Chapter 6: Searching

Linear Searching

Consider the following programming problem.

$$
\begin{array}{l}
|[\\
\mathbf{var}\ x : \mathit{int}; \\
\{(\exists i : 0 \le i : b.i)\} \\
\mathit{Linear\ Search} \\
\{x = (\mathbf{min}\ i : 0 \le i \wedge b.i : i)\} \\
]|.
\end{array}
$$

What is a possible invariant?

The post-condition can be rewritten as

$$R:\ 0 \le x \wedge b.x \wedge (\forall i : 0 \le i < x : \neg b.i)$$

So a possible invatiant is obtained by taking a conjunct:

$$P:\ 0 \le x \wedge (\forall i : 0 \le i < x : \neg b.i)$$

which is initialized by $x := 0$.

Investigation of $x := x + 1$ leads to

$$
\begin{array}{ll}
 & P(x := x + 1) \\
\equiv & \quad \{ \text{ definition of } P \text{ } \} \\
 & 0 \leq x + 1 \wedge (\forall i : 0 \leq i < x + 1 : \neg b.i) \\
\Leftarrow & \quad \{ \text{ heading for } P \text{ } \} \\
 & 0 \leq x \wedge (\forall i : 0 \leq i < x + 1 : \neg b.i) \\
\equiv & \quad \{ \text{ split off } i = x \text{ } \} \\
 & 0 \leq x \wedge (\forall i : 0 \leq i < x : \neg b.i) \wedge \neg b.x \\
\equiv & \quad \{ \text{ definition of } P \text{ } \} \\
 & P \wedge \neg b.x
\end{array}
$$

The final program for linear search:

$$
\begin{array}{l}
|[\\
\textbf{var } x : int;\\
\{(\exists i : 0 \leq i : b.i)\}\\
x := 0;\\
\textbf{do } \neg b.x \rightarrow x := x + 1 \textbf{ od}\\
\{x = (\textbf{min } i : 0 \leq i \wedge b.i : i)\}\\
]|.
\end{array}
$$

Does this program terminate? Why?

Bounded Linear Search

The specification of the problem:

$|[$

$\mathbf{con}\ N : int\ \{N \geq 0\}; b : \mathbf{array}\ [0..N)\ \mathbf{of}$ bool;

$\mathbf{var}\ x : int;$

bounded linear search

$\{x = (\mathbf{max}\ i : 0 \leq i \leq N \wedge (\forall j : 0 \leq j < i : \neg b.j) : i)\}$

$]|$

Could we use the invariant

$$0 \le x \le N \wedge (\forall i : 0 \le i < x : \neg b.i)$$

to obtain the following program?

$$x := 0$$
$$\mathbf{do}\ \neg b.x \wedge x \neq N \rightarrow x := x + 1\ \mathbf{od}$$

No, since N does not belong to the domain of b and $x = N$ is not excluded by the invariant. But we can define as invariant:

$$\begin{aligned} P_0 &: \; 0 \leq x \leq N \wedge (\forall i : 0 \leq i < x : \neg b.i) \wedge b.y \\ P_1 &: \; x \leq y \leq N \end{aligned}$$

Then

$$P_0 \wedge P_1 \wedge x \neq y \rightarrow 0 \leq x < N,$$

so $b.x$ may occur in the statement of the repetition.

The final program:

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

Binary Search

Consider the problem:

$$
\begin{array}{l}
|[\\
\mathit{keycon}\ N, A : \mathit{int}\ \{N \geq 1\};\ f : \mathbf{array}\ [0..N]\ \mathbf{of}\ \mathit{int}\ \{f.0 \leq A < f.N\}; \\
\mathbf{var}\ x : \mathit{int}; \\
\mathit{binary\ search} \\
\{f.x \leq A < f.(x+1)\} \\
]|
\end{array}
$$

Could we do better than linear search?

From the post-condition:

$$R: \; f.x \leq A < f.(x+1)$$

we generalize $x+1$ to y, and define as invariants:

$$\begin{aligned} P_0 : &\quad f.x \leq A < f.y \\ P_1 : &\quad 0 \leq x < y \leq N \end{aligned}$$

which is established by $x, y := 0, N$. And we may choose guard as

$$x + 1 \neq y.$$

What is a suitable bound function?

A straightforward bound function is $y - x$. To decrease it, we may choose a h such that $x < h < y$, and both

$$x := h$$

and

$$y := h$$

will decrease $y - x$.

How to keep invariants?

