Chapter 4: General Programming Techniques

Strengthening Invariants (Cont.)

Maximum Segment Sum Problem

The problem of computing the maximal sum of the elements of segments $A[p..q)$ of a given integer array A.

- Specification

$$
\begin{array}{l}
|[\\
\textbf{con } N : int\ \{N \geq 0\};\ A :\ \textbf{array } [0..N)\ \textbf{of } int; \\
\textbf{var } r : int; \\
maxsegsum \\
\{r = (\textbf{max } p, q\ :\ 0 \leq p \leq q \leq N\ :\ (\Sigma i : p \leq i < q : A.i))\} \\
]|
\end{array}
$$

- Derivation

  From the post-condition:

$$\begin{array}{ll} R: & r = (\mathbf{max}\ p, q\ :\ 0 \le p \le q \le N\ :\ S.p.q) \\ S: & S.p.q = (\Sigma i : p \le i < q : A.i) \end{array}$$

  we replace the constnat N by variable n, obtaining the invariants:

$$\begin{array}{ll} P_0 & r = (\mathbf{max}\ p, q\ :\ 0 \le p \le q \le n\ :\ S.p.q) \\ P_1 & 0 \le n \le N \end{array}$$

  which are initialized by $n, r := 0, 0$.

We investigate the effect of incrementing n by 1. Assuming $P_0 \wedge P_1 \wedge n \neq N$, we have

$$
\begin{array}{ll}
 & (\mathbf{max}\ p,q\ :\ 0 \leq p \leq q \leq n+1\ :\ S.p.q) \\
= & \quad \{\ \text{split off } q = n+1\ \} \\
 & (\mathbf{max}\ p,q\ :\ 0 \leq p \leq q \leq n\ :\ S.p.q)\ \mathbf{max} \\
 & \qquad (\mathbf{max}\ p\ :\ 0 \leq p \leq n+1\ :\ S.p.(n+1)) \\
= & \quad \{\ P_0\ \} \\
 & r\ \mathbf{max}\ \underline{(\mathbf{max}\ p\ :\ 0 \leq p \leq n+1\ :\ S.p.(n+1))}
\end{array}
$$

We introduce additional invariant Q:

$$Q:\ s = (\mathbf{max}\ p\ :\ 0 \le p \le n\ :\ S.p.n)$$

Then $Q(n := n+1)$ equals the relation that is needed, i.e.,

$$\begin{array}{ll}
 & (\mathbf{max}\ p, q\ :\ 0 \le p \le q \le n+1\ :\ S.p.q) \\
= & \quad \{\ \text{previous derivation}\ \} \\
 & r\ \mathbf{max}\ \underline{(\mathbf{max}\ p\ :\ 0 \le p \le n+1\ :\ S.p.(n+1))} \\
= & \quad \{\ \text{assume } Q(n := n+1)\ \} \\
 & r\ \mathbf{max}\ s
\end{array}$$

We thus obtain a solution of the following form.

$|[$
$\textbf{var}\ n, s : \mathit{int}$
$n, r, s := 0, 0, 0;$
invariant: $P_0 \wedge P_1 \wedge Q$, bound: $N - n$;
$\textbf{do}\ n \neq N \rightarrow$
    establish $Q(n := n + 1)$;
    $r := r\ \textbf{max}\ s;$
    $n := n + 1$
$\textbf{od}$
$]|$

For $Q(n := n+1)$, we derive, assuming $P_0 \wedge P_1 \wedge Q \wedge n \neq N$,

$$\begin{array}{ll}
 & (\mathbf{max}\ p : 0 \leq p \leq n+1 : S.p.(n+1)) \\
= & \quad \{\ \text{split off } p = n+1\ \} \\
 & (\mathbf{max}\ p : 0 \leq p \leq n : \underline{S.p.(n+1)})\ \mathbf{max}\ \underline{S.(n+1).(n+1)} \\
= & \quad \{\ \text{definition of } S\ \} \\
 & (\mathbf{max}\ p : 0 \leq p \leq n : S.p.n + A.n))\ \mathbf{max}\ 0 \\
= & \quad \{\ + \text{ distributes over } \mathbf{max}\text{, when the range is non-empty } (0 \leq n)\ \} \\
 & ((\mathbf{max}\ p : 0 \leq p \leq n : S.p.n) + A.n)\ \mathbf{max}\ 0 \\
= & \quad \{\ Q\ \} \\
 & (s + A.n)\ \mathbf{max}\ 0
\end{array}$$

