For the record: the Linear Search

Given a boolean sequence $b(i: 0 \le i)$ which is not everywhere false. The program Linear Search determines the smallest natural value x such that $b.x$. More formally

```
|[ $b(i: 0 \le i)$: array of bool $\{(\underline{E}i: 0 \le i: b.i)\}$
; |[ $x$: int
  ; Linear Search
    $\{R:\ b.x \wedge (\underline{A}i: 0 \le i < x: \neg b.i) \wedge 0 \le x\}$
  ]|
]|
```

Legenda In the outer block the constant environment is declared and what has been given about it is asserted. In the inner block the variable environment is declared and what has been given about it is asserted –when nothing has been given, as in this example, nothing is asserted–. The name of the program being specified is followed by the postcondition and the two blocks are closed. Note that variables only occur within their scope. The above is our preferred format for functional specifications. (End of Legenda.)

Aside From reviews of my books I have learned that many reviewers are annoyed by my closing brackets such as the above "(End of Legenda)", but they never give a convincing reason why I should depart from this convention, which, I think, is not without merit. So I shall stick to it. (End of Aside.)

The natural numbers being well-ordered, we conclude from the fact that equation $x: b.x$ has natural solutions that $x: b.x$ has a unique smallest solution, or, in other words, that $x: R$ has a unique solution. Let X be the solution of $x: R$, i.e. let X be given by the fact that it satisfies

(0) $\quad b.X \wedge (\underline{A}i: 0 \leq i < X: \neg b.i) \wedge 0 \leq X \qquad ,$

a definition that allows us to rephrase R as

$R: \quad x = X \qquad .$

Our first approximation to Linear Search could be

$$x := X$$

but as we have no closed expression for X, this approximation is much too crude.

For our next approximation we look for an initialization followed by a repetition, such that the dependence on b is confined to the latter. Since $0 \leq X$ is the only term in (0) in which b does not occur, this is then the only term we may use in justifying the initialization. That is, we are looking for an invariant that

(i) is a generalization of R

(ii) can be initialized on account of $0 \leq X \qquad .$

I suggest as invariant P, given by

$P: \qquad 0 \leq x \leq X$

which leads to our second approximation

```
{0 ≤ X} x := 0 {P}
; do x ≠ X → x := x+1 od {R}
```

for which $X-x$ serves for the termination argument since $P \Rightarrow X-x \geq 0$, and $X-x$ is decreased in each iteration.

Finally we deduce from P and (0) the lemma

$$P \Rightarrow (x \neq X \equiv \neg b.x) \quad ,$$

which allows us to transform the second solution into our final solution for Linear Search:

```
{0 ≤ X} x := 0 {P}
; do ¬b.x → x := x+1 od {R}    .
```

* * *

The validity of the above solution is known —for more than a decade— as the Linear Search Theorem. It is a very simple theorem: it is as unsurprising as it is useful. In the folklore, it is known as "when looking for a minimum value satisfying some criterion, search in increasing order", but I would like to point out that we should be careful with such ellipses: instead of defining x as the minimum natural value x such that $b.x$, we could have defined it as the <u>maximum</u> value x such that $(\underline{A}i: 0 \leq i < x: \neg b.i)$.

I recorded the above derivation because it differs from the one I gave formerly in that X —the final value of x— occurs explicitly in the whole argument. My former argument used for the invariant the (equivalent) formulation

$$(\underline{A}i: 0 \leq i < x: \neg b.i) \wedge 0 \leq x \quad ,$$

obtained from R by omission of the first conjunct. It contained an explicit appeal to

$$(\underline{A}i: 0 \leq i < x+1: \neg b.i) \quad \equiv \quad (\underline{A}i: 0 \leq i < x: \neg b.i) \wedge \neg b.x$$

and lacked the stepping stone of our second approximation.

I prefer the derivation given here, even though it might involve a little bit more formal labour had I proved the lemma explicitly. As the lemma is independent of the algorithm, the new version is a bit further disentangled. Hence my preference.

I am endebted to W.H.J. Feijen for pointing out to me the existence of this alternative derivation.

Austin, 1 Sep. 1985

prof. dr. Edsger W. Dijkstra
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712-1188
United States of America