To my fellow teachers of programming

This note is meant to report to you a trick that may streamline the reasoning about some of our very elementary programming exercises. I will explain the trick by means of an example.

Given N, $0 \le N$, and boolean array $b[0..N)$, we define L as follows:

$$L.n \equiv (\exists i : 0 \le i < n : b.i) \qquad (0 \le n \le N) .$$

With postcondition R and invariant P,

R: $\quad x \equiv L.N$

P: $\quad x \equiv L.n \quad ,$

we are invited to find termination condition C — the negation of the guard — such that

$$P \wedge C \Rightarrow R \quad .$$

We are used to the choice $n = N$ for C, and that is okay, but choice $L.n \equiv L.N$ is nicer because it is weaker. Therefore we investigate

$$
\begin{array}{ll}
 & L.n \equiv L.N \\
\equiv & \quad \{\text{pred. calc.}\} \\
 & (L.n \Rightarrow L.N) \wedge (L.N \Rightarrow L.n) \\
\equiv & \quad \{\text{because —as usual— } n \le N, \ \ L.n \Rightarrow L.N\} \\
 & L.N \Rightarrow L.n \\
\Leftarrow & \quad \{\text{pred. calc.}\} \\
 & n = N \vee L.n \\
\equiv & \quad \{P\} \\
 & n = N \vee x \quad ,
\end{array}
$$

and thus we have derived our beloved termination condition in one go.

If we choose to solve the above programming problem using a tail invariant, the reasoning is even more smooth. Define K by

$$K.n \;\equiv\; (\exists i : n \leq i < N : b.i)$$

Then R and P are given by

R: $K.0 \;\equiv\; x$

P: $K.0 \;\equiv\; x \vee K.n$,

and for the termination condition we may choose $x \equiv x \vee K.n$. We calculate

$$
\begin{array}{ll}
 & x \equiv x \vee K.n \\
\equiv & \quad \{\text{pred. calc.}\} \\
 & K.n \Rightarrow x \\
\equiv & \quad \{\text{pred. calc}\} \\
 & \neg K.n \vee x \\
\Leftarrow & \quad \{\text{def. } K\} \\
 & n = N \vee x \quad .
\end{array}
$$

* * *

There are quite some other examples in our elementary course that are amenable to the above streamlining. If you encounter a particular nice one, please report it.

Yours ever,

W.H.J. Feijen
2 October 1992