (\forall j \in [0, h-1] : b[j] \neq 0) \wedge (\forall j \in [m+1, n] : b[j] \neq 0)$

(a) **No**. Aparece libre en $0 \leq h \leq k \leq n$ y en el rango de la variable i.

(b) **Sí**. Aparecen libres en $0 \leq h \leq k \leq n$ y en los rangos de la variable i respectivamente.

(c) **No**. Ya que la variable i aparece ligada a los dos cuantificadores.

(d) **Sí**. Las apariciones de la variable libre k han sido sustituidas por m. Además, la variable cuantificada i ha sido cambiada por otra variable (j), lo cual es independiente de cualquier sustitución.

3. Dado el programa $i, j := 0, 0;$ **if** $B_1 \wedge B_2 \rightarrow i := 4; j := 3 |\ B_1 \vee B_2 \rightarrow j, i := 3, 4$ **fi**

(a) es siempre determinista

(b) es no determinista si B_1 es cierto y B_2 es cierto

(c) es siempre equivalente a $i, j, := 4, 3$

(d) es equivalente a $i, j := 0, 0;$ **if** $B_1 \rightarrow i := 4; j := 3$**fi**

(a) **No**. Si B_1 o B_2 son ciertos existen dos ramas con la misma condición y la traza generada puede variar para el mismo estado inicial.

(b) **Sí**. Es uno de los casos por el que no se cumple el apartado anterior.

(c) **No**. Si B_1 y B_2 son falsos el programa termina con i y j con valor cero.

(d) **No**. Si B_1 es falso, el resultado del programa del enunciado dependerá del valor de B_2, cosa que no ocurre con el programa de esta respuesta.

$\boxed{\mathbf{4.}}$ Si tenemos que $\{Q\}S\{j > 0 \wedge i < 0\}$ entonces:

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

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

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

(d) $\{Q \wedge A\}S\{j > 0 \wedge i < 0\}$

(a) **Sí**. Ya que el conjunto de estados finales del enunciado es un subconjunto de los estados finales de este caso.

(b) **No**. En este caso los conjuntos de estados finales de una y otra solución son disjuntos.

(c) **No**. La precondición en este caso es más débil que la del enunciado y no garantiza que podamos obtener $(j > 0 \wedge i < 0)$ si Q no se cumple.

(d) **Sí**. La precondición es más fuerte que Q y como $Q \wedge A \Rightarrow Q$, tenemos de nuevo el enunciado.

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

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

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

Primero calcularemos:

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

$\equiv dominio(b[k])$ cand $enrango(i, b)$ cand $enrango(k, b)$ cand $(b[i] \neq b[j])^b_{(b;i:b[k])}$

Desarrollando el último término obtenemos:

$(b[i] \neq b[j])^b_{(b;i:b[k])}$

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

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

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

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

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

A continuación sustituimos en la expresión original y resolvemos:

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

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

$\equiv dominio(b[b[j]])$ cand $dominio(b[k])$ cand $enrango(j, b)$ cand $enrango(i, b)$ cand $dominio(b[k])$
cand $enrango(i, b)$ cand $enrango(k, b)$ cand $(i \neq j \wedge b[k] \neq b[j])^b_{(b;i:b[b[j]];j:b[k])}$

$\equiv dominio(b[b[j]])$ cand $dominio(b[k])$ cand $enrango(j, b)$ cand $enrango(i, b)$ cand $enrango(k, b)$
cand $(i \neq j \wedge b[k] \neq b[j])^b_{(b;i:b[b[j]];j:b[k])}$

Resolviendo el último término:

$(i \neq j \wedge b[k] \neq b[j])^b_{(b;i:b[b[j]];j:b[k])}$

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

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

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

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

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

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

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

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

Simplificando y sustituyendo en la expresión de partida la solución queda como sigue:

