Chapter 46: Shortest Common Super-sequence

Given X[0..M) and Y[0..N) both of int. Z is a super-sequence of X and Y if both X and Y are subsequences of Z. We are asked to find the length of the shortest common super-sequence of X and Y.

(0) Scs.m.0 = m

This is where Y is empty.

(1) Scs.0.n = n

This is where X is empty.

When neither sequence is empty we have the following 2 cases,

(2) Scs.(m+1).(n+1) = Scs.m.n + 1 When X.m = Y.n

(3) Scs.(m+1).(n+1) = (Scs.(m+1).n + 1) ↓ (Scs.m.(n+1) + 1) When X.m ≠ Y.n

So this function gives the minimum length of the common super-sequence. We want to compute Scs.M.N

Post : r = Scs.M.N

Propose h[0..N] of int.

Invariants.

P0: ⟨∀ i : 0 ≤ i ≤ N : h.i = Scs.m.i ⟩

P1: 0 ≤ m ≤ M

Establish Invariants P0 and P1.

```
 i, m := 0, 0
;do i <> N+1 ->
        i, h.i := i+1, i
od
```

Note.

P0 ∧ P1 m = M => h.N = Scs.M.N

Guard.

m ≠ M

vf.

M-m

Loop structure.

$$
\begin{array}{ll}
 & (m := m+1).P0 \\
= & \quad \{\text{text sub.}\} \\
 & \langle \forall\, i : 0 \le i \le N \ : h.i = Scs.m.i \rangle
\end{array}
$$

We are not going to be able to achieve this in a single step. So we propose

Invariants.

Q0: $\langle \forall\, i : 0 \le i \le n \ : h.i = Scs.(m+1).i \rangle \ \wedge \ \langle \forall\, i : n < i \le N \ : h.i = Scs.m.i \rangle$

Q1: $0 \le n \le N$

Establish Invariants Q0 ∧ Q1

$$
\begin{array}{ll}
 & (n := 0).Q0 \\
= & \quad \{\text{text sub.}\} \\
 & \langle \forall\, i : 0 \le i \le 0 \ : h.i = Scs.(m+1).i \rangle \ \wedge \ \langle \forall\, i : 0 < i \le N \ : h.i = Scs.m.i \rangle \\
= & \quad \{\text{1-point}\} \\
 & h.0 = Scs.(m+1).0 \ \wedge \ \langle \forall\, i : 0 < i \le N \ : h.i = Scs.m.i \rangle \\
= & \quad \{ P0 \} \\
 & h.0 = Scs.(m+1).0 \\
= & \quad \{(0)\} \\
 & h.0 = m+1
\end{array}
$$

So Q0 ∧ Q1 can be established by

n, h.0 := 0, m+1

Note.

Q0 ∧ Q1 ∧ n = N => (m := m+1).Q0

Guard.

$n \neq N$

vf.

N-n

Loop body.

(n := n+1).Q0

= {text sub.}

$\langle \forall\ i : 0 \le i \le n+1\ : h.i = Scs.(m+1).i \rangle \wedge \langle \forall\ i : n+1 < i \le N\ : h.i = Scs.m.i \rangle$

= {split off n+1 term }

$\langle \forall\ i : 0 \le i \le n\ : h.i = Scs.(m+1).i \rangle \wedge$

$\langle \forall\ i : n+1 < i \le N\ : h.i = Scs.m.i \rangle \wedge$

h.(n+1) = Scs.(m+1).(n+1)

= {Q0}

h.(n+1) = Scs.(m+1).(n+1)

= {case $X.m \ne Y.n$ }

h.(n+1) = (Scs.(m+1).n + 1) ↓ (Scs.m.(n+1) + 1)

= {Q0}

h.(n+1) = (h.n + 1) ↓(h.(n+1) + 1)

Other case

h.(n+1) = Scs.(m+1).(n+1)

= {case X.m = Y.n (2) }

h.(n+1) = Scs.m.n + 1

= {Propose Q2: a = Scs.m.n }

h.(n+1) = a + 1

Inner loop.

```
n, h.0 := 0, m+1
;do n ≠ N —>

        If X.m ≠ Y.n  —> n, h.(n+1) := n+1,  (h.n + 1) ↓ (h.(n+1) + 1)
        [] X.m = Y.n —> n, h.(n+1) := n+1, a+1
        fi

od
```

Now we must establish and maintain Q2.

Recall Q2: a = Scs.m.n

Establish Q2.

```
        (n, h.0, a := 0, m+1, E).Q2
=               { Text Sub. }
        E = Scs.m.0
=               { (0) }
        E = m
```

Maintain Q2.

Consider

```
        (n, h.(n+1), a  := n+1,  (h.n + 1) ↓(h.(n+1) + 1), E ).Q2
=               { Text Sub. }
        E = Scs.m.(n+1)
=               { Q0 }
        E = h.(n+1)
```

Final algorithm.

```
         i, m := 0, 0
        ; do i ≠ N+1 ->
                i, h.i := i+1, i
        od

        ; do m ≠ M ->

                n, h.0, a := 0, m+1, m

                ; do n ≠ N —>

                        If X.m ≠ Y.n  —> n, h.(n+1), a  := n+1,  (h.n + 1) ↓ (h.(n+1) + 1), h.(n+1)
                        [] X.m = Y.n —> n, h.(n+1), a  := n+1, a+1, h.(n+1)
                        fi

                od

                ; m := m+1
        od

        { h.N = Scs.M.N }
```