44. (Abril 2010 #2) Predicados bits(E(1..r)) y rotado(F(1..r), (f_1, f_2, …, f_r), G(1..r), (g_1, g_2, …, g_r), H(1..r), (h_1, h_2, …, h_r), Q(1..r), pos) y programa que, dados cuatro vectores A(1..n), B(1..n), C(1..n) y D(1..n) donde al principio D(1..n) sólo contiene ceros y unos, si C(i) es 0 se rota A(i) → B(i) → C(i) → A(i) y si C(i) es 1 se rota A(i) → C(i) → B(i) → A(i). -- #

a) **bits(E(1..r))** $\equiv \forall k\,(1 \le k \le r \to E(k) = 0 \vee E(k) = 1)$

b) **rotado(F(1..r), (f_1, f_2, …, f_r), G(1..r), (g_1, g_2, …, g_r), H(1..r), (h_1, h_2, …, h_r), Q(1..r), pos)** ≡

$\{(0 \le pos \le r) \wedge$
$bits(Q(1..r)) \wedge$
$\forall k\,((1 \le k \le pos \wedge Q(k) = 0) \to (F(k) = h_k \wedge G(k) = f_k \wedge H(k) = g_k)) \wedge$
$\forall k\,((1 \le k \le pos \wedge Q(k) = 1) \to (F(k) = g_k \wedge G(k) = h_k \wedge H(k) = f_k))\}$

Otra posibilidad para b)

rotado(F(1..r), (f_1, f_2, …, f_r), G(1..r), (g_1, g_2, …, g_r), H(1..r), (h_1, h_2, …, h_r), Q(1..r), pos) ≡

$\{(0 \le pos \le r) \wedge$
$bits(Q(1..r)) \wedge$
$\forall k\,(1 \le k \le pos \to ((Q(k) = 0 \to (F(k) = h_k \wedge G(k) = f_k \wedge H(k) = g_k)) \wedge$
$(Q(k) = 1 \to (F(k) = g_k \wedge G(k) = h_k \wedge H(k) = f_k))))\}$

c) Las aserciones se darán en el orden en el que es más natural formularlas:

(1) {Precondición} ≡ $\{n \ge 1 \wedge$
$\forall k\,(1 \le k \le n \to (A(k) = a_k \wedge B(k) = b_k \wedge C(k) = c_k)) \wedge bits(D(1..n))\}$

En la precondición se indica que los vectores A, B, C y D tendrán por lo menos un elemento, que los valores iniciales de A, B y C los representaremos mediante *a*'s, b's y c's minúsculas con los correspondientes subíndices y que en cada posición de D se tiene un cero o uno.

(2) {Aserción intermedia} ≡ $\{(1) \wedge i = 0\}$

(10) {Postcondición} ≡
{rotado(A(1..n), (a_1, a_2, …, a_n), B(1..n), (b_1, b_2, …, b_n), C(1..n), (c_1, c_2, …, c_n), D(1..n), n)}

En la postcondición se indica que se han hecho todos los cambios hasta la posición n incluida, es decir, se ha recorrido todo el vector y se han hecho todos los cambios necesarios.

(3) {Invariante} ≡
{$(0 \le i \le n) \wedge$ rotado(A(1..n), ($a_1, a_2, \ldots, a_n$), B(1..n), ($b_1, b_2, \ldots, b_n$), C(1..n), ($c_1, c_2, \ldots, c_n$), D(1..n), i)}

En el invariante se indica que se han hecho todos los cambios hasta la posición i incluida, es decir, estamos situados en la posición i y ya se ha analizado lo que ocurre en la posición i.

(4) {Aserción intermedia} ≡
{$(0 \le i \le n-1) \wedge$ rotado(A(1..n), ($a_1, a_2, \ldots, a_n$), B(1..n), ($b_1, b_2, \ldots, b_n$), C(1..n), ($c_1, c_2, \ldots, c_n$), D(1..n), i)}

Por haber entrado en el while sabemos que se ha cumplido la condición del while y que i no es n.

