An insertion list is a data structure similar to a list, but with an associated insertion point.

[ ...; 4 ; 7 ; 1 ; 0 ; 3 ; 8 ; 9 ; 2 ; 5 ; ... ]

↑

insert point

*insert* puts an item at the insertion point (between two existing items), leaving the insertion point at its right. *erase* removes the item to the left of the insertion point, closing up the list. *item* gives the item to the left of the insertion point. *forward* moves the insertion point one item to the right. *back* moves the insertion point one item to the left.

(a) Design axioms for a doubly-infinite data-insertion list.

(b) Design axioms for a doubly-infinite program-insertion list.

§ Here is a weak theory.

\[
\begin{align*}
\text{item}' &= x \iff \text{insert } x \\
\text{item}' &= \text{item} \iff F \lor (\text{back. forward}) \\
\text{forward. back} &= \text{back. forward} = \text{ok} \\
F &= \text{ok} \lor (\exists x \cdot \text{insert } x) \lor \text{forward. erase } \lor \text{forward. back} \lor (F, F) \\
B &= \text{ok} \lor (\exists x \cdot \text{insert } x) \lor \text{erase } \lor \text{forward. back} \lor (B, B)
\end{align*}
\]

Here is a strong theory.

\[
\begin{align*}
\text{ok} &= F \land B \\
\text{forward. back} &= \text{back. forward} = \text{insert } x \cdot \text{erase} \\
\text{insert } x &= (\text{back. F}) \land \text{item}' = x \land B \\
F &= \text{ok} \lor (\exists x \cdot \text{insert } x) \lor \text{forward. erase } \lor \text{forward. back} \lor (F, F) \\
B &= \text{ok} \lor (\exists x \cdot \text{insert } x) \lor \text{erase } \lor \text{forward. back} \lor (B, B)
\end{align*}
\]

(c) Design axioms for a finite data-insertion list.

(d) Design axioms for a finite program-insertion list.