Chapter 6: Searching (Cont.)

Searching by Elimination

Given:

- a finite set W
- a boolean function S on W, such that $S.w$ holds for some $w \in W$

derive a program with the following post-condition.

$$S.x$$

Note: if S is identified with $\{x \in W \mid S.x\}$, then the post-condition can be written as

$$R: \; S \cap \{x\} \neq \emptyset.$$

What is a suitable invariant?

Generalization of the post-condition:

$$R: \; S \cap \underline{\{x\}} \neq \{\}.$$

gives as invariant

$$P: \; S \cap \underline{V} \neq \emptyset \wedge \underline{V \subseteq W}$$

which is established by $V := W$.

This leads to the following program scheme:

$\{S \cap W \neq \emptyset\}$
$V := W$;
$\{$invariant: $S \cap V \neq \emptyset \wedge V \subseteq W$, bound: $|V|\}$
do $|V| \neq 1 \rightarrow$
    decrease $|V|$ under invariance of P
od;
$x :=$ the unique element of V

Searching by elimination:

$\{S \cap W \neq \emptyset\}$
$V := W;$
$\{$invariant: $S \cap V \neq \emptyset \wedge V \subseteq W$, bound: $|V|\}$
do $|V| \neq 1 \rightarrow$
  choose a and b in $|V|$ such that $a \neq b$
  $\{a \in V \wedge b \in V \wedge a \neq b \wedge S \cap V \neq \emptyset\}$
  if $B_0 \rightarrow V := \underline{V \backslash \{a\}}$
  [] $B_1 \rightarrow V := \underline{V \backslash \{b\}}$
  fi
od;
$x :=$ the unique element of V

What are B_0 and B_1?

B_0 and B_1 should be the conditions keeping invariants, i.e.,

$$B_0 \Rightarrow (S \cap V \neq \emptyset \Rightarrow S \cap (V \backslash \{a\}) \neq \emptyset)$$
$$B_1 \Rightarrow (S \cap V \neq \emptyset \Rightarrow S \cap (V \backslash \{b\}) \neq \emptyset)$$

How to calculate out B_0 and B_1?

From the calculation

$$
\begin{array}{ll}
 & S \cap V \neq \emptyset \;\Rightarrow\; S \cap (V \backslash \{a\}) \neq \emptyset) \\
\equiv & \quad \{ \; a \in V \; \} \\
 & S.a \vee (S \cap (V \backslash \{a\})) \neq \emptyset \;\Rightarrow\; S \cap (V \backslash \{a\}) \neq \emptyset) \\
\equiv & \quad \{ \text{ predicate calculus } \} \\
 & S.a \;\Rightarrow\; S \cap (V \backslash \{a\}) \neq \emptyset \\
\Leftarrow & \quad \{ \; b \in V \backslash \{a\} \; \} \\
 & S.a \Rightarrow S.b \\
\equiv & \quad \{ \text{ prediate calculus } \} \\
 & \neg S.a \vee S.b
\end{array}
$$

we may have

$$B_0 = \neg S.a \vee S.b.$$

And similarly we may have

$$B_1 = \neg S.b \vee S.a.$$

So we obtain the program:

$\{S \cap W \neq \emptyset\}$

$V := W;$

$\{$invariant: $S \cap V \neq \emptyset \wedge V \subseteq W$, bound: $|V|\}$

do $|V| \neq 1 \rightarrow$

  choose a and b in $|V|$ such that $a \neq b$

  $\{a \in V \wedge b \in V \wedge a \neq b \wedge S \cap V \neq \emptyset\}$

  if $\neg S.a \vee S.b \rightarrow V := \underline{V \backslash \{a\}}$

  $[]\ \neg S.b \vee S.a \rightarrow V := \underline{V \backslash \{b\}}$

  fi

od;

$x :=$ the unique element of V

A special case when $W = [0..N]$. We may choose V can be represented by two integers a and b as $[a..b]$, and the program becomes

$$
\begin{array}{l}
\{(\exists i : 0 \le i \le N : S.i)\} \\
a, b := 0, N; \\
\mathbf{do}\ a \ne b \to \\
\qquad \mathbf{if}\ \neg S.a \vee S.b \to a := a + 1 \\
\qquad\ [\,]\ \neg S.b \vee S.a \to b := b - 1 \\
\qquad \mathbf{fi} \\
\mathbf{od}; \\
x := a;
\end{array}
$$

Application 1:

Derive a program that satisfies

$$
\begin{array}{l}
|[\\
\textbf{con } N : int\ \{N \geq 0\}; b : \textbf{array } [0..N] \textbf{ of } \mathrm{int}; \\
\textbf{var } x : int; \\
max\ location \\
\{0 \leq x \leq N \wedge f.x = (\textbf{max } i : 0 \leq i \leq N : f.i)\} \\
]| \\
\{S.x\}
\end{array}
$$

How to make use of "Searching by Elimination" to solve this problem?

