Chapter 9: The Partitioned Reduction Theorem II

In which we construct the generic solution to the partitioned reduction problem.

We are given an array f[0..N) of int which contains values and we are asked to construct a program to establish the following

$$\{ r = \langle \oplus j : \alpha \le j < \beta : g.(f.j) \rangle , 0 \le \alpha \le \beta \le N \}$$

where

g.x	=	h.x	$\Leftarrow$	Q.x
g.x	=	Id $\oplus$	$\Leftarrow$	not.(Q.x)

Postcondition.

$$\text{Post} : r = \langle \oplus j : \alpha \le j < \beta : g.(f.j) \rangle$$

Strengthen postcondition.

$$\text{Post}' : r = \langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \wedge n = \beta$$

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 = $\langle \oplus j : \alpha \le j < n : g.(f.j) \rangle$, $\alpha \le n \le \beta$

The function g is defined as follows:

* (1) g.x = h.x $\Leftarrow$ Q.x

* (2) g.x = Id$\oplus$ $\Leftarrow$ not.(Q.x)

We now explore some theorems.

Consider

$$
\begin{array}{ll}
 & C.\alpha \\
= & \quad \{(0) \text{ in model } \} \\
 & \langle \oplus j : \alpha \le j < \alpha : g.(f.j) \rangle \\
= & \quad \{ \text{ empty range } \} \\
 & Id\oplus
\end{array}
$$

Which gives us

- (3) $C.\alpha = \quad Id\oplus$

Consider

$C.(n+1)$

$=$ {(0) in model}

$\langle \oplus j : \alpha \le j < n+1 : g.(f.j) \rangle$

$=$ {split off $j = n$ term }

$\langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \oplus g.(f.n)$

$=$ { case $Q.(f.n)$, (1) in model }

$\langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \oplus h.(f.n)$

$=$ { (0) in model }

$C.n \oplus h.(f.n)$

Which gives us

- (4) $C.(n+1) \;=\; C.n \oplus h.(f.n) \quad \Leftarrow \quad Q.(f.n) \quad , \alpha \le n < \beta$

Now consider the other case.

$C.(n+1)$

$=$ {(0) in model}

$\langle \oplus j : \alpha \le j < n+1 : g.(f.j) \rangle$

$=$ {split off $j = n$ term }

$\langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \oplus g.(f.n)$

$=$ { case not.(Q.(f.n)), (2) in model }

$\langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \oplus Id\oplus$

$=$ { (0) in model }

$C.n \oplus Id\oplus$

Which gives us

- (5) $C.(n+1) \quad = \quad C.n \oplus Id\oplus \quad \Leftarrow \quad not.(Q.(f.n)) \quad , \alpha \le n < \beta$

This completes our model.

Rewrite postcondition in terms of model.

Post ' : $r = C.n \wedge n = \beta$

Invariants.

$P0 : r = C.n$

$P1 : \alpha \le n \le \beta$

Termination.

We observe that

$P0 \wedge P1 \wedge n= \beta \quad \Rightarrow \quad Post$

Establishing the invariants.

To establish P0 we need to bind r to the value of C.n, for some n. Theorem (1) in our model gives us the value of C.n when n= α. So the following assignment establishes P0 and also P1.

$n, r := \alpha, Id\oplus$

Guard.

$n \neq \beta$

Variant.

β-n

Loop body.

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

$=$ { textual substitution }

$E = C.(n+1)$

$=$ {Case analysis, $Q.(f.n)$, $P1$ and $n \neq \beta$ allow us to appeal to (4)}

$E = C.n \oplus h.(f.n)$

$=$ {$P0$ }

$E = r \oplus h.(f.n)$

Giving us the program fragment

$[]\ Q.(f.n) \quad \rightarrow \quad n, r := n+1, r \oplus h.(f.n)$

We now look at the other case.

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

$=$ { textual substitution }

$E = C.(n+1)$

$=$ {Case analysis, $not.(Q.(f.n))$, $P1$ and $n \neq \beta$ allow us to appeal to (5)}

$E = C.n \oplus Id\oplus$

$=$ {$P0$ }

$E = r \oplus Id\oplus$

Giving us the program fragment

$[]\ not.(Q.(f.n)) \rightarrow \quad n, r := n+1, r \oplus Id\oplus$

As $(Q.x \vee not.(Q.x)) \equiv$ true we have covered all possibilities so we can now write the finished loop program.

Finished program.

```
n, r := α, Id⊕
;do n ≠ β     →

               if Q.(f.n)        →    n, r := n+1, r ⊕h.(f.n)
               [] not.(Q.(f.n))  →    n, r := n+1, r ⊕Id⊕
               fi

od

{P0 ∧ P1 ∧ n = β}
```