## Chapter 32 : Head/Tail sum.

*In which we introduce a new technique: strengthening an invariant.*

We are given f[0..N) of strictly positive values, where {0≤N}. We are asked to establish the following postcondition.

Post : $r = \langle +\, i,j : 0 \le i \le N \wedge 0 \le j \le N : g.(H.i).(T.j) \rangle$

Where H and T are defined as follows.

* (0) $H.i = \langle +\, j : 0 \le j < i : f.j \rangle$ ,$0 \le i \le N$

* (1) $T.i = \langle +\, j : i \le j < N : f.j \rangle$ ,$0 \le i \le N$

And g is defined as

* (2) $g.x.y = 1 \Leftarrow x = y$

* (3) $g.x.y = 0 \Leftarrow x \neq y$

Let us see what nice theorems we can get from these definitions

- (4) $H.0 = 0$

- (5) $H.(n+1) = H.n + f.n$ , $0 \le n < N$

- (6) $T.N = 0$

- (7) $T.(n-1) = T.n + f.(n-1)$ , $0 < n \le N$

We also note that

- (8) $H.N = T.0$

* (9) $C.m.n = \langle +\, i,j : m \le i \le N \wedge 0 \le j \le n : g.(H.i).(T.j) \rangle$

- (10) $C.N.n = 1$

- (11) $C.m.0 = 1$

We observe

```
        C.m.n
=               {(9)}
        ⟨+ i,j : m ≤ i ≤ N ∧ 0 ≤ j ≤ n : g.(H.i).(T.j)⟩
=               { m < N, split off i=m term }
        ⟨+ i,j : m+1 ≤ i ≤ N ∧ 0 ≤ j ≤ n : g.(H.i).(T.j)⟩ + ⟨+ j : 0 ≤ j ≤ n : g.(H.m).(T.j)⟩
=               { (9), (14) }
        C.(m+1).n + D.n
```

- (12) $C.m.n = C.(m+1).n + D.n \quad ,0 \le m < N$

We observe

```
        C.m.n
=               {(9)}
        ⟨+ i,j : m ≤ i ≤ N ∧ 0 ≤ j ≤ n : g.(H.i).(T.j)⟩
=               { 0 < n, split off j=n term }
        ⟨+ i,j : m ≤ i ≤ N ∧ 0 ≤ j ≤ n-1 : g.(H.i).(T.j)⟩ + ⟨+ i : m ≤ i ≤ N : g.(H.i).(T.n)⟩
=               { (9), (15) }
        C.m.(n-1) + E.m
```

- (13) $C.m.n = C.m.(n-1) + E.m \quad ,0 < n \le N$

* (14) $D.n = \langle + j : 0 \le j \le n : g.(H.m).(T.j) \rangle$

* (15) $E.m = \langle + i : m \le i \le N : g.(H.i).(T.n) \rangle$

Now lets examine D and E. The definitions of g above suggest we consider the cases H.m = T.n and H.m ≠ T.n. However, as both H and T have monotonic properties, these should be taken into account so we consider the 3 cases below.

- (16) $D.n = 0 \Leftarrow H.m < T.n$

- (17) $D.n = 1 \Leftarrow H.m = T.n$

- (18) $D.n = ? \Leftarrow H.m > T.n$

- (19) E.m = ? ⇐ H.m < T.n

- (20) E.m = 1 ⇐ H.m = T.n

- (21) E.m = 0 ⇐ H.m > T.n

*Rewrite Postcondition.*

Post : r = C.0.N

*Invariants.*
As our invariants we choose

P0 : r + C.m.n = C.0.N
P1 : $0 \le m \le N \wedge 0 \le n \le N$

*Establishing the invariants.*
We can establish the invariants as follows

r, m, n := 0, 0, N

*Termination.*

Our model (10) and (11) suggests that we consider a number of cases.

Case m = N

P0 ∧ m = N
⇒ {Leibniz}
r + C.N.n = C.0.N
= {(10)}
r + 1 = C.0.N

Case n = 0

P0 ∧ n = 0
⇒ {Leibniz}
r + C.m.0 = C.0.N
= {(11)}
r + 1 = C.0.N

The pair of these suggest that

P0 ∧ P1 ∧ (m=N ∨ n=0) ⇒ r+1 = C.0.N

*Guard.*
Based on this we choose as guard

$$m \neq N \wedge n \neq 0$$

*Loop body.*
Now let us calculate the loop body. First we utilise what we know about D

```
        P0
=               {definition of P0}
        r + C.m.n = C.0.N
=               {P1 ∧ m ≠ N, (12)}
        r + C.(m+1).n + D.n = C.0.N
=               {case analysis H.m < T.n (16)}
        r + C.(m+1).n + 0 = C.0.N
=               {WP}
        (r, m := r + 0, m+1).P0
```

