The Partitioned Reduction Theorem.

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.

Precondition	f[0..N) contains values
Postcondition	$r = \langle + j : 0 \le j < N : g.(f.j) \rangle$

where

$$g.x = 1 \Leftarrow even.x$$
$$g.x = 0 \Leftarrow odd.x$$

Postcondition.

$$Post : r = \langle + j : 0 \le j < N : g.(f.j)\rangle$$

Strengthen postcondition.

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

Domain modelling.

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

* (1) $g.x = 1 \Leftarrow even.x$
* (2) $g.x = Id_+ \Leftarrow odd.x$

Consider

$$C.0$$
$$= \quad \{(0) \text{ in model }\}$$
$$\langle + j : 0 \le j < 0 : g.(f.j)\rangle$$
$$= \quad \{ \text{ empty range } \}$$
$$Id_+$$

Which gives us

\- (3) $C.0 = Id_+$

Consider

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

Which gives us

- (4) $C.(n+1) \;=\; C.n + 1 \quad \Leftarrow \quad even.(f.n) \quad , 0 \le n < N$

Also consider

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

Which gives us

- (5) $C.(n+1) \;=\; C.n + Id+ \quad \Leftarrow \quad odd.(f.n) \quad , 0 \le n < N$

Rewrite postcondition in terms of model.

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

Choose invariants.

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

Establishing the invariants.

n, r := 0, Id+

Guard.

$n \ne N$

Variant.

N-n

Loop body.

(n, r := n+1, E).P0
= { textual substitution }
E = C.(n+1)
= {Case analysis, even.(f.n) , P1 and n ≠ N allow us to appeal to (4)}
E = C.n + 1
= {P0 }
E = r + 1

Giving us the program fragment

[] even.(f.n) → n, r := n+1, r+1

We now look at the other case.

(n, r := n+1, E).P0
= { textual substitution }
E = C.(n+1)
= {Case analysis, odd.(f.n) , P1 and n ≠ N allow us to appeal to (5)}
E = C.n + Id+
= {P0 }
E = r + Id+

Giving us the program fragment

[] odd.(f.n) → n, r := n+1, r+Id+

Finished algorithm.

```
n, r := 0, Id+
;do n ≠ N     →
      if even.(f.n)   → n, r := n+1, r+1
      [] odd.(f.n)    → n, r := n+1, r+Id+
      fi
 od
```

We now look at the abstract version of this problem and calculate a generic solution. We are given f[α..β) of some particular type, which contains values, and we are asked to construct a program to achieve the following postcondition.

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

where

g.x	=	h.x	⇐	Q.x
g.x	=	Id⊕	⇐	¬Q.x

Strengthen postcondition.

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

Domain modelling.

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

* (1) g.x = h.x ⇐ Q.x
* (2) g.x = Id ⊕ ⇐ ¬Q.x

Consider

C.α
= {(0)}
$\langle \oplus j : \alpha \le j < \alpha : g.(f.j) \rangle$
= { Empty Range }
Id⊕

which gives us

- (3) C.α = Id⊕

Consider

C.(n+1)
= {(0)}
$\langle \oplus j : \alpha \le j < n+1 : g.(f.j) \rangle$
= { 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) }
$\langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \oplus h.(f.n)$
= {(0)}
C.n ⊕ h.(f.n)

Which gives us

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

Also consider

$$
\begin{array}{ll}
 & C.(n+1) \\
= & \qquad \{(0)\} \\
 & \langle \oplus j : \alpha \le j < n+1 : g.(f.j) \rangle \\
= & \qquad \{\text{ Split off } j{=}n \text{ term }\} \\
 & \langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \oplus g.(f.n) \\
= & \qquad \{\text{ Case } \neg Q.(f.n) \text{ (2) }\} \\
 & \langle \oplus j : \alpha \le j < n : g.(f.j) \rangle \oplus Id\oplus \\
= & \qquad \{(0)\} \\
 & C.n \oplus Id\oplus
\end{array}
$$

Which gives us

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

This completes our model.

Rewrite postcondition using model.

$r = C.n \;\wedge\; n = \beta$

Choose invariants.

P0: $r = C.n$
P1: $\alpha \le n \le \beta$

Guard.

$n \ne \beta$

Establish invariants.

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

Variant.

$\beta - n$

Loop body.

$$
\begin{array}{ll}
 & (n, r := n+1, E).P0 \\
= & \qquad \{ \text{ Textual Substitution } \} \\
 & E = C.(n+1) \\
= & \qquad \{ \text{ Case } Q.(f.n) \text{ (4) } \} \\
 & E = C.n \oplus h.(f.n) \\
= & \qquad \{ \text{ P0 } \} \\
 & E = r \oplus h.(f.n)
\end{array}
$$

Considering the other case

$$
\begin{array}{ll}
 & (n, r := n+1, E).P0 \\
= & \qquad \{ \text{ Textual Substitution } \} \\
 & E = C.(n+1) \\
= & \qquad \{ \text{ Case } \neg Q.(f.n) \text{ (5) } \} \\
 & E = C.n \oplus Id\oplus \\
= & \qquad \{ \text{ P0 } \} \\
 & E = r \oplus Id\oplus
\end{array}
$$

Finished algorithm.

```
n, r := α, Id⊕
; do n ≠ β  →
        if Q.(f.n) →  n, r := n+1, r ⊕ h.(f.n)
        [] ¬Q.(f.n) →  n, r := n+1, r ⊕ Id⊕
        fi
  od
```