$\equiv dominio(b[b[j]])$ cand $dominio(b[k])$ cand $enrango(j, b)$ cand $enrango(i, b)$ cand $enrango(k, b)$
cand $(i = k \wedge j \neq k \wedge b[b[j]] \neq b[k])$

$\boxed{6.}$ (3,5 puntos) Escribe un programa que obtenga en x el número de huecos de un array b con índices entre 0 y $n-1$, de valores enteros, con $n > 0$. Un hueco es una secuencia, lo más larga posible, de uno o más ceros consecutivos. Ejemplo: Dado el array $b = (3, 6, 0, 0, 0, 7, 4, 1, 0, 0, 0, 0, 8)$, el número de huecos sería 2. Se pide:

6.1 Establecer una precondición y una postcondición adecuadas

6.2 Fijar una invariante y una función cota

6.3 Escribir el programa y anotarlo; y

6.4 Demostrar su corrección *total*

- Si el array tiene todos sus elementos con valor cero, existe un hueco $\Rightarrow x = 1$.
- Definiciones de *hueco*:
  1. hueco $= (\exists I, K : I \in [0, n-1], K \in [0, n-1] : b[I : K] = 0)$
  2. Número de huecos = número de ceros - número de pares de ceros $x = (\mathbf{N}K : K \in [0, n-1] : b[K] = 0) - (\mathbf{N}K : K \in [0, n-2] : b[K] = 0 \wedge b[K+1] = 0)$
  3. Número de huecos = número de inicios de hueco. Es necesario considerar el caso especial del hueco al principio del array. $x = (\mathbf{N}K : K \in [-1, n-2] : (K = -1$ cor $b[K] \neq 0) \wedge b[K+1] = 0)$
  4. Número de huecos = número de fines de hueco. Es necesario considerar el caso especial del hueco al final del array. $x = (\mathbf{N}K : K \in [0, n-1] : b[k] = 0 \wedge (K = n-1$ cor $b[K+1] \neq 0))$
  5. $x = (\mathbf{N}K : K \in [0, n-2] : b[K] = 0 \wedge b[K+1] \neq 0) + (\mathbf{N}K : K \in [n-1, n-1] : b[K] = 0)$

6.1 Establecer una precondición y una postcondición adecuadas

La precondición es inmediata y para obtener la postcondición es necesario tener en cuenta el último elemento (o el primero) del array como caso particular. Existen varias soluciones y una de ellas es la siguiente:

$$\{Q: \ b[0 : n-1] : integer \wedge n > 0 \wedge b = B\}$$
$$\{R': \ x = (\mathbf{N}K : 0 \leq K < n-1 : b[K] = 0 \wedge b[K+1] \neq 0)\}$$
$$\{Q' \equiv R'\}$$
$$\{R: \ x = (\mathbf{N}K : 0 \leq K \leq n-1 : b[K] = 0 \wedge (K = n-1 \text{ cor } b[K+1] \neq 0))\}$$

6.2 Fijar una invariante y una función cota

Para construir una invariante aplicamos el método de sustituir una constante, $n-1$ por ejemplo, por una nueva variable i. Como resultado, el rango de i será en principio $i \leq n-1$. Para fijar el valor inicial basta con comprobar que cuando $i = 0$ el cuantificador pasa a tener un rango vacío, valiendo cero.

$$\{P: \ 0 \leq i \leq n-1 \wedge x = (\mathbf{N}K \in [0 : i-1] : b[K] = 0 \wedge b[K+1] \neq 0)\}$$
$$\{t: \ (n-1) - i\}$$

