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

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

Implement this data structure, with proof.

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

\begin{align*}
\texttt{set} & = \langle \texttt{x}: X \mapsto \texttt{old}:= \texttt{value}. \ \texttt{value}:= x \rangle \\
\texttt{reset} & = \texttt{value}:= \texttt{old}
\end{align*}

Proof:

\begin{align*}
(\texttt{value}'=x & \iff \texttt{set} x) \\
= & (\texttt{value}'=x \iff \texttt{old}:= \texttt{value}. \ \texttt{value}:= x) \\
= & (\texttt{value}'=x \iff \texttt{old}:=\texttt{value} \wedge \texttt{value'}=x) \\
= & T \\
(\texttt{value}'=\texttt{value} & \iff \texttt{set} \ x. \ \texttt{reset}) \\
= & (\texttt{value}'=\texttt{value} \iff \texttt{old}:= \texttt{value}. \ \texttt{value}:= x. \ \texttt{value}:= \texttt{old}) \\
= & (\texttt{value}'=\texttt{value} \iff \texttt{old}:=\texttt{value}'=\texttt{value}) \\
= & T \\
(\texttt{reset} \ . \ \texttt{reset} & = \ \texttt{reset}) \\
= & (\texttt{value}:= \texttt{old}. \ \texttt{value}:= \texttt{old} \ \Rightarrow \ \texttt{value}:= \texttt{old}) \\
= & T
\end{align*}