A program list is a list with an associated index, and the following operations: item gives the value of the indexed item; set \( x \) changes the value of the indexed item to \( x \); goLeft moves the index one item to the left; goRight moves the index one item to the right.

(a) Design axioms for a doubly infinite program list.

Let \( L \) mean that all items to the left of the indexed item remain the same.

Let \( R \) mean that all items to the right of the indexed item remain the same.

\[
\begin{align*}
ok &= L \land item' = item \land R = goLeft. goRight = goRight. goLeft \\
set x &= L \land item' = x \land R \\
goLeft. L \land item' = item &= L. goLeft \\
goRight. item' = item \land R &= R. goRight \\
L. L &= L \\
R. R &= R
\end{align*}
\]

(b) Using your theory from part (a), prove

\[
\begin{align*}
goLeft. set 3. goRight. set 4. goLeft & \implies item' = 3 \\
& \equiv goLeft. L \land item' = 3 \land R. goRight. L \land item' = 4 \land R. goLeft \\
& \implies goLeft. item' = 3. goRight. L. goLeft \\
& \equiv goLeft. item' = 3. goRight. goLeft. L \land item' = item \\
& \equiv goLeft. item' = 3. goRight. goLeft. item' = item \\
& \equiv item' = 3 \quad \text{definition of sequential composition twice}
\end{align*}
\]