Chapter. The Bounded Linear Search Theorem.

An example.

Given an array f[0..N) of int where $1 \le N$, and a target int value X. We are asked to find the smallest index i in $0 \le i < N$ where $f.i = X$. However we have no guarantee that X is actually present in f.

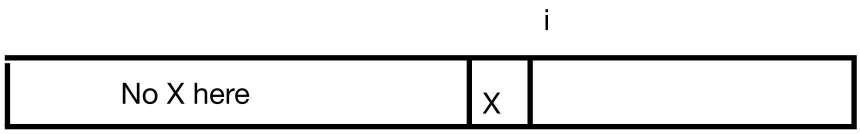
Precondition

f[0..N) has values

Postcondition.

But the postcondition requires us to do a little work. There are two possibilities; the value is present or the value is absent. We look at each in turn.

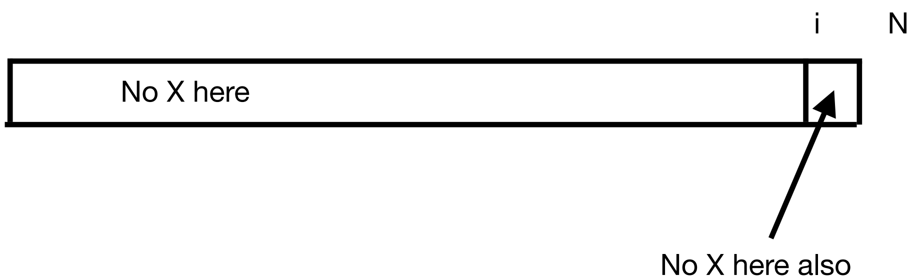
The value is present in f, the postcondition is the same as for the Linear Search



$$\langle \forall j \ : 0 \le j < i : f.j \ne X \rangle \wedge f.i = X$$

The value is absent, the postcondition could be written like this

$$\langle \forall j \ : 0 \le j < N : f.j \ne X \rangle$$



but we choose to write it in a slightly different way as

$$\langle \forall j \ : 0 \le j < i : f.j \ne X \rangle \wedge \ f.i \ \ne X \wedge i = N\text{-}1$$

We can now combine these together as

$$\langle \forall j \ : 0 \le j < i : f.j \ne X \rangle \wedge (f.i = X \vee (f.i \ \ne X \wedge i = N\text{-}1))$$

Applying a familiar law[1] allows us to simplify this to get our overall postcondition.

Post : $\langle \forall j : 0 \le j < i : f.j \ne X \rangle \wedge (f.i = X \vee i = N\text{-}1)$

Domain model.

* (0) $C.i \equiv \langle \forall j : 0 \le j < i : f.j \ne X \rangle$, $0 \le i \le N\text{-}1$

Consider

$$
\begin{array}{ll}
 & C.0 \\
= & \quad \{(0)\} \\
 & \langle \forall j : 0 \le j < 0 : f.j \ne X \rangle \\
= & \quad \{ \text{ Empty range } \} \\
 & Id\wedge
\end{array}
$$

which gives us

\- (1) $C.0 \equiv True$

Consider

$$
\begin{array}{ll}
 & C.(i+1) \\
= & \quad \{(0)\} \\
 & \langle \forall j : 0 \le j < i+1 : f.j \ne X \rangle \\
= & \quad \{ \text{ Split off } j=i \text{ term } \} \\
 & \langle \forall j : 0 \le j < i : f.j \ne X \rangle \wedge f.i \ne X \\
= & \quad \{(0)\} \\
 & C.i \wedge f.i \ne X
\end{array}
$$

which gives us

\- (2) $C.(i+1) \equiv C.i \wedge f.i \ne X$, $0 \le i < N\text{-}1$

Rewrite postcondition.

Post: $C.i \wedge (f.i = X \vee i = N\text{-}1)$

Choose invariants.

P0: $C.i$
P1: $0 \le i \le N\text{-}1$

[1] We are appealing to a compliment law $[X \vee (\neg X \wedge Y) \equiv X \vee Y]$

Guard.

$f.i \neq X \wedge i \neq N\text{-}1$

Establish invariants.

