Chapter 41 : Fibolucci

Given f : Nat --> Nat defined as

$*$ (0) $f.0 = 0$
$*$ (1) $f.1 = 1$
$*$ (2) $f.(n+2) = f.n + f.(n+1)$

Given N : Int $1 \le N$ establish

Post : $r = \langle + i : 0 \le i \le N : f.i * f.(N-i) \rangle$

We are asked to ensure that out program is efficient and doesn't contain any explicit references to f.

We strengthen to give

Post' : $r = \langle + i : 0 \le i \le n : f.i * f.(n-i) \rangle \wedge n = N$

Model

$*$ (3) $C.n = \langle + i : 0 \le i \le n : f.i * f.(n-i) \rangle$

We observe

C.0
= { (3) }
$\langle + i : 0 \le i \le 0 : f.i * f.(0-i) \rangle$
= { 1-point }
f.0 * f.0
= { (0) }
0

- (4) C.0 = 0

We observe

C.1
= { (3) }
$\langle + i : 0 \le i \le 1 : f.i * f.(1-i) \rangle$
= { 1-point twice }
f.0 * f.1 + f.1 * f.0
= { (0) (1) }
0

- (5) C.1 = 0

We observe

$$
\begin{array}{ll}
 & C.(n+1) \\
= & \qquad \{ (3) \} \\
 & \langle + i : 0 \leq i \leq n+1 : f.i * f.((n+1)-i) \rangle \\
= & \qquad \{ \text{Split off } i = n+1 \text{ term} \} \\
 & \langle + i : 0 \leq i \leq n : f.i * f.((n+1)-i) \rangle + f.(n+1) * f.((n+1) - (n+1)) \\
= & \qquad \{ \text{algebra} \} \\
 & \langle + i : 0 \leq i \leq n : f.i * f.((n+1)-i) \rangle + f.(n+1) * f.0 \\
= & \qquad \{ (0) \} \\
 & \langle + i : 0 \leq i \leq n : f.i * f.((n+1)-i) \rangle \\
= & \qquad \{ \text{Split off } i = n \text{ term [1]} \} \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * f.((n+1)-i) \rangle + f.n * f.((n+1) - n) \\
= & \qquad \{ \text{Algebra} \} \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * f.((n+1)-i) \rangle + f.n * f.1 \\
= & \qquad \{ (1) \} \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * f.((n+1)-i) \rangle + f.n \\
= & \qquad \{ (2) \} \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * (f.((n-1) - i) + f.(n -i)) \rangle + f.n \\
= & \qquad \{ */+ \} \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * f.((n-1)-i) + f.i * f.(n -i) \rangle + f.n \\
= & \qquad \{ + \text{ associative} \} \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * f.((n-1)-i) \rangle + \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * f.(n-i) \rangle + f.n \\
= & \qquad \{ f.n * f.(n-n) = 0, + \text{ associative} \} \\
 & \langle + i : 0 \leq i \leq n-1 : f.i * f.((n-1)-i) \rangle + \\
 & \langle + i : 0 \leq i \leq n : f.i * f.(n-i) \rangle + f.n \\
= & \qquad \{ (3) \} \\
 & C.(n-1) + C.n + f.n
\end{array}
$$

- (6) $C.(n+1) = C.(n-1) + C.n + f.n$

Rewrite Postcondition.

$\text{Post'} : r = C.n \wedge n = N$

Invariants.

$P0 : r = C.n \wedge s = C.(n-1)$
$P1 : 1 \leq n \leq N$

Guard.

$n \neq N$

---

[1] We must ensure that $1 \leq n$ to guarantee this is possible. We do this so as to ensure $2 \leq (n+1) - i$

Establish Invariants.

n, r, s := 1. 0, 0

Variant.

N-n

Loop body.

$$
\begin{array}{ll}
 & (n, r, s := n+1, E, E').P0 \\
= & \quad \{ \text{ text sub. } \} \\
 & E = C.(n+1) \wedge E' = C.n \\
= & \quad \{ (6) \} \\
 & E = C.(n-1) + C.n + f.n \wedge E' = C.n \\
= & \quad \{ P0 \} \\
 & E = s + r + f.n \wedge E' = r
\end{array}
$$

Finished Algorithm.

n, r, s := 1. 0, 0
; do n ≠ N →
  n, r, s := n+1, s + r + f.n, r
od

Now we want to eliminate any explicit references to f in the program. Towards doing this we strengthen our invariant.

P2: $a = f.n \wedge b = f.(n-1)$

These are established by n, a, b := 1, 1, 0

Within the loop body we seek to maintain P2 while decreasing vf

$$
\begin{array}{ll}
 & (n, a, b := n+1, E, E').P2 \\
= & \quad \{ \text{ text sub. } \} \\
 & E = f.(n+1) \wedge E' = f.n \\
= & \quad \{ (2) \} \\
 & E = f.(n-1) + f.n \wedge E' = f.n \\
= & \quad \{ P2 \} \\
 & E = b + a \wedge E' = a
\end{array}
$$

We can now eliminate references to f entirely

n, r, s, a, b := 1. 0, 0, 1, 0

```
; do n ≠ N →
    n, r, s, a, b := n+1, s + r + a, r, b+a, a
 od
```

Of course now we can make the observation that within the loop body the expressions on the right hand side of the assignment are linear combinations of the variables on the left hand side. So, a Log(N) solution is possible. We leave the development of that solution to the reader.