A program list is a list with an associated index, and the following operations: \textit{item} gives the value of the indexed item; \textit{set} \(x\) changes the value of the indexed item to \(x\); \textit{goLeft} moves the index one item to the left; \textit{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.
\[
ok = L \land \text{item} = \text{item} \land R = \text{goLeft. goRight} = \text{goRight. goLeft}
\]
\[
\text{set} x = L \land \text{item} = x \land R
\]
\[
\text{goLeft. } L \land \text{item} = \text{item} = L. \text{goLeft}
\]
\[
\text{goRight. } \text{item} = \text{item} \land R = R. \text{goRight}
\]
\[
L. L = L
\]
\[
R. R = R
\]

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

\[
\text{goLeft. set} 3. \text{goRight. set} 4. \text{goLeft} \implies \text{item} = 3
\]

\[
\text{goLeft. set} 3. \text{goRight. set} 4. \text{goLeft}
\]
\[
= \text{goLeft. } L \land \text{item} = 3 \land R. \text{goRight. } L \land \text{item} = 4 \land R. \text{goLeft}
\]
\[
\implies \text{goLeft. } \text{item} = 3. \text{goRight. } L. \text{goLeft}
\]
\[
= \text{goLeft. } \text{item} = 3. \text{goRight. } \text{goLeft. } L \land \text{item} = \text{item}
\]
\[
\implies \text{goLeft. } \text{item} = 3. \text{goRight. } \text{goLeft. } \text{item} = \text{item}
\]
\[
= \text{goLeft. } \text{item} = 3. \text{item} = \text{item}
\]
\[
definition of dependent composition twice
\]
\[
= \text{item} = 3
\]