2. Programa que intercambia los elementos de A(1..n) y B(1..n) --

(1) ≡ $\{n \geq 1 \wedge \forall k(1 \leq k \leq n \rightarrow (A(k) = a_k \wedge B(k) = b_k))\}$
(9) ≡ $\{\forall k(1 \leq k \leq n \rightarrow (A(k) = b_k \wedge B(k) = a_k))\}$

A continuación, a partir de la precondición se dan las aserciones correspondientes a las inicializaciones previas al while:

(2) ≡ $\{n \geq 1 \wedge \forall k(1 \leq k \leq n \rightarrow (A(k) = a_k \wedge B(k) = b_k)) \wedge i = 1\}$

El (2) se podría dar de manera abreviada:

(2) ≡ $\{(1) \wedge i = 1\}$, ya que la primera parte coincide con el (1)

Luego, a partir de la postcondición se calcula el invariante:

(3) ≡ {INV} ≡ $\{(1 \leq i \leq n + 1) \wedge \forall k(1 \leq k \leq i - 1 \rightarrow (A(k) = b_k \wedge B(k) = a_k))\}$

A partir del invariante se calculan las aserciones que van dentro del while:

(4) ≡ $\{(1 \leq i \leq n) \wedge \forall k(1 \leq k \leq i - 1 \rightarrow (A(k) = b_k \wedge B(k) = a_k))\}$

(5) ≡ $\{(1 \leq i \leq n) \wedge \forall k(1 \leq k \leq i - 1 \rightarrow (A(k) = b_k \wedge B(k) = a_k)) \wedge aux = A(i) = a_i\}$

En el (5) hay que poner $aux = A(i) = a_i$ para indicar claramente que A(i) todavía tiene su valor inicial.

El (5) se podría dar de manera abreviada:

(5) ≡ $\{(4) \wedge aux = A(i) = a_i\}$, ya que la primera parte coincide con el (4)

(6) ≡ $\{(1 \leq i \leq n) \wedge \forall k(1 \leq k \leq i - 1 \rightarrow (A(k) = b_k \wedge B(k) = a_k)) \wedge aux = a_i \wedge A(i) = B(i) = b_i\}$

En el (6) no se puede poner $aux = A(i) = a_i$ porque en A(i) se ha guardado el valor de B(i). Por otro lado B(i) todavía tiene su valor inicial b_i.

El (6) se podría dar de manera abreviada:

(6) ≡ $\{(4) \wedge aux = a_i \wedge A(i) = B(i) = b_i\}$, ya que la primera parte coincide con el (4)

¡¡CUIDADO!! (6) ≡ $\{(5) \wedge A(i) = B(i) = b_i\}$ ESTÁ MAL ya que no se cumple $aux = A(i)$.

(7) ≡ $\{(1 \leq i \leq n) \wedge \forall k(1 \leq k \leq i - 1 \rightarrow (A(k) = b_k \wedge B(k) = a_k)) \wedge B(i) = aux = a_i \wedge A(i) = b_i\}$

Pero esa fórmula se suele poner de la siguiente manera:

$(7) \equiv \{(1 \le i \le n) \wedge \forall k(1 \le k \le i \rightarrow (A(k) = b_k \wedge B(k) = a_k))\}$

Ya que una vez hecho el intercambio correspondiente a la posición i, el valor de la variable aux deja de ser relevante.

De todas formas, también sería correcto poner:

$(7) \equiv \{(1 \le i \le n) \wedge \forall k(1 \le k \le i \rightarrow (A(k) = b_k \wedge B(k) = a_k)) \wedge aux = a_i = B(i)\}$

$(8) \equiv \{(2 \le i \le n + 1) \wedge \forall k(1 \le k \le i - 1 \rightarrow (A(k) = b_k \wedge B(k) = a_k)) \wedge aux = a_{i-1} = B(i - 1)\}$

En la aserción intermedia (8) es importante fijarse en $aux = a_{i-1} = B(i - 1)$ donde tenemos $i - 1$.

Para terminar se da la expresión cota:
$(10) \equiv \{E = n + 1 - i\}$

La expresión E es "último valor que tomará i menos i" e indica cuántas vueltas quedan por dar. La información para calcular E está ya en el invariante:

$(3) \equiv \{INV\} \equiv \{(1 \le i \le n + 1) \wedge \forall k(1 \le k \le i - 1 \rightarrow (A(k) = b_k \wedge B(k) = a_k))\}$