A Dutch derivation of the program imp

In one of his first lectures at the University of Austin, professor Edsger W. Dijkstra addressed the question of deriving a program —he called it imp— satisfying the functional specification

$$
\begin{array}{l}
|[\ N: \text{int}\ \{N \geq 0\}\ ;\ X(i: 0 \leq i < N): \underline{\text{array}}\ \underline{\text{of}}\ \text{bool} \\
;\ |[\ w: \text{bool} \\
\quad ;\ \text{imp} \\
\quad \{w \equiv (\underline{A} i,j : 0 \leq i \leq j < N : \neg X(i) \vee X(j))\} \\
\quad]| \\
]| .
\end{array}
$$

I haven't seen his solution. The solution given below is the result of pure calculation. The emerging program is new to me.

*

For imp we choose an inner block with a repetitive construct maintaining the relation $P0 \wedge P1$, defined by

$$P0: \quad 0 \leq n \leq N$$
$$P1: \quad w \equiv (\underline{A} i,j : 0 \leq i \leq j < n : \neg X(i) \vee X(j)) .$$

The computation begins with "$n, w := 0, \text{true}$", establishing $P0 \wedge P1$, and it terminates with $n = N$ thus establishing the desired postcondition. In the mean time n is increased by 1, repeatedly, and the required adjustment of w follows from

$$
\begin{array}{ll}
 & (\underline{A} i,j : 0 \leq i \leq j < n+1 : \neg X(i) \vee X(j)) \\
= & \quad \{0 \leq n < N, \text{ from } P0 \wedge n \neq N\} \\
 & (\underline{A} i,j : 0 \leq i \leq j < n : \neg X(i) \vee X(j)) \\
 & \ \wedge\ (\underline{A} i : 0 \leq i \leq n : \neg X(i) \vee X(n)) \\
=
\end{array}
$$

$$
\begin{array}{ll}
= & \{P1\} \\
& w \;\wedge\; (\underline{A} i : 0 \le i \le n : \neg X(i) \vee X(n)) \\
= & \{0 \le n < N,\quad \neg X(n) \vee X(n) \equiv true\} \\
& w \;\wedge\; (\underline{A} i : 0 \le i < n : \neg X(i) \vee X(n)) \\
= & \{\text{predicate calculus}\} \\
& w \;\wedge\; ((\underline{A} i : 0 \le i < n : \neg X(i)) \vee X(n)) \\
= & \{P2, \text{ see below}\} \\
& w \;\wedge\; (h \vee X(n)) .
\end{array}
$$

The interest in the maintenance of the additional

$P2: \quad h \equiv (\underline{A} i : 0 \le i < n : \neg X(i))$

followed from the above calculation.
For imp we obtain

$$
\begin{array}{l}
|[\, n : int;\; h : bool \\
;\; n, h, w := 0, true, true \\
;\; \underline{do}\ n \ne N \\
\quad \rightarrow\ w := w \wedge (h \vee X(n));\; h := h \wedge \neg X(n);\; n := n+1 \\
\quad \underline{od} \\
]| ,
\end{array}
$$

or – massaging the repeatable statement a little bit –

$$
\begin{array}{l}
|[\, n : int;\; h : bool \\
;\; n, h, w := 0, true, true \\
;\; \underline{do}\ n \ne N \\
\quad \rightarrow\ \underline{if}\ X(n) \rightarrow h := false \\
\qquad\quad [\!]\ \neg X(n) \rightarrow w := w \wedge h \\
\qquad\quad \underline{fi} \\
\qquad ;\; n := n+1 \\
\quad \underline{od} \\
]| .
\end{array}
$$

The latter code expresses quite nicely that each element of X is inspected exactly once.

24 september 1984,

W.H.J. Feijen