Chapter 19 : Generic longest segments.

In which we abstract from the solution to the longest zero segment..

Given f[0..N) of int, $\{0 \le N\}$, we are asked to construct a program to determine the length of the longest segment in f where each value in the segment satisfies property Q.

$\{$f[0..N) of int contains values$\}$

S

$\{ r = \langle \uparrow i,j : 0 \le i \le j \le N \wedge AQ.i.j : j\text{-}i \rangle\}$

Domain model.

As usual we begin by building a model of the domain.

$*$ (0) $AQ.i.j \equiv \langle \forall k : i \le k < j : Q.(f.k)\rangle$ $\quad ,0 \le i \le j \le N$

Exploiting the empty range and associativity gives us

\- (1) $AQ.i.i \equiv true$ $\quad ,0 \le i \le N$

\- (2) $AQ.i.(j+1) \equiv AQ.i.1 \wedge Q.(f.j)$ $\quad ,0 \le i \le j < N$

We name the quantified expresion in our postcondition.

$*$ (3) $C.n = \langle \uparrow i,j : 0 \le i \le j \le n \wedge AQ.i.j : j\text{-}i \rangle$ $\quad ,0 \le n \le N$

Appealing to the “1 point” rule and (1) gives us

\- (4) $C.0 = 0$

In an effort to exploit associativity we calculate as follows

$C.(n+1)$

$=$ $\{(3)\}$

$\langle \uparrow i,j : 0 \le i \le j \le n+1 \wedge AQ.i.j : j\text{-}i \rangle$

$=$ $\{$split off $j = n+1$ term$\}$

$\langle \uparrow i,j : 0 \le i \le j \le n \wedge AQ.i.j : j\text{-}i \rangle \uparrow \langle \uparrow i : 0 \le i \le n+1 \wedge AQ.i.(n+1) : (n+1)\text{-}i \rangle$

$=$ $\{(3)\}$

$C.n \uparrow \langle \uparrow i : 0 \le i \le n+1 \wedge AQ.i.(n+1) : (n+1)\text{-}i \rangle$

$=$ $\{$name the new expression (6)$\}$

$C.n \uparrow D.(n+1)$

Which gives us

- (5) $C.(n+1) \; = \; C.n \uparrow D.(n+1) \qquad ,0 \le n < N$

* (6) $D.n \; = \; \langle \uparrow i : 0 \le i \le n \wedge AQ.i.n : n-i \rangle \qquad ,0 \le n \le N$

An appeal to the "1 point rule" and (1) gives us

- (7) $D.0 \; = \; 0$

Seeking to exploit associativity, we calculate as follows

$D.(n+1)$

$=$ $\{(6)\}$

$\langle \uparrow i : 0 \le i \le n+1 \wedge AQ.i.(n+1) : (n+1)-i \rangle$

$=$ {split off $i = n+1$ term}

$\langle \uparrow i : 0 \le i \le n \wedge AQ.i.(n+1) : (n+1)-i \rangle \uparrow (n+1)-(n+1)$

$=$ {arithmetic}

$\langle \uparrow i : 0 \le i \le n \wedge AQ.i.(n+1) : (n+1)-i \rangle \uparrow 0$

$=$ $\{(2)\}$

$\langle \uparrow i : 0 \le i \le n \wedge AQ.i.n \wedge Q.(f.n) : (n+1)-i \rangle \uparrow 0$

{case analysis, $Q.(f.n)$, $true \equiv Id\wedge$}

$\langle \uparrow i : 0 \le i \le n \wedge AQ.i.n : (n+1)-i \rangle \uparrow 0$

$=$ {$+/\uparrow$ for non-empty ranges}

$(1 + \langle \uparrow i : 0 \le i \le n \wedge AQ.i.n : n-i \rangle) \uparrow 0$

$=$ $\{(6)\}$

$(1 + D.n) \uparrow 0$

Now let us calculate using the other case. We observe

$D.(n+1)$

$=$ $\{(6)\}$

$\langle \uparrow i : 0 \le i \le n+1 \wedge AQ.i.(n+1) : (n+1)-i \rangle$

$=$ {split off $i = n+1$ term}

$\langle \uparrow i : 0 \le i \le n \wedge AQ.i.(n+1) : (n+1)-i \rangle \uparrow (n+1)-(n+1)$