6.3 Escribir el programa y anotarlo

$$
\begin{array}{l}
\{Q\} \\
S_0 : i, x := 0, 0; \\
\{P\} \\
\{t\} \\
\text{do } B : i < n-1 \rightarrow \ \text{if } B_1 : b[i] = 0 \wedge b[i+1] \neq 0 \rightarrow \{P \wedge B_1\} S_1 : i, x := i+1, x+1; \{P\} \\
\qquad\qquad\qquad\qquad\ \ \| \ B_2 : b[i] = 0 \wedge b[i+1] = 0 \rightarrow \{P \wedge B_2\} S_2 : i := i+1; \{P\} \\
\qquad\qquad\qquad\qquad\ \ \| \ B_3 : b[i] \neq 0 \rightarrow \{P \wedge B_3\} S_3 : i := i+1; \{P\} \\
\qquad\qquad\qquad\qquad\ \ \text{fi} \\
\text{od} \\
\{P \wedge \neg B\} \\
\{R'\} \\
\{Q'\} \\
\text{if } B_4 : b[n-1] = 0 \rightarrow \{P \wedge \neg B \wedge B_4\} S_4 : x := x+1; \{P \wedge \neg B\} \\
\| \ B_5 : b[n-1] \neq 0 \rightarrow \{P \wedge \neg B \wedge B_5\} S_5 : skip; \{P \wedge \neg B\} \\
\text{fi} \\
\{R\}
\end{array}
$$

6.4 Demostrar su corrección *total*

1. $Q \Rightarrow wp(\text{"}i, x := 0, 0\text{"}, P)$

$b[0:n-1] : integer \wedge n > 0 \wedge b = B \Rightarrow wp(\text{"}i, x := 0, 0\text{"}, P)$
$\equiv n > 0 \Rightarrow 0 \leq 0 \leq n-1 \wedge 0 = (\mathbf{N}K \in [0:-1] : b[K] = 0 \wedge b[K+1] \neq 0)$
El cuantificador numérico de un rango vacío vale 0
$\equiv n > 0 \Rightarrow 0 \leq n-1 \wedge 0 = 0$
$\equiv n > 0 \Rightarrow 0 \leq n-1$
T

2. $\{P \wedge B\} IF_1 \{P\}$

2.1 $\{P \wedge B \wedge B_1\} S_1 : i, x := i+1, x+1; \{P\}$

El wp correspondiente es, en este caso:

$wp(\text{"}i, x := i+1, x+1\text{"}, P) = 0 \leq i+1 \leq n-1 \wedge x+1 = (\mathbf{N}K \in [0:i] : b[K] = 0 \wedge b[K+1] \neq 0)$

Separando el caso $K = i$

$\equiv 0 \leq i+1 \leq n-1 \wedge x+1 = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0) + \alpha(i)$

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

$$\alpha(i) \begin{cases} 1 \text{ si } b[i] = 0 \wedge b[i+1] \neq 0 \\ 0 \text{ si } b[i] \neq 0 \end{cases}$$

Por el valor de $B_1 : b[i] = 0 \wedge b[i+1] \neq 0$, $\alpha(i)$ toma el valor 1 y la expresión anterior queda:

$\equiv 0 \leq i+1 \leq n-1 \wedge x+1 = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0) + 1$

Restando 1 en ambos lados, la expresión $x = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0)$ se obtiene directamente de P. Por otro lado,

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

2.2 $\{P \wedge B \wedge B_2\} S_2 : i := i+1; \{P\}$

El wp correspondiente es, en este caso:

$wp(\text{"}i := i+1\text{"}, P) = 0 \leq i+1 \leq n-1 \wedge x = (\mathbf{N}K \in [0:i] : b[K] = 0 \wedge b[K+1] \neq 0)$

Separando el caso $K = i$

$\equiv 0 \leq i+1 \leq n-1 \wedge x = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0) + \alpha(i)$

donde $\alpha(i)$ puede tomar los valores indicados en el apartado [2.1].

