(Bounded) Linear Search by Tail Recursion

Tom Verhoeff*

18 February 2007 (first draft)

1 Introduction

The *Linear Search*, as presented in [1, 4], provides a solution for the following specification:

$$
\begin{array}{l}
|[\ \mathbf{con}\ b\colon \mathbf{array}\ [0..\infty)\ \mathbf{of}\ \mathit{int} \\
\ ;\ \mathbf{var}\ x\colon \mathit{int} \\
\triangleright \\
\quad \{\ \text{pre: } (\exists\, i : 0 \le i : b.i)\ \} \\
\quad S \\
\quad \{\ \text{post: } x = (\min i : 0 \le i\ \wedge\ b.i : i)\ \} \\
]|
\end{array}
$$

The derivation in [1, 4] proceeds by rewriting the postcondition into a conjunction:

$$0 \le x\ \wedge\ b.x\ \wedge\ (\forall\, i : 0 \le i : \neg b.i) \tag{1}$$

The middle conjunct is taken as the negation of the guard (because it is hard to initialize as part of an invariant), whereas the other two conjuncts are taken as invariant. The precondition is only needed to guarantee termination.

The *Bounded Linear Search*, as presented in [2, 4], provides a solution for the following specification:

$$
\begin{array}{l}
|[\ \mathbf{con}\ b\colon \mathbf{array}\ [0..N)\ \mathbf{of}\ \mathit{int} \\
\ ;\ \mathbf{var}\ x\colon \mathit{int} \\
\triangleright \\
\quad \{\ \text{pre: } b.N \text{ (thought element) }\} \\
\quad S \\
\quad \{\ \text{post: } x = (\min i : 0 \le i \le N\ \wedge\ b.i : i)\ \} \\
]|
\end{array}
$$

where $b.N$ may not be inspected by S.

*Software Engineering & Technology, Dept. of Math. & CS, Technische Universiteit Eindhoven, Den Dolech 2, 5612 AZ EINDHOVEN, The Netherlands. E-mail: T.Verhoeff@tue.nl

2 Linear Search by Tail Recursion

The right-hand side of the postcondition of the Linear Search can be generalized by replacing the constant 0 by a variable. Therefore, we define function F by

$$F.m \quad = \quad (\min i : m \leq i \ \wedge \ b.i : i) \qquad \text{for } 0 \leq m. \tag{2}$$

Using F, the postcondition can be expressed as

$$x \quad = \quad F.0 \tag{3}$$

To develop a tail recursive solution, we derive

$$\begin{array}{ll} & F.m \\ = & \quad \{ \text{ def. } F \ \} \\ & (\min i : m \leq i \ \wedge \ b.i : i) \\ = & \quad \{ \text{ case distinction on } b.m \ \} \\ & \left[\begin{array}{ll} m & \text{if } b.m \\ F.(m+1) & \text{if } \neg b.m \end{array} \right. \end{array}$$

Thus, we obtained the recurrence equations:

$$F.m \quad = \quad m \qquad \text{if } b.m \tag{4}$$

$$F.m \quad = \quad F.(m+1) \qquad \text{if } \neg b.m \tag{5}$$

Consequently, if we take as (tail) invariant

$$\begin{array}{lll} P_0 & : & 0 \leq x \\ P_1 & : & F.x = F.0 \end{array}$$

then

- Initialization is realized by $x := 0$.
- Finalization: Concerning the guard we calculate from the postcondition:

  $$\begin{array}{ll} & x = F.0 \\ \equiv & \quad \{ \ F.x = F.0 \text{ on account of } P_1 \ \} \\ & x = F.x \\ \Leftarrow & \quad \{ \ (4) \ \} \\ & b.x \end{array}$$

  Hence, $\neg b.x$ suffices as guard.
- Invariance: if $\neg b.x$, then $x := x + 1$ maintains $P_{0..1}$ on account of (5).

