A resettable variable is defined as follows. There are three new names: value (of type X), set (a procedure with one parameter of type X), and reset (a program). Here are the axioms:

\[
\text{value}' = x \iff \text{set } x \\
\text{value}' = \text{value} \iff \text{set } x. \text{ reset} \\
\text{reset. reset } = \text{ reset}
\]

Implement this data structure, with proof.

Let value: X be a user's variable, and let old: X be an implementer's variable.

\[
\text{set } = \langle x: X \mapsto \text{old}: = \text{value}. \text{value}: = x \rangle \\
\text{reset } = \text{value}: = \text{old}
\]

Proof:

\[
\begin{align*}
(value' = x & \iff \text{set } x) \\
& = (value' = x \iff \text{old}: = \text{value}. \text{value}: = x) \\
& = (value' = x \iff \text{old} = \text{value} \land \text{value}' = x) \\
& = \top \\
&\quad (value' = \text{value} \iff \text{set } x. \text{ reset}) \\
& = (value' = \text{value} \iff \text{old}: = \text{value}. \text{value}: = x. \text{value}: = \text{old}) \\
& = (value' = \text{value} \iff \text{old}' = \text{value}' = \text{value}) \\
& = \top \\
&\quad (\text{reset. reset } = \text{ reset}) \\
& = (\text{value}: = \text{old}. \text{value}: = \text{old } = \text{value}: = \text{old}) \\
& = \top
\end{align*}
\]