A formal program derivation for the record

The program imptest satisfy the specification

$|[\ N: int\ \{N \geq 0\}\ ;\ b(i: 0 \leq i < N)$ <u>array</u> <u>of</u> bool
$;\ |[\ x: bool$
$\quad ;\ imptest\ \{R:\ x \equiv (\underline{A}\, i,j: 0 \leq i < j < N:\ b\,i \Rightarrow b\,j)\}$
$\quad]|$
$]|$.

<u>Legenda for the unfamiliar</u> The bracket pairs $|[\cdots]|$ delineate scopes for the variables. The environment of imptest is declared in two stages: in the outer block the constants, in the inner block the variables. The postcondition is given the name R . (End of Legenda for the unfamiliar.)

To start with, we massage the postcondition:

R
$= \{$elimination of $\Rightarrow\}$
$\quad x \equiv (\underline{A}\, i,j: 0 \leq i < j < N:\ \neg\, b\,i \vee b\,j)$
$= \{$separation of quantifications$\}$
$\quad x \equiv (\underline{A}\, j: 0 \leq j < N:\ (\underline{A}\, i: 0 \leq i < j:\ \neg\, b\,i \vee b\,j))$
$= \{\vee$ distributes over $\underline{A}\}$
$\quad x \equiv (\underline{A}\, j: 0 \leq j < N:\ (\underline{A}\, i: 0 \leq i < j:\ \neg\, b\,i) \vee b\,j)$
$= \{$naming and parameterizing the universal quantifications$\}$
$\quad x \equiv H\,N$ (0)
$\quad$ where $H\,n \equiv (\underline{A}\, j: 0 \leq j < n:\ K\,j \vee b\,j)$ (1)
$\quad\quad K\,j \equiv (\underline{A}\, i: 0 \leq i < j:\ \neg\, b\,i)$ (2)

<u>Note</u>. Like b, H and K are treated as functions, and functional application is given the greatest binding power. (End of Note.)

From (1) we deduce {property of universal quantification}

$$H(n+1) \equiv Hn \wedge (Kn \vee bn) \qquad (3a)$$

and similarly from (2)

$$K(n+1) \equiv Kn \wedge \neg bn \qquad . \qquad (3b)$$

This suggests as invariant

$$P:\quad (x \equiv Hn) \wedge (y \equiv Kn) \wedge (0 \leq n \leq N) \quad , \qquad (4)$$

for which we observe

(i) $x, y, n := true, true, 0$ establishes P {vacuously}

(ii) $n \neq N \rightarrow$

$$x, y, n := x \wedge (y \vee b(n)),\ y \wedge \neg b(n),\ n+1$$

maintains P {(3) and (4)}

(iii) $n = N \wedge P \Rightarrow R$ {(0) and (4)} .

Before rushing into program construction, however, we observe that {(0), (1), and (4)}

$$\neg x \wedge P \Rightarrow R \quad ,$$

which yields in combination with (iii)

(iv) $(n = N \vee \neg x) \wedge P \Rightarrow R$,

so the guard of (ii) can be strengthened to

$n \neq N \wedge x$, i.e. to consider, instead of (ii)

(v) $n \neq N \wedge x \rightarrow$
$$x, y, n := x \wedge (y \vee b(n)), y \wedge \neg b(n), n+1 \quad ,$$

which maintains P a forteriori. The observation that (v) can be simplified to

$$n \neq N \wedge x \rightarrow$$
$$x, y, n := y \vee b(n), y \wedge \neg b(n), n+1$$

yields for imptest the solution:

|[y: bool ; n: int
; x, y, n := true, true, 0 {P}
; <u>do</u> n≠N ∧ x →
        x, y, n := y ∨ b(n), y ∧ ¬b(n), n+1 {P}
  <u>od</u> {(n = N ∨ ¬x) ∧ P, hence}
]| {R}   ;

on account of the last term of P, termination is obvious.

* * *

In the above I have assumed the reader familiar with (de Morgan's Law and) the central theorem about repetitions —i.e. how to use invariants and how to prove termination—. In dealing with a specific program, one should concentrate on what is specific to that program.

When I showed the above derivation to Dr. H. Richards Jr., he justifiably remarked that

I had violated my own principle of developing program and correctness proof hand in hand: "You did almost all of the proof beforehand.". Proof development usually leads program derivation; what Richards had observed is to be expected with programs as small as this one.

I must draw attention to the notational benefit we derived from the introduction of the functions H and K . To the uninitiated it may seem that, at the bottom of p.0, they have just been pulled out of a hat and that, on p.1, they miraculously emerge to be precisely what we needed. To the experienced, however, their introduction is almost a routine job, and I did not want to clog this presentation of the program with heuristics (which are quite a different matter).

Furthermore I must draw attention that we did not need to mention a single special case (say: $N=0$, $N=1$, or all the $b(i)$ false). I mention this because avoidable case analyses should be avoided: they tend to lengthen the program text as well as the justifying argument and are a source of errors.

One final remark, be it of a different order. The omission of parentheses, possible thanks to the high priority of functional application, was certainly a convenience, but I observe myself becoming more and more doubtful about the convention of representing such a fundamental operation as functional application "invisibly" by just juxtaposition. Were we to indicate it explicitly, a very small symbol should be chosen in view of its high binding power. I can only think of the period: this would lead to b.i , H.n , but also to H.(n+1) . I expect I shall consider the suggestion seriously.

Austin, 8 Sep. 1984

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
AUSTIN , TX 78712-1188
U.S.A.