Chapter 6: Searching (Cont.)

Searching by Elimination
– Program Refinement –

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:

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

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:

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

Refinement

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

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

Application 1:

Derive a program that satisfies

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

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

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

we have

$$f.a \le f.b \ \Rightarrow \ \neg S.a \vee S.b$$

Similarly, we can derive

$$f.b \le f.a \ \Rightarrow \ \neg S.b \vee S.a$$

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:

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

Chapter 7: Segment Problems

This chapter is to show

- how problems may be solved;
- what decisions are made in the derivations;
- which properties play a specific role.

Longest Segment Problems

Let $N \geq 0$ and let $X[0..N)$ be an integer array. Find the longest subsegment $[p..q)$ of $[0..N)$ that satisfies a certain predicate like

- all elements are zero: $(\forall i : p \leq i < q : X.i = 0)$
- the segment is left-minimal: $(\forall i : p \leq i < q : X.p \leq X.i)$
- the segment contains at most 10 zeros: $(\#i : p \leq i < q : X.i = 0) \leq 10$
- all values are different: $(\forall i, j : p \leq i < j < q : X.i \neq X.j)$

All Zeros

Determine the legngth of a longest segment of $X[0..N)$ that contains zeros only.

$|[$

$\textbf{con}\ N : int\ \{N \geq 0\};\ X : \textbf{array}\ [0..N)\ \textbf{of}\ int;$

$\textbf{var}\ r : int;$

$all\ zeros$

$\{r = (\textbf{max}\ p, q : 0 \leq p \leq q \leq N \wedge (\forall i : p \leq i < q : X.i = 0) :\ q - p)\}$

$]|$

The post-condition is:

$$R: \ r = (\mathbf{max}\ p, q : 0 \leq p \leq q \leq N \wedge A.p.q : \ q - p)$$

where

$$A.p.q = (\forall i : p \leq i < q : X.i = 0)$$

What properties does A have?

For

$$A.p.q = (\forall i : p \leq i < q : X.i = 0)$$

we have

- A holds for empty segments: $A.n.n$
- A is prefix-closed: $A.p.q \Rightarrow (\forall i : p \leq i \leq q : A.p.i)$
- A is postfix-closed: $A.p.q \Rightarrow (\forall i : p \leq i \leq q : A.i.q)$

Our invariants are from R by replacing constant N by variable n:

$$P_0: \quad r = (\mathbf{max}\ p, q : 0 \le p \le q \le n \wedge A.p.q :\ q - p)$$
$$P_1: \quad 0 \le n < N$$

which is established by $n, r := 0, 0$.

What if $n := n + 1$?

$$
\begin{array}{ll}
 & (\mathbf{max}\ p, q : 0 \leq p \leq q \leq n+1 \wedge A.p.q :\ q-p) \\
= & \quad \{\ \text{split off } q = n+1\ \} \\
 & (\mathbf{max}\ p, q : 0 \leq p \leq q \leq n \wedge A.p.q :\ q-p)\ \mathbf{max} \\
 & \qquad (\mathbf{max}\ p, q : 0 \leq p \leq n+1 \wedge A.p.(n+1) :\ n+1-p) \\
= & \quad \{\ P_0\ \} \\
 & r\ \mathbf{max}\ (\mathbf{max}\ p : 0 \leq p \leq n+1 \wedge A.p.(n+1) :\ n+1-p) \\
= & \quad \{\ + \text{ distributes over } \mathbf{max}\ \} \\
 & r\ \mathbf{max}\ (n+1+(\mathbf{max}\ p : 0 \leq p \leq n+1 \wedge A.p.(n+1) :\ -p) \\
= & \quad \{\ \text{property of } \mathbf{max} \text{ and } \mathbf{min}\ \} \\
 & r\ \mathbf{max}\ (n+1-(\mathbf{min}\ p : 0 \leq p \leq n+1 \wedge A.p.(n+1) :\ p)) \\
= & \quad \{\ \underline{\text{invariant strengthening: } Q :\ s = (\mathbf{min}\ p : 0 \leq p \leq n \wedge A.p.n :\ p)}\ \} \\
 & r\ \mathbf{max}\ (n+1-s)
\end{array}
$$

We thus obtain a program of the following form.

$$
\begin{array}{l}
n, r, s := 0, 0, 0; \{\text{invariant: } P_0 \wedge P_1 \wedge Q, \text{ bound: } N - n\} \\
\mathbf{do}\ n \neq N \rightarrow \\
\qquad \text{establish } Q(n := n + 1) \\
\qquad r := r\ \mathbf{max}\ (n + 1 - s); \\
\qquad n := n + 1 \\
\mathbf{od}
\end{array}
$$

