A formal program derivation for the record

The program `imptest` satisfy the specification

1. `N: int { N >= 0 } ; b(i: 0 <= i < N) array of bool
2. x: bool
3. imptest { R: x \equiv (\forall i, j: 0 <= i < j < N: bi \Rightarrow bj) }`  

Legend for the unfamiliar: The bracket pairs `[[...]]` 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:

\[
\text{R} = \{ \text{elimination of } \Rightarrow \} \\
\quad x \equiv (\forall i, j: 0 \leq i < j < N: \neg bi \lor bj) \\
= \{ \text{separation of quantifications} \} \\
\quad x \equiv (\forall j: 0 \leq j < N: (\forall i: 0 \leq i < j: \neg bi \lor bj)) \\
= \{ \lor \text{ distributes over } \forall \} \\
\quad x \equiv (\forall j: 0 \leq j < N: (\forall i: 0 \leq i < j: \neg bi) \lor bj) \\
= \{ \text{naming and parameterizing the universal quantifications} \} \\
\quad x \equiv \mathcal{H} N \quad (0) \\
\text{where} \quad \mathcal{H} n \equiv (\forall j: 0 \leq j < n: \neg \mathcal{K} j) \quad (1) \\
\quad \mathcal{K} j \equiv (\forall i: 0 \leq i < j: \neg bi) \quad (2)
\]
Note. 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]

$$H(n+1) \equiv Hn \land (Kn \lor bn) \quad (3a)$$

and similarly from (2)

$$K(n+1) \equiv Kn \land \neg bn \quad (3b)$$

This suggests as invariant

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

for which we observe

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

(ii) $n \neq N \Rightarrow$

$x, y, n := x \land (y \lor b(n)), y \land \neg b(n), n+1$

maintains $P$ \{and (3) and (4)\}

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

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

$$\neg x \land P \Rightarrow R,$$

which yields in combination with (iii)

(iv) $(n = N \lor \neg x) \land P \Rightarrow R,$

so the guard of (ii) can be strengthened to
n \notin N \wedge x$, i.e. to consider, instead of (ii)

(v) \( n \notin N \wedge x \rightarrow \)
\[ x, y, n := x \wedge (y \vee b(n)), y \wedge \neg b(n), n+1 \]

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

\[ n \notin N \wedge x \rightarrow \]
\[ x, y, n := y \vee b(n), y \wedge \neg b(n), n+1 \]

yields for impent the solution:

\[
\begin{align*}
\text{if} & \quad y: \text{bool}; n: \text{int} \\
& \quad ; x, y, n := \text{true}, \text{true}, 0 \{ P^3 \} \\
& \quad ; \text{do} n \notin N \wedge x \rightarrow \\
& \quad \quad x, y, n := y \vee b(n), y \wedge \neg b(n), n+1 \{ P^3 \} \\
& \quad \text{ad} \quad \{ (n = N \vee \neg x) \wedge P, \text{hence} \} \\
\text{else} & \quad \{ P^3 \} \\
\end{align*}
\]

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.