We investigate the effects of $x := h$ and $y := h$ on the invariants.

$$
\begin{array}{ll}
 & P_0(x := h) \\
\equiv & \quad \{ \text{ substitution } \} \\
 & f.h \leq A < f.y \\
\Leftarrow & \quad \{ \text{ definition of } P_0 \ \} \\
 & P_0 \wedge f.h \leq A
\end{array}
$$

$$
\begin{array}{ll}
 & P_0(y := h) \\
\equiv & \quad \{ \text{ substitution } \} \\
 & f.x \leq A < f.h \\
\Leftarrow & \quad \{ \text{ definition of } P_0 \ \} \\
 & P_0 \wedge A < f.h
\end{array}
$$

This leads to

$$
\begin{array}{l}
\{f.0 \leq A < f.N\} \\
|[\\
\textbf{var } y : \mathit{intl} \\
x, y := 0, N; \\
\{\mathit{invarinat}\text{: } P_0 \wedge P_1,\ \mathit{bound}\text{: } y - x\} \\
\textbf{do } x + 1 \neq y \rightarrow \\
\qquad |[\\
\qquad \textbf{var } h : \mathit{int}; \\
\qquad \underline{\mathit{establish}\ x < h < y}; \\
\qquad \textbf{if } f.h \leq A \rightarrow x := h \\
\qquad \ [\,]\ A \leq f.h \rightarrow y := h \\
\qquad \textbf{fi} \\
\qquad]| \\
\textbf{od} \\
]|
\end{array}
$$

How to establish the underline?

So we obtain

$$
\begin{array}{l}
x, y := 0, N; \\
\textbf{do } x + 1 \neq y \rightarrow \\
\quad\quad |[\\
\quad\quad \textbf{var } h : \mathit{int}; \\
\quad\quad h = (x + y) \textbf{ div } 2; \\
\quad\quad \textbf{if } f.h \leq A \rightarrow x := h \\
\quad\quad \ [] \ A \leq f.h \rightarrow y := h \\
\quad\quad \textbf{fi} \\
\quad\quad]| \\
\textbf{od}
\end{array}
$$

Two remarks:

- From the fact that $0 < h < N$, we know that $f.0$ and $f.N$ are not inspected.
- Precondition is only used for the initialization of x and y.

Exercise: Derive a program for

$$\begin{array}{l}
|[\\
\textbf{con}\ N, A : \mathit{int}\ \{N \geq 1\};\ f : \textbf{array}\ [0..N]\ \textbf{of}\ \mathit{int}\ \{f.0 \leq A < f.N\}; \\
\{\underline{(\forall i, j : 0 \leq i \leq j < N : f.i \leq f.j)}\} \\
\textbf{var}\ r : \mathit{bool}; \\
S \\
\{r \equiv (\exists i : 0 \leq i < N : f.i = A)\} \\
]|
\end{array}$$

Hint: Consider the post-condition as

$$R:\ 0 \leq x < N \wedge (f.x \leq A < f.(x+1) \wedge A < f.0)$$

while virtually assuming $f.N = \infty$.

Program for "the binary search":

$$
\begin{array}{l}
x, y := 0, N; \\
\mathbf{do}\ x + 1 \neq y \rightarrow \\
\qquad |[\\
\qquad \mathbf{var}\ h : int; \\
\qquad h = (x + y)\ \mathbf{div}\ 2; \\
\qquad \mathbf{if}\ f.h \leq A \rightarrow x := h \\
\qquad\ [\,]\ A \leq f.h \rightarrow y := h \\
\qquad \mathbf{fi} \\
\qquad]| \\
\mathbf{od}
\end{array}
$$

Square Root Problem

Consider writing a program for

$$\begin{array}{l} |[\\ \mathbf{con}\ N : \mathit{int}\ \{N \geq 0\}; \\ \mathbf{var}\ x : \mathit{int}; \\ \mathit{square\ root} \\ \{x^2 \leq N \land (x+1)^2 > N\} \end{array}$$

whose time complexity is $\mathcal{O}(\log N)$.

Idea: Make use of the binary search method with the invariant of

$$P: \quad 0 \le x < y \le N + 1 \wedge x^2 \le N < y^2$$

and the guard of $x + 1 \neq y$, and we can obtain the following program.

```
|[
var y : int;
x, y := 0, N + 1;
do x + 1 ≠ y →
    |[
    var h : int;
    h = (x + y) div 2;
    if h * h ≤ N → x := h
     [] N < h * h → y := h
    fi
    ]|
od
```