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

\[
\begin{align*}
\text{value}' &= x \iff \text{set } x \\
\text{value}' &= \text{value} \iff \text{set } x. \text{ reset} \\
\text{reset. reset} &= \text{reset}
\end{align*}
\]

Implement this data structure, with proof.

§ Let \textit{value}: \(X\) be a user's variable, and let \textit{old}: \(X\) be an implementer's variable.

\[
\begin{align*}
\text{set} &= \langle x: X \rightarrow \text{old} := \text{value}. \text{ value} := x \rangle \\
\text{reset} &= \text{value} := \text{old}
\end{align*}
\]

Proof:

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