Mathematical Induction expressed in a dummy-free notation

On the ETAC's session of August 21, Edsger W. Dijkstra posed the question of how to formulate mathematical induction in the notation of relational calculus. Just for our files, we record the outcome of that afternoon's exploration. (We expect EWD to further elaborate on it and put it into perspective.)

The principle of mathematical induction is that for well-founded binary relation $\triangleleft$ we have

for any P,

$$(0)\qquad \begin{array}{l} (\underline{A}x :: P.x) \\ \Leftarrow \\ (\underline{A}x :: P.x \;\vee\; (\underline{E}y : y \triangleleft x : \neg P.y)) \end{array} \quad .$$

Now, for the benefit of translating (0) into a relational form we introduce the binary relation (or predicate on a "doubled" space) R, given by

$$R: \qquad (\underline{A}x, x' :: x'Rx \equiv P.x) \; .$$

For the consequent of (0) we then obtain

$$\begin{array}{ll} & (\underline{A}x :: P.x) \\ = & \{\text{pred. calc.}\} \end{array}$$

$$
\begin{array}{ll}
 & (\underline{A}x,x' :: P.x) \\
= & \quad \{\text{definition of } R\} \\
 & (\underline{A}x,x' :: x' R x) \\
= & \quad \{\text{definition of square brackets}\} \\
 & [R] \quad .
\end{array}
$$

For the antecedent of (0) we obtain

$$
\begin{array}{ll}
 & (\underline{A}x :: P.x \ \vee \ (\underline{E}y : y \lhd x : \neg P.y)) \\
= & \quad \{\text{pred. calc.}\} \\
 & (\underline{A}x :: P.x \ \vee \ (\underline{E}y :: \neg P.y \ \wedge \ y \lhd x)) \\
= & \quad \{\text{pred. calc. and} \\
 & \quad \ \ \text{definition of } R\} \\
 & (\underline{A}x,x' :: x' R x \ \vee \ (\underline{E}y :: x'(\neg R)y \ \wedge \ y \lhd x)) \\
= & \quad \{\text{definition of } ;\} \\
 & (\underline{A}x,x' :: x' R x \ \vee \ x' (\neg R ; \lhd) x) \\
= & \quad \{\text{definition of } \vee \text{ between relations}\} \\
 & (\underline{A}x,x' :: x' (R \vee (\neg R ; \lhd)) \ x) \\
= & \quad \{\text{definition of square brackets}\} \\
 & [R \vee (\neg R ; \lhd)] \quad .
\end{array}
$$

Predicates like R that do not "depend on their left argument" are usually called postconditions, and with that nomenclature mathematical induction can now be phrased as

for all postconditions R,

$$[R] \Leftarrow [R \vee (\neg R ; \lhd)] \quad .$$

* * *

Note that for this occasion we haven't formulated (0)'s antecedent as usual, viz.

$$(\underline{A}x:: P.x \Leftarrow (\underline{A}y: y \lhd x: P.y)) ,$$

since the form with the existential quantifier is better geared towards the definition of the ; .

Finally note that the definition of R is no rabbit: in fact it was conscientiously constructed by Rob Hoogerwoord so as to make the introduction of the ; as smooth as possible.

W.H.J. Feijen,

Eindhoven, 10 Sept. 1990

P.S. R being a postcondition can be expressed in the relational notation as

$$[\,true ; R \equiv R\,] \quad .$$