$i := 0$

Variant.[2]

$(K \downarrow N\text{-}1) - i$

Loop body.

$$
\begin{array}{ll}
 & (i := i+1).P0 \\
= & \quad \{ \text{ Text Substitution } \} \\
 & C.(i+1) \\
= & \quad \{(2)\} \\
 & C.i \wedge f.i \neq X \\
= & \quad \{ \text{ P0 \& guard } \} \\
 & \text{True}
\end{array}
$$

Finished algorithm.

```
 i := 0
;do f.i ≠ X ∧ i ≠ N-1 →
        i := i+1
 od

;if  f.i = X → "found case"
 []  f.i ≠ X ∧ i = N-1 → "not found case"
 fi
```

This problem is an example of a more general type of problem called a Bounded Linear Search. We will now look at the more abstract case and construct the generic algorithm.

---

[2] K is the smallest index where f.K = X, should such a place exist

Generic bounded linear search.

Given a *finite, ordered, non-empty domain* f[α..β) of some type, and a predicate Q defined on the individual elements of the domain, find the smallest index i where $\alpha \leq i < \beta$ where Q.(f.i) holds true, if such a place exists.

Postcondition $\langle \forall j : \alpha \leq j < i : \neg Q.(f.j) \rangle \wedge (Q.(f.i) \vee i = \beta\text{-}1)$

Domain model.

* (0) C.i $\equiv$ $\langle \forall j : \alpha \leq j < i : \neg Q.(f.j) \rangle$, $\alpha \leq i \leq \beta\text{-}1$

Consider

$$
\begin{aligned}
& C.\alpha \\
= \quad & \{(0)\} \\
& \langle \forall j : \alpha \leq j < \alpha : \neg Q.(f.j) \rangle \\
= \quad & \{ \text{ Empty range } \} \\
& Id\wedge
\end{aligned}
$$

which gives us

- (1) C. α $\equiv$ True

Consider

$$
\begin{aligned}
& C.(i+1) \\
= \quad & \{(0)\} \\
& \langle \forall j : \alpha \leq j < i+1 : \neg Q.(f.j) \rangle \\
= \quad & \{ \text{ Split off } j=i \text{ term } \} \\
& \langle \forall j : \alpha \leq j < i : \neg Q.(f.j) \rangle \wedge \neg Q.(f.i) \\
= \quad & \{(0)\} \\
& C.i \wedge \neg Q.(f.i)
\end{aligned}
$$

which gives us us

- (2) C.(i+1) $\equiv$ C.i $\wedge \neg$Q.(f.i) , $\alpha \leq i < \beta\text{-}1$

Rewrite postcondition.

Post: C.i $\wedge$ (Q.(f.i) $\vee$ i = β-1)

Choose invariants.

P0: C.i
P1: $\alpha \leq i \leq \beta\text{-}1$

Guard.

$\neg Q.(f.i) \wedge i \neq \beta\text{-}1$

Establish invariants.

$i := \alpha$

Variant.[3]

$(K \downarrow \beta\text{-}1) - i$

Loop body.

$(i := i+1).P0$
$=$ { Text Substitution }
$C.(i+1)$
$=$ {(2)}
$C.i \wedge \neg Q.(f.i)$
$=$ { P0 & guard }
True

Finished algorithm.

```
 i := 0
;do ¬Q.(f.i) ∧ i ≠ β-1 →
        i := i+1
 od

;if Q.(f.i) → "found case"
 [] f ¬Q.(f.i) ∧ i = β-1 → "not found case"
 fi
```

[3] K is the smallest index where f.K = X, should such a place exist

Two useful patterns.

There are two commonly occurring shapes for the Bounded Linear Search and it is useful to be familiar with them. Given the finite, non-empty, ordered domain $f[\alpha..\beta)$ as above, and a predicate Q defined on the individual elements of the domain, we can ask

(i) Does Q hold true everywhere

$$\begin{array}{l} \langle \forall j : \alpha \le j < i : Q.(f.j) \rangle \wedge \\ \qquad ((\neg Q.(f.i) \wedge \text{“no it doesn’t”}) \\ \qquad\qquad \vee \\ \qquad (i = \beta\text{-}1) \wedge Q.(f.i) \wedge \text{“yes it does”})) \end{array}$$

(ii) Does Q hold true anywhere

$$\begin{array}{l} \langle \forall j : \alpha \le j < i : \neg Q.(f.j) \rangle \wedge \\ \qquad ((Q.(f.i) \wedge \text{“yes it does”}) \\ \qquad\qquad \vee \\ \qquad (i = \beta\text{-}1) \wedge \neg Q.(f.i) \wedge \text{“no it doesn’t”})) \end{array}$$