$=$ {arithmetic}

$\langle \uparrow i : 0 \le i \le n \wedge AQ.i.(n+1) : (n+1)-i \rangle \uparrow 0$

$=$ $\{(2)\}$

$\langle \uparrow i : 0 \le i \le n \wedge AQ.i.n \wedge Not.(Q.(f.n)) : (n+1)-i \rangle \uparrow 0$

$=$ {case analysis, $Not.(Q.(f.n))$, $P \wedge false \equiv false$}

$\langle \uparrow i : false : (n+1)-i \rangle \uparrow 0$

= {empty range}

$Id\uparrow \uparrow 0$

= {defn. of Id}

0

So we have

- (8) $D.(n+1) = (1+D.n)\uparrow 0 \quad \Leftarrow Q.(f.n) \quad ,0 \le n < N$

- (9) $D.(n+1) = 0 \quad \Leftarrow Not.(Q.(f.n)) \quad ,0 \le n < N$

Rewrite and strengthen the postcondition.

Post : $r = C.N$

Choose invariants.

We choose the following invariants

P0 : $r = C.n \wedge d = D.n$
P1 : $0 \le n \le N$

Establish invariants.

From our model, in particular laws (4) and (7), we can see that the following assignment establishes the invariants

$n, r, d := 0, 0, 0$

Guard.

$n \ne N$

Variant.

n

Calculate the loop body.

We achieve our standard decrease of vf by increasing n by 1. We calculate the loop body using this.

$(n, r, d := n+1, E, E').P0$

= {textual substitution}

$E = C.(n+1) \wedge E' = D.(n+1)$

= {(5)}

$E = C.n \uparrow D.(n+1) \wedge E' = D.(n+1)$

= {case analysis, f.n=0, (8) twice}

$E = C.n \uparrow (1+D.n)\uparrow 0 \wedge E' = (1+D.n)\uparrow 0$

= {P0}

$E = r \uparrow (1+d)\uparrow 0 \wedge E' = (1+d)\uparrow 0$

This gives us

If f.n=0 $\rightarrow$ n. r. d := n+1, $r \uparrow (1+d)\uparrow 0$, $(1+d)\uparrow 0$

We now consider the other case

(n, r, d := n+1, E, E').P0

= {textual substitution}

$E = C.(n+1) \wedge E' = D.(n+1)$

= {(5)}

$E = C.n \uparrow D.(n+1) \wedge E' = D.(n+1)$

= {case analysis, f.n≠0, () twice}

$E = C.n \uparrow 0 \uparrow 0 \wedge E' = 0\uparrow 0$

= {P0 and $\uparrow$ idempotent}

$E = r\uparrow 0 \wedge E' = 0$

This gives us

If f.n≠0 $\rightarrow$ n. r. d := n+1, $r\uparrow 0$, 0

Finished program.

```
n, r, d := 0, 0, 0 {P0 ∧ P1}
;do n≠N →      {P0 ∧ P1 ∧ n≠N}

               If Q.(f.n) → n. r. d := n+1, r ↑ (1+d)↑0, (1+d)↑0
               [] Not.(Q.(f.n)) → n. r. d := n+1, r↑0, 0
               fi

               {P0 ∧ P1}
od
{r = C.N}
```

Note.

Our final observation is that as both D.n and C.n are natural values they cannot be negative. Therefore we can simplify some of the expressions. We note that

$r \uparrow 0 = r$

$d \uparrow 0 = d$

Using this we can rewrite our finished program like this.

```
n, r, d := 0, 0, 0 {P0 ∧ P1}
;do n≠N →     {P0 ∧ P1 ∧ n≠N}

              If Q.(f.n) → n. r. d := n+1, r ↑ (1+d), (1+d)
              [] Not.(Q.(f.n)) → n. r. d := n+1, r, 0
              fi

              {P0 ∧ P1}
od
{r = C.N}
```

Note that this works as long as Q is defined for individual elements of f.