Por el valor de $B_2 : b[i] = 0 \wedge b[i+1] = 0$, $\alpha(i)$ toma el valor 0 y la expresión anterior queda:

$\equiv 0 \leq i+1 \leq n-1 \wedge \overbrace{x = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0) + 0}$

La expresión obtenida $\overbrace{\quad}$ aparece directamente de P. Por otro lado,

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

2.3 $\{P \wedge B \wedge B_3\} S_3 : i := i+1; \{P\}$

El wp correspondiente es, en este caso:

$wp(\text{"}i := i+1\text{"}, P) = 0 \leq i+1 \leq n-1 \wedge x = (\mathbf{N}K \in [0:i] : b[K] = 0 \wedge b[K+1] \neq 0)$

Separando el caso $K = i$

$\equiv 0 \leq i+1 \leq n-1 \wedge x = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0) + \alpha(i)$

donde $\alpha(i)$ puede tomar los valores indicados en el apartado [2.1].

Por el valor de $B_3 : b[i] = 0 \wedge b[i+1] = 0$, $\alpha(i)$ toma el valor 0, y la expresión anterior queda:

$\equiv 0 \leq i+1 \leq n-1 \wedge \overbrace{x = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0) + 0}$

La expresión obtenida $\overbrace{\qquad}$ aparece directamente de P. Por otro lado,

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

3. $P \wedge \neg B \Rightarrow R'$

$\equiv 0 \leq i \leq n-1 \wedge x = (\mathbf{N}K \in [0:i-1] : b[K] = 0 \wedge b[K+1] \neq 0) \wedge i \geq n-1$

$\Rightarrow x = (\mathbf{N}K : 0 \leq K < n-1 : b[K] = 0 \wedge b[K+1] \neq 0)$

Que es cierto ya que i sólo puede tomar el valor $n-1$, y si sustituimos i por este valor en la parte izquierda de la implicación tenemos directamente R'.

Quedaría por demostrar el segundo IF, es decir, $\{Q'\} IF_2 \{R\}$. Ello implica demostrar:

3.1 $Q' \Rightarrow BB$

$\equiv x = (\mathbf{N}K : 0 \leq K < n-1 : b[K] = 0 \wedge b[K+1] \neq 0) \Rightarrow (b[n-1] = 0 \vee b[n-1] \neq 0)$

$\equiv x = (\mathbf{N}K : 0 \leq K < n-1 : b[K] = 0 \wedge b[K+1] \neq 0) \Rightarrow T$

Que es cierto.

3.2 $Q' \wedge B_4 \Rightarrow wp(S_4, R)$

$\equiv x = (\mathbf{N}K : 0 \leq K < n-1 : b[K] = 0 \wedge b[K+1] \neq 0) \wedge b[n-1] = 0$
$\Rightarrow x = (\mathbf{N}K : 0 \leq K \leq n-1 : b[K] = 0 \wedge (K = n-1 \text{ cor } b[K+1] \neq 0))$

Con P y B_4 se cumple $x = (\mathbf{N}K : 0 \leq K \leq n-1 : b[K] = 0 \wedge K = n-1)$, haciendo cierta la parte derecha de la implicación.

3.3 $Q' \wedge B_5 \Rightarrow wp(S_5, R)$

$\equiv x = (\mathbf{N}K : 0 \leq K < n-1 : b[K] = 0 \wedge b[K+1] \neq 0) \wedge b[n-1] \neq 0$
$\Rightarrow x = (\mathbf{N}K : 0 \leq K \leq n-1 : b[K] = 0 \wedge (K = n-1 \text{ cor } b[K+1] \neq 0))$

Con P y B_5 se cumple $x = (\mathbf{N}K : 0 \leq K \leq n-1 : b[K] = 0 \wedge b[K+1] \neq 0)$, haciendo cierta la parte derecha de la implicación.

