

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

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

where

$$\begin{array}{llll} g.x & = & h.x & \Leftarrow Q.x \\ g.x & = & Id \oplus & \Leftarrow not.(odd.x) \end{array}$$

Postcondition.

$$Post : r = \langle \oplus j : \alpha \leq j < \beta : g.(f.j) \rangle$$

Strengthen postcondition.

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

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.

$$* (0) C.n \quad = \quad \langle \oplus j : \alpha \leq j < n : g.(f.j) \rangle \quad , \ \alpha \leq n \leq \beta$$

The function g is defined as follows:

$$\begin{array}{llll} * (1) g.x & = & h.x & \Leftarrow Q.x \\ * (2) g.x & = & Id \oplus & \Leftarrow not.(Q.x) \end{array}$$

We now explore some theorems.

Consider

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

Which gives us

$$- (3) C.\alpha = \text{Id} \oplus$$

Consider

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

Which gives us

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

Now consider the other case.

$$\begin{aligned}
 & C.(n+1) \\
 = & \{ (0) \text{ in model} \} \\
 & \langle \oplus j : \alpha \leq j < n+1 : g.(f.j) \rangle \\
 = & \{ \text{split off } j = n \text{ term} \} \\
 & \langle \oplus j : \alpha \leq j < n : g.(f.j) \rangle \oplus g.(f.n) \\
 = & \{ \text{case not.}(Q.(f.n)), (2) \text{ in model} \} \\
 & \langle \oplus j : \alpha \leq j < n : g.(f.j) \rangle \oplus \text{Id} \oplus \\
 = & \{ (0) \text{ in model} \} \\
 & C.n \oplus \text{Id} \oplus
 \end{aligned}$$

Which gives us

$$- (5) C.(n+1) = C.n \oplus Id \oplus \Leftarrow \text{not.}(Q.(f.n)) , \alpha \leq n < \beta$$

This completes our model.

Rewrite postcondition in terms of model.

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

Invariants.

$$P0 : r = C.n$$

$$P1 : \alpha \leq n \leq \beta$$

Termination.

We observe that

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

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 $n = \alpha$. So the following assignment establishes P0 and also P1.

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

Guard.

$$n \neq \beta$$

Variant.

$$\beta - n$$

Loop body.

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

Giving us the program fragment

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

We now look at the other case.

$$\begin{aligned}
& (n, r := n+1, E).P0 \\
= & \quad \{ \text{textual substitution} \} \\
& E = C.(n+1) \\
= & \quad \{ \text{Case analysis, not.}(Q.(f.n)) , P1 \text{ and } n \neq \beta \text{ allow us to appeal to (5)} \} \\
& E = C.n \oplus Id \oplus \\
= & \quad \{ P0 \} \\
& E = r \oplus Id \oplus
\end{aligned}$$

Giving us the program fragment

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

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

Finished program.

$$\begin{aligned}
& n, r := \alpha, Id \oplus \\
& ; \text{do } n \neq \beta \rightarrow \\
& \quad \text{if } Q.(f.n) \rightarrow n, r := n+1, r \oplus h.(f.n) \\
& \quad [] \text{not.}(Q.(f.n)) \rightarrow n, r := n+1, r \oplus Id \oplus \\
& \quad \text{fi} \\
& \text{od} \\
& \{ P0 \wedge P1 \wedge n = \beta \}
\end{aligned}$$