How to solve the subproblem establishing $Q(n := n+1)$:

$$Q: \ \ s = (\mathbf{min}\ p : 0 \le p \le n \land A.p.n : \ p)$$

We may remove **min** to the conjunction of the following predicates:

$$\begin{aligned} Q_0 : &\quad 0 \le s \le n \\ Q_1 : &\quad A.s.n \\ Q_2 : &\quad (\forall p : 0 \le p < s : \neg A.p.n) \end{aligned}$$

Lemma. If A is prefix-closed, then

$$Q_0 \land Q_2 \land A.s.(n+1) \Rightarrow Q(n := n+1)$$

From

$$Q_0 \wedge Q_2 \wedge A.s.(n+1) \Rightarrow Q(n := n+1)$$

we can establish $Q(n := n+1)$ by considering

- $Q_0 \wedge Q_2$ as invariant,
- $\neg A.s.(n+1)$ as guard, and
- $n+1-s$ as bound function.

Theorem

$$
\begin{array}{l}
|[\\
\{A \text{ holds for empty segment and prefix-closed}\} \\
\textbf{var } n, s : \mathit{int}; \\
n, r, s := 0, 0, 0; \\
\textbf{do } n \neq N \rightarrow \\
\qquad \textbf{do } \neg A.s.(n+1) \rightarrow s := s+1 \textbf{ od}; \\
\qquad r := r \textbf{ max } (n+1-s); \\
\qquad n := n+1 \\
\textbf{od} \\
\{r = (\textbf{max } p, q : 0 \leq p \leq q \leq N \wedge A.p.q : q-p)\} \\
]|
\end{array}
$$

What is its time complexity?

Add the variable t for counting the steps.

$$
\begin{array}{l}
|[\\
\textbf{var } n, s, t : int; \\
n, r, s, t := 0, 0, 0, 0; \\
\{\text{invariant} : Q : \ s = (\textbf{min } p : 0 \le p \le n \wedge A.p.n : \ p)\} \\
\textbf{do } n \ne N \rightarrow \\
\qquad \textbf{do } \neg A.s.(n+1) \rightarrow s := s+1; t := t+1 \textbf{ od}; \\
\qquad r := r \textbf{ max } (n+1-s); \\
\qquad n := n+1; t := t+1 \\
\textbf{od} \\
]|
\end{array}
$$

It is not difficult to see that $t = s + n \le 2N$. So if $A.s.(n+1)$ can be computed in constant time, then the above program is linear.

For the all-zeros problem, how $A.s.(n+1)$ can be computed in constant time?

from

$$A.p.q = (\forall i : p \le i < q : X.i = 0)?$$

Point: try to make use the the invariant:

$$Q : \;\; s = (\mathbf{min}\, p : 0 \le p \le n \wedge A.p.n : \;\; p)$$

Let us investigate the effect of $A.s.(n+1)$:

$$
\begin{array}{ll}
 & A.s.(n+1) \\
\equiv & \quad \{ \text{ definition of } A \ \} \\
 & (\forall i : s \leq i < n+1 : X.i = 0) \\
\equiv & \quad \{ \text{ split off } i = n, \ \underline{s \leq n} \ \} \\
 & (\forall i : s \leq i < n : X.i = 0) \wedge X.n = 0 \\
\equiv & \quad \{ \text{ definition of } A \ \} \\
 & A.s.n \wedge X.n = 0 \\
\equiv & \quad \{ \text{ by the invirant } Q \ \} \\
 & X.n = 0
\end{array}
$$

The final program:

$$
\begin{array}{l}
|[\\
\quad \mathbf{var}\ n, s : \mathit{int}; \\
\quad n, r, s := 0, 0, 0; \\
\quad \mathbf{do}\ n \neq N \rightarrow \\
\qquad \mathbf{do}\ X.n \neq 0 \wedge s \leq n \rightarrow s := s + 1\ \mathbf{od}; \\
\qquad r := r\ \mathbf{max}\ (n + 1 - s); \\
\qquad n := n + 1 \\
\quad \mathbf{od} \\
]|
\end{array}
$$

Exercises

[Problem 8-1] 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

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

[Problem 8-2]

Derive an $O(N)$ solution to the following problem.

$|[$

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

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

$all\ equal$

$\{r = (\mathbf{max}\ p, q : 0 \leq p \leq q \leq N \wedge (\forall i, j : p \leq i \leq j < q : X.i = X.j) :\ q - p)\}$

$]|$

About the Final Report

- Solve 8 problems freely selected from the exercises in the lecture notes.
- Submit your report to my post-box in the first floor of Engineering Building 6 no later than 17pm, June 19 (Thursday), 2008. Never forget writing your name, student identification number, and your department.
- Note you will be asked to present one or two of your solutions in class.