It follows that $Q(n := n+1)$ is established by $s := (s + A.n)\ \mathbf{max}\ 0$.

We therefore obtain the following $\mathcal{O}(N)$ program:

$|[$
$\textbf{var}\ n, s : int$
$n, r, s := 0, 0, 0;$
invariant: $P_0 \wedge P_1 \wedge Q$, bound: $N - n$;
$\textbf{do}\ n \neq N \rightarrow$
$\quad s := (s + A.n)\ \textbf{max}\ 0$
$\quad r := r\ \textbf{max}\ s;$
$\quad n := n + 1$
$\textbf{od}$
$]|$

A nice solution to a not so simple problem!

Tail Invariants

Design a program whose post-condition is

$$R: \; r = F.N$$

where F is defined in the following tail-recursive way:

$$\begin{array}{lcll} F.x & = & h.x & \text{if } b.x \\ F.x & = & F.(g.x) & \text{if } \neg b.x \end{array}$$

What is a suitable invariant? $\Rightarrow$ tail invariants!

A Direct Solution

$|[$

var x;

$x := X$;

$\{$<u>invariant: $F.x = F.X$, bound: assume that F terminates</u>$\}$

do $\neg b.x \rightarrow x := g.x$ **od**;

$r := h.x$

$]|$

$\{r := F.X\}$

Solving Problems by Tail Invariants

- *Example 1*

  Derive a program satisfying the following specification:

$$
\begin{array}{l}
|[\\
\mathbf{con}\ N : \mathit{int}\ \{N \geq 0\};\ A : \mathbf{array}\ [0..N]\ \mathbf{of}\ \mathit{int}; \\
\mathbf{var}\ r : \mathit{int}; \\
S \\
\{r = (\mathbf{max}\ i : 0 \leq i \leq N : A.i)\} \\
]|
\end{array}
$$

Define the function F by

$$F.x.y = (\mathbf{max}\ i : x \leq i \leq y : A.i)$$

which can be defined by the following tail recursion:

$$\begin{array}{lcll} F.x.y & = & A.x & \text{if } x = y \\ F.x.y & = & F.(x+1).y & \text{if } A.x \leq A.y \\ & = & F.x.(y-1) & \text{if } A.y \leq A.x \end{array}$$

$\textbf{var}\ x, y : \mathit{int};$

$x, y := 0, N;$

$\{\text{invariant } P\text{: } F.x.y = F.0.N \wedge 0 \leq x \leq y \leq N, \text{ bound: } y - x\}$

$\textbf{do}\ x \neq y \rightarrow$

$\quad \textbf{if}\ A.x \leq A.y \rightarrow x := x + 1$

$\quad [\,]\ A.y \leq A.x \rightarrow y := y - 1$

$\quad \textbf{fi}$

$\textbf{od};$

$r := A.x;$

$\{r = (\textbf{max}\ i : 0 \leq i \leq N : A.i)\}]|$

- *Example 2*:

  Design a program with post-condition

  $$r = G.N$$

  where N is a natural number, and $G.x$ is defined by

  $$\begin{aligned} G.0 &= 0 \\ G.x &= x \,\mathbf{mod}\, 10 + G.(x \,\mathbf{div}\, 10) \end{aligned}$$

  Is G a tail recursion?

