The maximal AB-segment

(a very simple programming problem, just for our records)

Given two integers A and B, and integer array $f[0..N)$, $1 \le N$, we wish to develop a program establishing postcondition

$R: \quad r = \langle \uparrow i,j : 0 \le i < j < N \wedge f.i = A \wedge f.j = B : S.i.j \rangle$,

where $\quad S.i.j = \langle \Sigma k : i \le k \le j : f.k \rangle$.

* * *

We follow a standard procedure by introducing

$P0: \quad 1 \le n \le N$

$P1: \quad r = \langle \uparrow i,j : 0 \le i < j < n \wedge f.i = A \wedge f.j = B : S.i.j \rangle$,

heading for a program of the form

```
   n := 1 ; r := -∞
 ; {inv. P0 ∧ P1}{vf N-n}
   do n ≠ N →
        r : S
      ; { ? P1(n:=n+1) }
        n := n+1
   od
   {R} .
```

* * *

<u>Re $P1(n := n+1)$</u>

$\|[\ 1 \le n,\ P1,\ 0 \le n < N,\ P2(n := n+1)$

$\triangleright\quad \langle \uparrow i,j : 0 \le i < j < n+1 \wedge f.i = A \wedge f.j = B : S.i.j \rangle$

$=\qquad$ { split off $j = n$, using $1 \le n$; use $P1$ }

$\qquad r \uparrow \langle \uparrow i : 0 \le i < n \wedge f.i = A \wedge f.n = B : S.i.n \rangle$

$=\qquad$ { case distinction on $f.n$, using $0 \le n < N$ }

$\qquad$ <u>if</u> $f.n \ne B \rightarrow r$

$\qquad$ [] $f.n = B \rightarrow$

$\qquad\qquad r \uparrow \langle \uparrow i : 0 \le i < n \wedge f.i = A \wedge f.n = B : S.i.n \rangle$

$\qquad$ <u>fi</u>

$=\qquad$ { intro $P2$, defined below; use $P2(n := n+1)$ }

$\qquad$ <u>if</u> $f.n \ne B \rightarrow r$

$\qquad$ [] $f.n = B \rightarrow r \uparrow s$

$\qquad$ <u>fi</u>

$]|$.

$P2: \qquad s = \langle \uparrow i : 0 \le i < n-1 \wedge f.i = A : S.i.(n-1) \rangle$.

By strengthening the invariant with $P2$, we thus find for $r: S$

$$
\begin{array}{l}
s:\ T \\
;\ \{?\ P2(n:=n+1)\}\ \{1\le n<N\}\ \{P1\} \\
\quad \underline{\text{if}}\ f.n \ne B \ \to\ \text{skip} \\
\quad [\!]\ \ f.n = B \ \to\ r := r \uparrow s \\
\quad \underline{\text{fi}} \\
\quad \{P1\,(n:=n+1)\}\ .
\end{array}
$$

* * *

Re P2 (n := n+1)

$$
\begin{array}{l}
|[\ 1\le n,\ P2, \\
\quad \triangleright\ \langle \uparrow i : 0\le i<n\ \wedge\ f.i = A : S.i.n\rangle \\
\quad =\qquad \{\ S.i.n\ =\ f.n\ +\ S.i.(n-1)\ ; \\
\qquad\qquad +\ \text{over}\ \uparrow\ ,\ \text{see footnote}\} \\
\qquad f.n\ +\ \langle \uparrow i : 0\le i<n\ \wedge\ f.i = A : S.i.(n-1)\rangle \\
\quad =\qquad \{\ \text{split off } i=n-1,\ \text{using } 1\le n\ ; \\
\qquad\qquad \text{use } P2 \text{ for } i<n-1\ ; \\
\qquad\qquad \text{case distinction on } f.(n-1) \text{ for } i=n-1, \\
\qquad\qquad \text{using } 0\le n-1<N\ ;\quad S.(n-1).(n-1) = f.(n-1)\} \\
\qquad \underline{\text{if}}\ f.(n-1)\ne A \ \to\ f.n + (s \uparrow -\infty) \\
\qquad [\!]\ \ f.(n-1) = A \ \to\ f.n + (s\uparrow f.(n-1)) \\
\qquad \underline{\text{fi}} \\
]|,
\end{array}
$$

which solves $s:T$.

* * *

We adopt the rule $x + (-\text{inf}) = -\text{inf}$.

We thus found (calculated) as a solution to our programming problem

```
n := 1 ; r := −∞ ; s := −∞
; do n ≠ N →
      if f.(n−1) ≠ A  →  s := f.n + s
      [] f.(n−1) = A  →  s := f.n + (s ↑ f.(n−1))
      fi
    ; if f.n ≠ B  →  skip
      [] f.n = B  →  r := r ↑ s
      fi
    ; n := n+1
  od .
```

* * *

- I am rather sure that the above solution is not within easy reach of the operationally inclined.
- The above problem and its solution can act as a typical example in our first-year course on program design.
- Some faculty within our department does not "believe" [sic] —without motivation— in calculational program design, an attitude which the other day has been rated as very unprofessional.

WHJ Feijen
9 May 2003