We consider the case H.m = T.n.

```
        P0
=               {definition of P0}
        r + C.m.n = C.0.N
=               {P1 ∧ m ≠ N, (12)}
        r + C.(m+1).n + D.n = C.0.N
=               {case analysis H.m = T.n (17)}
        r + C.(m+1).n + 1 = C.0.N
=               {WP}
        (r, m := r + 1, m+1).P0
```

Now lets utilise what we know about E

We consider the case H.m = T.n.

```
        P0
=               {definition of P0}
        r + C.m.n = C.0.N
=               {P1 ∧ n ≠ 0, (17)}
        r + C.m.(n-1) + E.m = C.0.N
=               {case analysis H.m = T.n (20)}
        r + C.m.(n-1) + 1 = C.0.N
=               {WP}
        (r, n := r + 1, n-1).P0
```

We consider the case H.m > T.n

```
        P0
=               {definition of P0}
        r + C.m.n = C.0.N
=               {P1 ∧ n ≠ 0, (13)}
        r + C.m.(n-1) + E.m = C.0.N
=               {case analysis H.m > T.n (21)}
        r + C.m.(n-1) + 0 = C.0.N
=               {WP}
        (r, n := r + 0, n-1).P0
```

*Finished Algorithm.*

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

                     If H.m < T.n → r, m := r+0, m+1
                     [] H.m = T.n → r, m := r+1, m+1
                     [] H.m = T.n → r, n := r+1, n-1
                     [] H.m > T.n → r, n := r+0, n-1
                     Fi

                     {P0 ∧ P1}
od
{r+1 = C.0.N}
;r := r+1
{r = C.0.N}
```

*Sad realisation.*
Nice and all as this is, we are faced with a problem. Notice the way the guards contain H.m and T.n. Sadly, if we have to evaluate these each time we enter the loop then we are raising the complexity of the algorithm by an order. It would be nice to avoid this.
*End of sad realisation.*

We propose to bind H.m and T.n to variables. This involves strengthening our invariant as follows.

P2 : x = H.m ∧ y = T.n

We now need to ensure that this invariant is established or maintained by each of the program assignments. We will consider each in turn.

** *m, n, r := 0, N, 0*

(m, n, r, x, y := 0, N, 0, U, U').P2
= {textual substitution}
U = H.0 ∧ U' = T.N
= {(4) (6)}
U = 0 ∧ U' = 0

So the assignment becomes

m, n, r, x, y := 0, N, 0, 0, 0

** *r, m := r+0, m+1*

(r, m, x, y := r+0, m+1, U, U').P2
= {textual substitution}
U = H.(m+1) ∧ U' = T.n
= {(5)}
U = H.m + f.m ∧ U' = T.n
= {P2}
U = x + f.m ∧ U' = y

So the assignment becomes

r, m, x, y := r+0, m+1, x+f.m, y

** *r, m := r+1, m+1*

(r, m, x, y := r+1, m+1, U, U').P2
= {textual substitution}
U = H.(m+1) ∧ U' = T.n
= {(5)}
U = H.m + f.m ∧ U' = T.n
= {P2}
U = x + f.m ∧ U' = y

So the assignment becomes

r, m, n, x, y := r+1, m+1, x + f.m, y

** *r, m, n := r+1, n-1*

(r, m, n, x, y := r+1, n-1, U, U').P2
= {textual substitution}
U = H.m ∧ U' = T.(n-1)
= {(7)}
U = H.m ∧ U' = T.n + f.(n-1)
= {P2}
U = x ∧ U' = y + f.(n-1)

So the assignment becomes

r, n, x, y := r+1, n-1, x  y + f.(n-1)

** *r, n := r+0, n-1*

(r, n, x, y := r+0, n-1, U, U').P2
= {textual substitution}
U = H.m ∧ U' = T.(n-1)
= {(7)}
U = H.m ∧ U' = T.n + f.(n-1)
= {P2}
U = x ∧ U' = y + f.(n-1)

So the assignment becomes

r, n, x, y := r+0, n-1, x, y + f.(n-1)

As P2 is now an invariant of our program, we can replace the references to H.m and T.n with references to x and y. And so we get our final program

m, n, r, x, y := 0, N, 0, 0, 0 {P0 ∧ P1 ∧ P2}
;do m ≠ N ∧ n ≠ 0 → {P0 ∧ P1 ∧ P2 ∧ m ≠ N ∧ n ≠ 0}

If x < y → r, m, x, y := r+0, m+1, x + f.m, y
[] x = y → r, m, x, y := r+1, m+1, x + f.m, y
[] x = y → r, n, x, y := r+1,  n-1, x y + f.(n-1)
[] x > y → r, n, x, y := r+0, n-1, x, y + f.(n-1)
Fi
{P0 ∧ P1 ∧ P2}
od
{r+1 = C.0.N}
;r := r+1
{r = C.0.N}