4. $P \wedge B \Rightarrow (n-1) - i > 0$

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

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

5. $\{P \wedge B\} t_1 := t; S_i \{t < t_1\}$

5.1 $\{P \wedge B\} t_1 := t; S_1 \{t < t_1\}$

Si calculamos el wp correspondiente obtenemos:

$wp(\text{"}t_1 := t; i, x := i+1, x+1\text{"}, (n-1) - i < t_1) = (((n-1) - i < t_1)^{i,x}_{i+1,x+1})^{t_1}_{(n-1)-i}$

$\equiv ((n-1) - (i+1) < t_1)^{t_1}_{(n-1)-i}$

$\equiv (n-1) - i - 1 < (n-1) - i$

$\equiv -1 < 0$

$\equiv T$

5.2 $\{P \wedge B\} t_1 := t; S_2 \{t < t_1\}$

Si calculamos el wp correspondiente obtenemos:

$wp(\text{"}t_1 := t; i := i+1\text{"}, (n-1) - i < t_1) = (((n-1) - i < t_1)^{i}_{i+1})^{t_1}_{(n-1)-i}$

$\equiv ((n-1) - (i+1) < t_1)^{t_1}_{(n-1)-i}$

$\equiv (n-1) - i - 1 < (n-1) - i$

$\equiv -1 < 0$

$\equiv T$

5.3 $\{P \wedge B\} t_1 := t; S_3 \{t < t_1\}$

Si calculamos el wp correspondiente obtenemos:

$wp(\text{"}t_1 := t; i := i+1\text{"}, (n-1) - i < t_1) = (((n-1) - i < t_1)^{i}_{i+1})^{t_1}_{(n-1)-i}$

$\equiv ((n-1) - (i+1) < t_1)^{t_1}_{(n-1)-i}$

$\equiv (n-1) - i - 1 < (n-1) - i$

$\equiv -1 < 0$

$\equiv T$

$\boxed{\textbf{7.}}$ (1,5 punto) El siguiente programa deja en la variable $maxdif$ la mayor diferencia entera entre un elemento y su anterior.

$\{Q:\ b[0:n-1]:integer \wedge n>1 \wedge b=B\}$
$i, maxdif := 1, b[1]-b[0];$
do $i \neq n-1 \rightarrow dif := b[i+1]-b[i];$
    if $dif \geq maxdif \rightarrow maxdif := dif;$
    $\mid dif \leq maxdif \rightarrow skip;$
    fi
    $i := i+1;$
od

Se pide:

7.1 Establecer una postcondición adecuada

7.2 Fijar una invariante, y

7.3 Demostrar su corrección *parcial*

7.1 Establecer una postcondición adecuada $\{R:\ (\forall K \in [0,n-2]: maxdif \geq b[K+1]-b[K]) \wedge (\exists K \in [0,n-2]: maxdif = b[K+1]-b[K])\}$

7.2 Fijar una invariante$\{P:\ (\forall K \in [0,i-1]: maxdif \geq b[K+1]-b[K]) \wedge (\exists K \in [0,i-1]: maxdif = b[K+1]-b[K])\}$

7.3 Demostrar su corrección *parcial*

Establecemos que:

a) $R_{IF} = P^{i}_{i+1}$

b) Q_{IF} debe cumplir $P \wedge i \neq n-1 \Rightarrow wp(\text{"}dif := b[i+1]-b[i]\text{"}, Q_{IF})$

por lo tanto: $Q_{IF} = P \wedge dif = b[i+1]-b[i]$ y el programa anotado quedaría:

$\{Q\}$
$S_0: i, maxdif := 1, b[1]-b[0];$
$\{P\}$
do $B: i \neq n-1 \rightarrow S_1: dif := b[i+1]-b[i];$
$\{Q_{IF}\}$
    if $B_1: dif \geq maxdif \rightarrow S_2: maxdif := dif;$
    $\mid B_2: dif \leq maxdif \rightarrow S_3: skip;$
    fi
    $S_4: i := i+1;$
$\{R_{IF}\}$
od
$\{R\}$

Demostramos entonces la corrección parcial del programa:

7.3.1 $Q \Rightarrow wp(S_0, P)$

Si calculamos el wp obtenemos:

$wp(\text{"}i, maxdif := 1, b[1]-b[0]\text{"}, P)$

$\equiv (\forall K \in [0,0]: b[1]-b[0] \geq b[K+1]-b[K]) \wedge (\exists K \in [0,0]: b[1]-b[0] = b[K+1]-b[K])$

$\equiv b[1]-b[0] \geq b[1]-b[0]) \wedge (b[1]-b[0] = b[1]-b[0])$

Que es T. Por lo tanto la implicación $Q \Rightarrow wp(S_0, P)$ también es cierta.

