

Chapter 8: The Partitioned Reduction Theorem I.

In which we introduce another generic problem.

We are given an array $f[0..N)$ of int which contains values and we are asked to construct a program to count the number of even values in f . We begin by specifying the problem.

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

S

$\{r = \langle +j : 0 \leq j < N : g.(f.j) \rangle\}$

where

$$\begin{array}{llll} g.x & = & 1 & \Leftarrow \text{even}.x \\ g.x & = & 0 & \Leftarrow \text{odd}.x \end{array}$$

Postcondition.

$\text{Post} : r = \langle +j : 0 \leq j < N : g.(f.j) \rangle$

Strengthen postcondition.

$\text{Post}' : r = \langle +j : 0 \leq j < n : g.(f.j) \rangle \wedge n = N$

Domain modelling.

Inspired by the shape of our postcondition, we now proceed to develop a little mathematical model of our domain. We begin with a single postulate.

$* (0) C.n = \langle +j : 0 \leq j < n : g.(f.j) \rangle, 0 \leq n \leq N$

The function g is defined as follows:

$* (1) g.x = 1 \Leftarrow \text{even}.x$

$* (2) g.x = \text{Id}+ \Leftarrow \text{odd}.x$

We now explore some theorems.

Consider

$$\begin{aligned}
 & C.0 \\
 = & \quad \{(0) \text{ in model } \} \\
 & \langle +j : 0 \leq j < 0 : g.(f.j) \rangle \\
 = & \quad \{ \text{empty range} \} \\
 & Id+
 \end{aligned}$$

Which gives us

$$- (3) C.0 = Id+$$

Consider

$$\begin{aligned}
 & C.(n+1) \\
 = & \quad \{(0) \text{ in model} \} \\
 & \langle +j : 0 \leq j < n+1 : g.(f.j) \rangle \\
 = & \quad \{ \text{split off } j = n \text{ term} \} \\
 & \langle +j : 0 \leq j < n : g.(f.j) \rangle + g.(f.n) \\
 = & \quad \{ \text{case even.}(f.n), (1) \text{ in model} \} \\
 & \langle +j : 0 \leq j < n : g.(f.j) \rangle + 1
 \end{aligned}$$

Which gives us

$$- (4) C.(n+1) = C.n + 1 \quad \Leftarrow \quad \text{even.}(f.n) \quad , 0 \leq n < N$$

Now consider the other case.

$$\begin{aligned}
 & C.(n+1) \\
 = & \quad \{(0) \text{ in model} \} \\
 & \langle +j : 0 \leq j < n+1 : g.(f.j) \rangle \\
 = & \quad \{ \text{split off } j = n \text{ term} \} \\
 & \langle +j : 0 \leq j < n : g.(f.j) \rangle + g.(f.n) \\
 = & \quad \{ \text{case odd.}(f.n), (2) \text{ in model} \} \\
 & \langle +j : 0 \leq j < n : g.(f.j) \rangle + Id+
 \end{aligned}$$

Which gives us

$$- (5) \quad C.(n+1) = C.n + \text{Id}^+ \quad \Leftarrow \quad \text{odd.}(f.n) \quad , \quad 0 \leq n < N$$

This completes our model.

Rewrite postcondition in terms of model.

$$r = C.n$$

Invariants.

$$P0 : r = C.n \wedge n=N$$

$$P1 : 0 \leq n \leq N$$

Termination.

We observe that

$$P0 \wedge P1 \wedge n=N \quad \Rightarrow \quad \text{Post}$$

Establishing the invariants.

To establish P0 we need to bind r to the value of C.n, for some n. Theorem (3) in our model gives us the value of C.n when n=0. So the following assignment establishes P0 and also P1.

$$n, r := 0, \text{Id}^+$$

Guard.

$$n \neq N$$

Variant.

$$N-n$$

Loop body.

$$\begin{aligned}
 & (n, r := n+1, E).P0 \\
 = & \quad \{ \text{textual substitution} \} \\
 & E = C.(n+1) \\
 = & \quad \{ \text{Case analysis, even.(f.n) , P1 and } n \neq N \text{ allow us to appeal to (4)} \} \\
 & E = C.n + 1 \\
 = & \quad \{ P0 \} \\
 & E = r + 1
 \end{aligned}$$

Giving us the program fragment

$$[] \text{ even.(f.n) } \rightarrow n, r := n+1, r+1$$

We now look at the other case.

$$\begin{aligned}
 & (n, r := n+1, E).P0 \\
 = & \quad \{ \text{textual substitution} \} \\
 & E = C.(n+1) \\
 = & \quad \{ \text{Case analysis, odd.(f.n) , P1 and } n \neq N \text{ allow us to appeal to (5)} \} \\
 & E = C.n + Id+ \\
 = & \quad \{ P0 \} \\
 & E = r + Id+
 \end{aligned}$$

Giving us the program fragment

$$[] \text{ even.(f.n) } \rightarrow n, r := n+1, r+Id+$$

As $(\text{even}.x \vee \text{odd}.x) \equiv \text{true}$ we have covered all possibilities so we can now write the finished loop program.

Finished program.

$$\begin{aligned}
 & n, r := 0, Id+ \\
 & ; \text{do } n \neq N \rightarrow
 \end{aligned}$$

$$\begin{aligned}
 & \text{If even.(f.n) } \rightarrow n, r := n+1, r+1 \\
 & [] \text{ odd.(f.n) } \rightarrow n, r := n+1, r+Id+ \\
 & \text{Fi}
 \end{aligned}$$

od

$$\{ P0 \wedge P1 \wedge n = N \}$$

Exercises.

Given an array $f[0..1000)$ of int, calculate programs to do the following.

- sum the odd values in f
- determine the smallest positive value in the 2nd half of f .