The post-condition can be rewritten as

$$R:\ 0 \leq x \leq N \wedge (\forall i : 0 \leq i \leq N : f.i \leq f.x)$$

and we can define S as

$$S.x \equiv (\forall i : 0 \leq i \leq N : f.i \leq f.x)$$

What is a sufficient condition for $\neg S.a \vee S.b$?

Since

$$
\begin{array}{ll}
 & \neg S.a \vee S.b \\
\equiv & \quad \{ \text{ predicate calculus } \} \\
 & S.a \Rightarrow S.b \\
\equiv & \quad \{ \text{ definition of } S \text{ } \} \\
 & (\forall i : 0 \leq i \leq N : f.i \leq f.a) \;\Rightarrow\; (\forall i : 0 \leq i \leq N : f.i \leq f.b) \\
\Leftarrow & \quad \{ \text{ transitivity of } \leq \text{ } \} \\
 & f.a \leq f.b
\end{array}
$$

we have

$$f.a \leq f.b \;\Rightarrow\; \neg S.a \vee S.b$$

Similarly, we can derive

$$f.b \leq f.a \;\Rightarrow\; \neg S.b \vee S.a$$

Our solution:

$$
\begin{array}{l}
\mathbf{var}\ a, b : int; \\
a, b := 0, N; \\
\mathbf{do}\ a \neq b \rightarrow \\
\qquad \mathbf{if}\ f.a \leq f.b \rightarrow a := a + 1 \\
\qquad\ \ [\,]\ f.b \leq f.a \rightarrow b := b - 1 \\
\qquad \mathbf{fi} \\
\mathbf{od}; \\
x := a;
\end{array}
$$

Application 2: The Celebrity Problem

Design a program to compute a *celebrity* among $N+1$ persons. A person is a celebrity if he is known by everyone but does not know anyone.

$$
\begin{array}{l}
|[\\
\textbf{con}\ N : \mathit{int}\ \{N \geq 0\};\ k : \textbf{array}\ [0..N] \times [0..N]\ \textbf{of}\ \mathit{bool};\\
\{(\exists i : 0 \leq i \leq N : (\forall j : j \neq i : k.j.i \wedge \neg k.i.j))\}\\
\textbf{var}\ x : \mathit{int};\\
\text{celebrity}\\
\{0 \leq x \leq N \wedge (\forall j : j \neq x : k.j.x \wedge \neg k.x.j))\}\\
]|
\end{array}
$$

Here $k.i.j$ denotes i knows j.

We could consider the set W as $[0..N]$. What is S?

We choose

$$S.x \equiv (\forall j : j \neq x : k.j.x \wedge \neg k.x.j)$$

We then derive

$$\begin{array}{ll}
 & \neg S.a \vee S.b \\
\Leftarrow & \quad \{ \text{ predicate calculus } \} \\
 & \neg S.a \\
\equiv & \quad \{ \text{ definition of } S \ \} \\
 & \neg(\forall j : j \neq a : k.j.a \wedge \neg k.a.j) \\
\equiv & \quad \{ \text{ De Morgan } \} \\
 & (\exists j : j \neq a : \neg k.j.a \vee k.a.j) \\
\Leftarrow & \quad \{ \ b \neq a \ \} \\
 & \neg k.b.a \vee k.a.b
\end{array}$$

We thus obtain the following program:

$$
\begin{array}{l}
\{(\exists i : 0 \leq i \leq N : S.i)\} \\
a, b := 0, N; \\
\textbf{do } a \neq b \rightarrow \\
\qquad \textbf{if } \neg k.b.a \vee k.a.b \rightarrow a := a + 1 \\
\qquad \;[]\; \neg K.a.b \vee K.b.a \rightarrow b := b - 1 \\
\qquad \textbf{fi} \\
\textbf{od}; \\
x := a;
\end{array}
$$

Exercises

Problem 7: The Starting Pit Location Problem

Given are $N + 1$ pits located along a circular racetrack. The pits are numbered clockwise from 0 up to and including N. At pit i, there are $p.i$ gallons of petrol available. To race from pit i to its clockwise neighbor one needs $q.i$ gallons of petrol. One is asked to design a linear algorithm to determine a pit from which it is possible to race a complete lap, starting with an empty fuel tank. To guarantee the existence of such a starting pit, we assume that

$$(\Sigma i : 0 \leq i \leq N : p.i) = (\Sigma i : 0 \leq i \leq N : q.i).$$