We are asked to calculate the maximum segment sum and its start and end indices. The program for calculating the sum is

$|[$ **con** N : int $\{0 \le N\}$; f : **array**$[0..N)$ **of** int;

var r, n, s : int;

$n, r, s \ := \ 0, 0, 0 \ \{ \ inv : \ (1) \wedge (2) \wedge (3) \}$

; **do** $n \neq N \rightarrow$

if $0 \le s + f.n \rightarrow s := s + f.n$

$[]\ s + f.n \le 0 \rightarrow s := 0$

fi

; **if** $s \le r \rightarrow skip$

$[]\ r \le s \rightarrow r := s$

fi

; $n := n + 1$

od

$]|$.

where

(1) $r = \langle \uparrow i, j : \ 0 \le i \le j \le n : \ S.i.j \rangle$

(2) $0 \le n \le N$

(3) $s = \langle \uparrow i : \ 0 \le i \le n : \ S.i.n \rangle$

To calculate the indices we strengthen the invariant with the conjunction of

(4) $r = S.p.q$

(5) $s = S.k.n$

(6) $0 \le p \le q \le N$

(7) $0 \le k \le N$

whch we establish with $n, r, s, k, p, q \ := \ 0, 0, 0, 0, 0, 0$.

* * *

Our first task is to maintain the invariant $s = S.k.n$. There are three assignment statements which could conceivably violate this invariant. We investigate each in turn, working backwards. First we observe

|[

▷

$(n \ := \ n+1).(s = S.k.n)$

$=$ { substitution }

$s = S.k.(n+1)$

]|

This gives us the postcondition of the other two commands. As they are at the beginning of the loop body, we must ensure that their precondition is $s = S.k.n$.

|[*Cxt:* $s \ = \ S.k.n$

▷

$(s \ := \ s + f.n).(s = S.k.(n+1))$

$=$ { substitution }

$s + f.n \ = \ S.k.(n+1)$

$=$ { arithmetic }

$s \ = \ S.k.(n+1) - f.n$

$=$ { definition of S }

$s \ = \ S.k.n$

$=$ { context }

true

]|

and

|[*Cxt:* $s = S.k.n$

▷

$(s := 0).(s = S.k.(n+1))$

$=$ { substitution }

$0 = S.k.(n+1)$

$=$ { assuming $k = n+1$ }

true

]|

The above calculation tells us that if we wish $s := 0$ to establish $s = S.k.(n+1)$ we had better assign $n+1$ to k. We now have a program of the form:

$n, r, s, k, p, q \ := \ 0, 0, 0, 0, 0, 0$

; **do** $n \neq N \quad \rightarrow$

  if $0 \ \leq \ s + f.n \quad \rightarrow \quad s \ := \ s + f.n$

$[]\ s + f.n \ \le\ 0 \quad \to \quad s, k \ :=\ 0, n+1$

fi

; **if** $s \le r \quad \to \quad \mathit{skip}$

$[]\ r \le s \quad \to \quad r := s$

fi

; $n \ :=\ n+1$

od

Now we may deal with the invariant $r = S.p.q$.

$|[$ *Cxt:* $\ s = S.k.(n+1)$

$\triangleright$

$(r := s).(r = S.p.q)$

$=$ { substitution }

$s = S.p.q$

$=$ { assuming $p = k \quad \wedge \quad q \ =\ n+1$ }

$s = S.k.(n+1)$

$=$ { context }

true

$]|$

Thus to establish $r = S.p.q$ we need to assign k to p and $n+1$ to q resulting in the final program

$n, r, s, k, p, q \ :=\ 0, 0, 0, 0, 0, 0$

; **do** $\quad n \ne N \quad \to$

if $0 \ \le\ s + f.n \quad \to \quad s \ :=\ s + f.n$

$[]\ s + f.n \ \le\ 0 \quad \to \quad s, k \ :=\ 0, n+1$

fi

; **if** $s \le r \quad \to \quad \mathit{skip}$

$[]\ r \le s \quad \to \quad r, p, q \ :=\ s, k, n+1$

fi

; $n \ :=\ n+1$

od

31 December 2006