42. (Septiembre 2009) Predicados nocerosunos(C(1..r)), bits(C(1..r)) y sustituido(D(1..q), (d_1, d_2, …, d_q), E(1..q), (e_1, e_2, …, e_q), pos) y programa que que, dados dos vectores de enteros A(1..n) y B(1..n), donde A(1..n) no contiene ningún cero ni ningún 1 y B(1..n) solo contiene unos y ceros, sustituye cada cero de B(1..n) por el elemento de A(1..n) que ocupa la misma posición dejando el 0 en esa posición de A(1..n). -- #

a) nocerosunos(C(1..r)) $\equiv \{\forall k\, (1 \le k \le r \rightarrow (C(k) \ne 0 \wedge C(k) \ne 1))\}$
b) bits(C(1..r)) $\equiv \{\forall k\, (1 \le k \le r \rightarrow (C(k) = 0 \vee C(k) = 1))\}$

c) sustituido(D(1..q), (d_1, d_2, …, d_q), E(1..q), (e_1, e_2, …, e_q), pos) $\equiv$
$\{(0 \le pos \le q) \wedge$
$\forall k\, (1 \le k \le pos \rightarrow ((e_k = 0 \rightarrow (D(k) = 0 \wedge E(k) = d_k)) \wedge$
$\wedge\ (e_k = 1 \rightarrow (D(k) = d_k \wedge E(k) = e_k))))\}$

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

(1) {Precondición} $\equiv \{n \ge 1 \wedge$ nocerosunos(A(1..n)) $\wedge$ bits(B(1..n)) $\wedge$
$\forall k\, (1 \le k \le n \rightarrow (A(k) = a_k \wedge B(k) = b_k))\}$

En la precondición se indica que los vectores A y B tendrán por lo menos un elemento, que en A(1..n) no aparecen el cero y el uno, que en B(1..n) sólo hay unos y ceros y que los valores iniciales de A(1..n) y B(1..n) los representaremos mediante *a*'s y b's minúsculas con los correspondientes subíndices.

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

(9) {Postcondición} $\equiv$
{sustituido(A(1..n), (a_1, a_2, …, a_n), B(1..n), (b_1, b_2, …, b_n), n)}

En la postcondición se indica que se han hecho todas las sustituciones hasta la posición n incluida, es decir, se ha recorrido todo el vector y se han hecho todas las sustituciones necesarias.

(3) {Invariante} $\equiv \{(0 \le i \le n) \wedge$
sustituido(A(1..n), (a_1, a_2, …, a_n), B(1..n), (b_1, b_2, …, b_n), i)}

En el invariante se indica que se han hecho todas las sustituciones hasta la posición i incluida, es decir, estamos situados en la posición i y el intervalo que va de 1 a i ya se ha analizado.

(4) {Aserción intermedia} $\equiv \{(0 \le i \le n - 1) \wedge$
sustituido(A(1..n), (a_1, a_2, …, a_n), B(1..n), (b_1, b_2, …, b_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} ≡ {$(1 \le i \le n) \wedge$
sustituido(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, $i - 1$)}

En el punto (4) se ha indicado que se han hecho todas las sustituciones hasta la posición i incluida. Ahora el valor de i ha crecido, se ha incrementado, pero no hemos analizado la nueva posición que señala i, por tanto, las sustituciones no han avanzado y están hechas hasta la posición i – 1. Por otro lado en (4) decíamos que i podía estar entre 0 y n – 1, pero como ha crecido ahora podemos asegurar que estará entre 1 y n.

(6) {Aserción intermedia} ≡ {$(1 \le i \le n) \wedge B(i) = b_i = 0 \wedge$
sustituido(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, $i - 1$)}

Por haber entrado en la rama then del if sabemos que en la posición i del vector B hay un cero, es decir, $b_i = 0$.

(7) {Aserción intermedia} ≡ {$(1 \le i \le n) \wedge b_i = 0 \wedge B(i) = A(i) = a_i \wedge$
sustituido(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, $i - 1$)}

Tras ejecutarse la asignación B(i) : = A(i); sabemos que b_i (el valor inicial de B(i)) es 0 y que ahora en B(i) y en A(i) tenemos a_i (el valor inicial de A(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 *sustituido* seguimos poniendo i – 1.

(8) {Aserción intermedia} ≡ {$(1 \le i \le n) \wedge b_i = 0 \wedge B(i) = a_i \wedge A(i) = 0$
sustituido(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, i)}

Tras ejecutarse la asignación A(i) : = 0; sabemos que b_i (el valor inicial de B(i)) es 0 y que ahora en B(i) tenemos a_i y que en A(i) tenemos 0. Ahora ya están hechos todos los cambios correspondientes a la posición i y por ello en el predicado *sustituido* ponemos i.

(10) $E = n - i$

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.