(5) {Aserción intermedia} ≡
{$(1 \le i \le n) \wedge$ rotado(A(1..n), ($a_1, a_2, \ldots, a_n$), B(1..n), ($b_1, b_2, \ldots, b_n$), C(1..n), ($c_1, c_2, \ldots, c_n$), D(1..n), i – 1)}

Se ha incrementado el valor de i y por tanto también los límites del intervalo en el que se mueve i. Pero ahora los cambios en los vectores ya no están hechos hasta i sino que hasta i – 1.

(6) {Aserción intermedia} ≡
{$(1 \le i \le n) \wedge$ rotado(A(1..n), ($a_1, a_2, \ldots, a_n$), B(1..n), ($b_1, b_2, \ldots, b_n$), C(1..n), ($c_1, c_2, \ldots, c_n$), D(1..n), i – 1) $\wedge$ D(i) = 0}

Por haber entrado en la rama then del if sabemos que en D(i) tenemos un 0.

(7) {Aserción intermedia} ≡
{$(1 \le i \le n) \wedge$ rotado(A(1..n), ($a_1, a_2, \ldots, a_n$), B(1..n), ($b_1, b_2, \ldots, b_n$), C(1..n), ($c_1, c_2, \ldots, c_n$), D(1..n), i – 1) $\wedge$ D(i) = 0 $\wedge$ aux = C(i) = c_i}

Tras ejecutarse la asignación aux : = C(i); sabemos que ahora en aux tenemos lo mismo que en C(i) que es el valor inicial c_i. Pero de momento los cambios completos están hechos hasta la posición i – 1, en la posición i estamos a medias por lo que en el predicado rotado seguimos poniendo i – 1.

(8) {Aserción intermedia} ≡
{$(1 \le i \le n) \wedge$ rotado(A(1..n), ($a_1, a_2, \ldots, a_n$), B(1..n), ($b_1, b_2, \ldots, b_n$), C(1..n), ($c_1, c_2, \ldots, c_n$), D(1..n), i – 1) $\wedge$ D(i) = 0 $\wedge$ aux = c_i $\wedge$ C(i) = B(i) = b_i}

Tras ejecutarse la asignación C(i) : = B(i); sabemos que ahora en C(i) tenemos lo mismo que en B(i) que es el valor inicial b_i. Ahora en aux ya no se tiene lo mismo que en C(i), se tiene el valor inicial de C(i), es decir, c_i. Todavía los cambios completos están hechos hasta la posición i – 1, en la posición i estamos a medias por lo que en el predicado rotado seguimos poniendo i – 1.

(9) {Aserción intermedia} ≡
$\{(1 \le i \le n) \wedge$ rotado(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, C(1..n), $(c_1, c_2, \ldots, c_n)$, D(1..n), $i - 1) \wedge D(i) = 0 \wedge aux = c_i \wedge C(i) = b_i \wedge B(i) = A(i) = a_i\}$

Tras ejecutarse la asignación B(i) : = A(i); sabemos que ahora en B(i) tenemos lo mismo que en A(i) que es el valor inicial a_i. Ahora en aux ya no se tiene lo mismo que en C(i), se tiene el valor inicial de C(i), es decir, c_i y en C(i) ya no se tiene lo mismo que en B(i), se tiene el valor inicial de B(i), es decir, b_i. Todavía los cambios completos están hechos hasta la posición i – 1, en la posición i estamos a medias por lo que en el predicado rotado seguimos poniendo i – 1.

(11) E = n – i

La expresión cota nos indica, siempre que estemos en el punto donde se cumple el invariante, cuántas vueltas quedan por dar. Es "el último valor que tomará i" menos "i". La expresión E es al fin y al cabo la distancia entre n y la variable i. Cuando i crece, la distancia disminuye y el número de vueltas pendientes decrece.

Las partes coloreadas resaltan los cambios que hay de una aserción a otra.