7.3.2 $\{P \wedge i \neq n-1\}S_1; IF; S_4\{P\}$

Para hacer la demostración de este paso necesitamos demostrar:

a) $Q_{IF} : P \wedge dif = b[i+1] - b[i] \Rightarrow BB$

$\equiv P \wedge dif = b[i+1] - b[i] \Rightarrow (dif \geq maxdif \vee dif \leq maxdif)$

$\equiv P \wedge dif = b[i+1] - b[i] \Rightarrow T$

que es cierto.

b) $Q_{IF} \wedge B_2 \Rightarrow wp(S_2, R_{IF})$

$\equiv (\forall K \in [0, i-1] : maxdif \geq b[K+1] - b[K]) \wedge (\exists K \in [0, i-1] : maxdif = b[K+1] - b[K])$
$\wedge dif = b[i+1] - b[i] \wedge dif \geq maxdif \Rightarrow wp(\text{"}maxdif := dif\text{"}, R_{IF})$

$\equiv (\forall K \in [0, i-1] : maxdif \geq b[K+1] - b[K]) \wedge (\exists K \in [0, i-1] : maxdif = b[K+1] - b[K]))$
$\wedge dif = b[i+1] - b[i] \wedge dif \geq maxdif \Rightarrow (\forall K \in [0, i] : dif \geq b[K+1] - b[K]) \wedge (\exists K \in [0, i] : dif = b[K+1] - b[K])$

que es cierto ya que por un lado (a) el cuantificador existencial es cierto por $dif = b[i+1] - b[i]$, y (b) el cuantificador universal es cierto en el rango $[0, i]$ porque $dif = b[i+1] - b[i] \geq maxdif$ y $maxdif$ es la mayor diferencia en el rango $[0, i-1]$.

c) $Q_{IF} \wedge B_3 \Rightarrow wp(S_3, R_{IF})$

$\equiv (\forall K \in [0, i-1] : maxdif \geq b[K+1] - b[K]) \wedge (\exists K \in [0, i-1] : maxdif = b[K+1] - b[K]))$
$\wedge dif = b[i+1] - b[i] \wedge dif \leq maxdif \Rightarrow wp(\text{"}skip\text{"}, R_{IF})$

$\equiv (\forall K \in [0, i-1] : maxdif \geq b[K+1] - b[K]) \wedge (\exists K \in [0, i-1] : maxdif = b[K+1] - b[K]))$
$\wedge dif = b[i+1] - b[i] \wedge dif \leq maxdif \Rightarrow (\forall K \in [0, i-1] : maxdif \geq b[K+1] - b[K])$
$\wedge (\exists K \in [0, i-1] : maxdif = b[K+1] - b[K]))$

que es cierto ya que si existe en el intervalo $[0, i-1]$ un valor $maxdif$ que cumple $maxdif \geq b[i+1] - b[i]$, también existe en el intervalo $[0, i]$. Y además, el cuantificador universal de P junto con S_1 y B_3 nos permiten establecer como cierto el cuantificador universal de R_{IF}.

7.3.3 $P \wedge \neg B \Rightarrow R$

$(\forall K \in [0, i-1] : maxdif \geq b[K+1] - b[K]) \wedge (\exists K \in [0, i-1] : maxdif = b[K+1] - b[K]) \wedge (i = n-1)$
$\Rightarrow (\forall K \in [0, n-2] : maxdif \geq b[K+1] - b[K]) \wedge (\exists K \in [0, n-2] : maxdif = b[K+1] - b[K])$

Que es trivial ya que si en P sustituimos i por $n-1$ tal como indica $\neg B$ obtenemos directamente R.