41. (Junio 2009) Predicados par(x), parimpar(C(1..r), D(1..r)) y cambiadopar(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos) y programa que dados dos vectores A(1..n) y B(1..n) de signo opuesto, decrementa en 1 cada elemento impar de A(1..n) e incrementa en 1 cada elemento par de B(1..n). -- #

a) $par(x) \equiv \{x \bmod 2 = 0\}$

b) **parimpar(C(1..r), D(1..r))** $\equiv$
$\{\forall k\, (1 \le k \le r \rightarrow (par(C(k)) \rightarrow \neg par(D(k)))) \wedge$
$\forall k\, (1 \le k \le r \rightarrow (\neg par(C(k)) \rightarrow par(D(k))))\}$

c) **cambiopar(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos)** $\equiv$
$\{(0 \le pos \le r) \wedge$
$\forall k\, (1 \le k \le pos \rightarrow ((\neg par(c_k) \rightarrow (C(k) = c_k - 1 \wedge D(k) = d_k + 1)) \wedge$
$\wedge\ (par(c_k) \rightarrow (C(k) = c_k \wedge D(k) = d_k))))\}$

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

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

En la precondición se indica que los vectores A y B tendrán por lo menos un elemento, que siempre que en una posición de A se tenga un valor par en B se tendrá uno impar y viceversa y que los valores iniciales de A y B los representaremos mediante *a*'s y b's minúsculas con los correspondientes subíndices.

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

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

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} $\equiv \{(1 \le i \le n + 1) \wedge$
$cambiopar(A(1..n), (a_1, a_2, \ldots, a_n), B(1..n), (b_1, b_2, \ldots, b_n), i - 1)\}$

En el invariante se indica que se han hecho todos los cambios hasta la posición i – 1 incluida, es decir, estamos situados en la posición i pero todavía no se ha analizado lo que ocurre en la posición i, por ello lo que sí está analizado es hasta la posición i – 1.

(4) {Aserción intermedia} ≡ {$(1 \le i \le n) \wedge$
cambiopar(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 el while sabemos que se ha cumplido la condición del while y que i no es n + 1.

(5) {Aserción intermedia} ≡ {$(1 \le i \le n) \wedge A(i) = a_i \wedge \neg$par(A(i)) $\wedge$
cambiopar(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 el elemento A(i), que es justo el valor inicial a_i de la posición i, no es par.

(6) {Aserción intermedia} ≡ {$(1 \le i \le n) \wedge A(i) = a_i - 1 \wedge \neg$par($a_i$) $\wedge$
cambiopar(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 A(i) : = A(i) + 1; sabemos que ahora en A(i) tenemos $a_i - 1$ y que a_i no es par. 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 cambiopar seguimos poniendo i – 1.

(7) {Aserción intermedia} ≡
{$(1 \le i \le n) \wedge A(i) = a_i - 1 \wedge \neg$par($a_i$) $\wedge B(i) = b_i + 1 \wedge$
cambiopar(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 B(i) : = B(i) – 1; sabemos que ahora en A(i) tenemos $a_i - 1$ y que a_i no es par y que en B(i) tenemos $b_i + 1$. Ahora ya están hechos todos los cambios correspondientes a la posición i y por ello en el predicado cambiopar ponemos i.

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

La diferencia entre (7) y (8) es que en (7) sabemos que se ha ido por la rama **then** y por tanto podemos asegurar que $A(i) = a_i - 1 \wedge \neg$par($a_i$) $\wedge B(i) = b_i + 1$, pero en (8) no sabemos si se ha ido por la rama then o si no se ha cumplido la condición del if y por ello no podemos asegurar que se cumpla $A(i) = a_i - 1 \wedge \neg$par($a_i$) $\wedge B(i) = b_i + 1$ ya que si no se ha cumplido la condición del if, $A(i) = a_i - 1 \wedge \neg$par($a_i$) $\wedge B(i) = b_i + 1$ no será cierto.

(10) $E = n + 1 - i$

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 + 1 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.