From

$$\begin{aligned} G.0 &= 0 \\ G.x &= x \ \mathbf{mod}\ 10 + G.(x \ \mathbf{div}\ 10) \end{aligned}$$

we can define a new function G' for accumulating the result with another argument r:

$$G.x = G'.x.0$$

where

$$\begin{aligned} G'.0.r &= r \\ G'.x.r &= G'.(x \ \mathbf{div}\ 10).(r + x \ \mathbf{mod}\ 10) \end{aligned}$$

... applying the standard method ...

What kind of G can be transformed into tail recursion?

Let $\oplus$ is associative and has identity e. Then the function G defined by

$$\begin{array}{lcll} G.x & = & a & \text{if } b.x \\ G.x & = & h.x \oplus G.(g.x) & \text{if } \neg b.x \end{array}$$

can be transformed into

$$G.x = G'.x.e$$

where G' is a tail recursion defined by

$$\begin{array}{lcll} G'.x.r & = & r \oplus a & \text{if } b.x \\ G'.x.r & = & G'.(g.x).(r \oplus h.x) & \text{if } \neg b.x \end{array}$$

- *Example 3*

  Reconsider the problem of computation of A to the power B for given naturals A and B:

  $$
  \begin{array}{l}
  |[\\
  \textbf{con}\ A, B : int; \\
  \textbf{var}\ r : int; \\
  exponentiation \\
  \{r = A^B\} \\
  ]|
  \end{array}
  $$

The post-condition can be described by

$$r = G.A.B$$

where

$$\begin{array}{rcll} G.x.0 & = & 1 & \\ G.x.y & = & 1 * G.(x * x).(y \textbf{ div } 2) & \text{if } y \textbf{ mod } 2 = 0 \\ & = & x * G.x.(y - 1) & \text{if } y \textbf{ mod } 2 = 1 \end{array}$$

What are h and g?

From

$$\begin{array}{lcll} G.x.0 & = & 1 & \\ G.x.y & = & 1 * G.(x*x).(y \ \mathbf{div}\ 2) & \text{if } y \ \mathbf{mod}\ 2 = 0 \\ & = & x * G.x.(y-1) & \text{if } y \ \mathbf{mod}\ 2 = 1 \end{array}$$

we get the definition for h and g as follows.

$$\begin{array}{lcll} h.x.y & = & 1 & \text{if } y \ \mathbf{mod}\ 2 = 0 \\ & = & x & \text{if } y \ \mathbf{mod}\ 2 = 1 \\ g_1.x.y & = & x * x & \text{if } y \ \mathbf{mod}\ 2 = 0 \\ & = & x & \text{if } y \ \mathbf{mod}\ 2 = 1 \\ g_2.x.y & = & y \ \mathbf{div}\ 2 & \text{if } y \ \mathbf{mod}\ 2 = 0 \\ & = & y - 1 & \text{if } y \ \mathbf{mod}\ 2 = 1 \end{array}$$

Therefore, we obtain the following program:

$$
\begin{array}{l}
\textbf{var } x, y : \mathit{int}; \{A \geq 0, B \geq 0\} \\
r, x, y = 1, A, B; \\
\{\text{invariant: } r * G.x.y = G.A.B \wedge 0 \leq 0, \text{ bound: } y\} \\
\textbf{do } y \neq 0 \rightarrow \\
\quad \text{if } y \textbf{ mod } = 0 \rightarrow x, y, r = x * x, y \textbf{ div } 2, r * 1 \\
\quad [] \; y \textbf{ mod } 2 = 1 \rightarrow x, y, r = x, y - 1, r * x \\
\quad \textbf{fi} \\
\textbf{od} \\
]|
\end{array}
$$

An $\mathcal{O}(\log B)$ program!

Summary of Chapter 4

We discussed four general techniques that show how a suitable invariant may be derived from a given pre and post-condition.

- Taking conjuncts
- Replacing constants by variables
- Strengthening invariants
- Tail invariants

Exercises

Problem 5

Solve

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

by defining for $0 \leq n \leq N$

$$G.n \equiv (\exists i : n \leq i < N : f.i = 0)$$

and deriving a suitable recurrence relation for G.