What is wrong with defining local variable declaration as follows:

\[ \text{var } x: T \cdot P = \forall x: T \cdot \exists x': T \cdot P \]

§ Programs are implementable. Consider the program

\text{var } x: \text{int } y := x

with global integer variables \( y \) and \( z \). Using the suggested definition,

\[ \forall x \cdot \exists y'. x' = x \wedge y' = x \wedge z' = z \]

\[ \bot \]

we get something unimplementable. If we had used the proper definition

\text{var } x: \text{int } y := x

\[ \exists x, x'. x' = x \wedge y' = x \wedge z' = z \]

\[ z' = z \]

we get something implementable, as we should. Less importantly, with the new definition, the \text{null} type is implementable:

\text{var } x: \text{null } x := 0

\[ \forall x: \text{null} \cdot \exists x': \text{null} \cdot x' = 0 \wedge y' = y \wedge z' = z \]

\[ T \]

If we had used the proper definition

\text{var } x: \text{null } x := 0

\[ \exists x: \text{null} \cdot \exists x': \text{null} \cdot x' = 0 \wedge y' = y \wedge z' = z \]

\[ \bot \]

the \text{null} type is unimplementable, as it should be. Note that the assignment is out-of-range.