Chapter 11a: The Generic Bounded Linear Search.

General Solution.

This problem is just one instance of a set of problems called the Bounded Linear Searches. We will now describe this family of problems and construct the generic solution.

Suppose we are given a finite, non-empty, ordered domain, $f[\alpha..\beta)$ and a predicate Q defined on the elements of f. We are to determine whether Q holds true at at least one point in the domain. Of course there is the possibility that it may not hold anywhere.

Our postcondition is

$$\langle \forall j : \alpha \le j < i : \neg Q.(f.j) \rangle \wedge (Q.(f.i) \vee i = \beta - 1)$$

As usual we develop our model

* (0) $C.i \quad \equiv \quad \langle \forall j : \alpha \le j < i : \neg Q.(f.j) \rangle \qquad , \alpha \le i \le \beta$

Appealing to the empty range and associativity we get the following theorems

Consider.

$C.\alpha$

= {(0) in model }

$\langle \forall j : \alpha \le j < \alpha : \neg Q.(f.j) \rangle$

= { empty range }

true

- (1) $C.\alpha \quad \equiv \quad$ true

Consider

$C.(i+1)$

= {(0) in model }

$\langle \forall j : \alpha \le j < i+1 : \neg Q.(f.j) \rangle$

= { split off j = i term}

$\langle \forall j : \alpha \le j < i : \neg Q.(f.j) \rangle \wedge \neg Q.(f.i) \rangle$

= {(0) in model}

$C.n \wedge \neg Q.(f.i)$

- (2) $C.(i+1) \equiv C.i \wedge \neg Q.(f.i) \qquad , \alpha \le i < \beta$

Rewrite postcondition in terms of model.

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

Choose Invariants.

We choose as our invariants

P0: $C.i$

P1: $\alpha \leq i < \beta$

Termination.

We note that

$$P0 \wedge P1 \wedge (Q.(f.i) \vee i = \beta - 1) \Rightarrow \text{Post}$$

Establish Invariants.

Our model (1) shows us that we can establish P0 by the assignment

$$i := \alpha$$

This also establishes P1.

Guard

We choose our loop guard to be

$$B : \neg Q.(f.i) \wedge i \neq \beta - 1$$

Calculate Loop body.

Decreasing the variant by the assignment $i := i+1$ is a standard step and maintains P1. Let us se what effect it has on P0

$(i := i+1).\ P0$

= {textual substitution}

$C.(i+1)$

= {(2) above}

$C.i \wedge \neg Q.(f.i)$

= {Remember $P0 \wedge \neg Q.(f.i) \wedge i \neq \beta - 1$ at start of loop body}

true

Finished Program.

So our finished program is

```
i := α
; do ¬Q.(f.i) ∧ i ≠ β - 1→

          i := i+1

  od
{C.i ∧ (Q.(f.i) ∨ i = β - 1)}
```

We can now determine whether Q holds anywhere and communicate this with the user by adding the following if..fi after the loop

```
if Q.(f.i)               → write('X found at position', i )
[] i = β -1 ∧ ¬Q.(f.i)   → write('X is not in f')
fi
```

This is called the Bounded Linear Search Theorem.

Important note.

There are 2 important variations of the Bounded Linear Search. We illustrate them below. Given a finite, non-empty, ordered domain f[α..β) and a predicate Q defined on the elements of f.

Does Q hold true *everywhere*

$$\langle \forall j : \alpha \leq j < i : Q.(f.j) \rangle \wedge ((\neg Q.(f.i) \wedge \text{"no it doesn't"}) \vee (Q.(f.i) \wedge i = \beta - 1 \wedge \text{"yes it does"}))$$

Does Q hold true *anywhere*

$$\langle \forall j : \alpha \leq j < i : \neg Q.(f.j) \rangle \wedge ((Q.(f.i) \wedge \text{"yes it does"}) \vee (\neg Q.(f.i) \wedge i = \beta - 1 \wedge \text{"no it doesn't"}))$$