Termination is guaranteed by the precondition. This yields the following (well-known) Linear Search program:

$$\begin{array}{l} \quad x := 0 \\ ; \ \mathbf{do} \ \neg b.x \rightarrow x := x + 1 \ \mathbf{od} \end{array}$$

3 Bounded Linear Search by Tail Recursion

The right-hand side of the postcondition of the Bounded Linear Search can be generalized by replacing the constants 0 and N by variables (*why both?*). Therefore, we define function G by

$$G.m.n \quad = \quad (\min i : m \leq i \leq n \ \wedge \ b.i : i) \qquad \text{for } 0 \leq m \leq n \leq N. \tag{6}$$

Using G, the postcondition can be expressed as

$$x \quad = \quad G.0.N \tag{7}$$

Note that on account of the precondition $b.N$, we have

$$G.0.N \leq N < \infty \tag{8}$$

To develop a tail recursive solution, we derive

$$\begin{array}{ll} & G.m.n \\ = & \quad \{ \text{ def. } G \ \} \\ & (\min i : m \leq i \leq n \ \wedge \ b.i : i) \\ = & \quad \{ \bullet \ m = n \text{ and } G.m.n < \infty \ \} \\ & m \end{array}$$

In case $m < n$, we derive

$$\begin{array}{ll} & G.m.n \\ = & \quad \{ \text{ def. } G \ \} \\ & (\min i : m \leq i \leq n \ \wedge \ b.i : i) \\ = & \quad \{ \text{ case distinction on } b.m \ \} \\ & \left[\begin{array}{ll} G.m.m & \text{if } b.m \\ G.(m+1).n & \text{if } \neg b.m \end{array} \right. \end{array}$$

Thus, we obtained the recurrence equations:

$$G.m.n \quad = \quad m \qquad \text{if } m = n \text{ and } G.m.n < \infty \tag{9}$$

$$G.m.n \quad = \quad G.m.m \qquad \text{if } m < n \text{ and } b.m \tag{10}$$

$$G.m.n \quad = \quad G.(m+1).n \qquad \text{if } m < n \text{ and } \neg b.m \tag{11}$$

Consequently, if we take as (tail) invariant

$$\begin{array}{lll} P_0 & : & 0 \leq x \leq y \leq N \\ P_1 & : & G.x.y = G.0.N \end{array}$$

then

- Initialization is realized by $x, y := 0, N$

- Finalization: Concerning the guard we calculate from the postcondition

$$
\begin{array}{ll}
 & x = G.0.N \\
\equiv & \quad \{ \; G.x.y = G.0.N \text{ on account of } P_1\text{, and (8) using } P_1 \; \} \\
 & x = G.x.y \;\wedge\; G.x.y < \infty \\
\Leftarrow & \quad \{ \; (9) \; \} \\
 & x = y
\end{array}
$$

  Hence, the guard $x \neq y$ suffices.

- Invariance: Because of P_0 and guard $x \neq y$, we have $0 \leq x < y \leq N$ as precondition of the loop body. If $b.x$, then $y := x$ maintains P_1 on account of (10). If $\neg b.x$, then $x := x+1$ maintains P_1 on account of (11). Invariance of P_0 is now standard.

Termination is guaranteed by variant function $y - x$. This yields the following (well-known) Bounded Linear Search program:

$$
\begin{array}{l}
\quad x, y := 0, N \\
; \; \textbf{do} \; x \neq y \rightarrow \\
\qquad \textbf{if} \quad b.x \rightarrow y := x \\
\qquad [] \; \neg b.x \rightarrow x := x + 1 \\
\qquad \textbf{fi} \\
\quad \textbf{od}
\end{array}
$$

4 Conclusion

These derivations show —once more [3]— the merits of tail invariants.

References

[1] W. Feijen. *The Linear Search, and some more*. WF116, 1990.
Internet: `http://www.mathmeth.com/wf/files/wf1xx/wf116.pdf`.

[2] W. Feijen. *The Bounded Linear Search*. WF118, 1990.
Internet: `http://www.mathmeth.com/wf/files/wf1xx/wf118.pdf`.

[3] W. Feijen. *On the ascendingness-test and on tail invariants*. WF155, 1992.
Internet: `http://www.mathmeth.com/wf/files/wf1xx/wf155.pdf`.

[4] A. Kaldewaij. *Programming: The Derivation of Algorithms*. Prentice Hall, 1990.