In Subsection 7.1.3 we designed a program-stack theory so weak that we could add axioms to count pushes and pops without inconsistency. Design a similarly weak data-stack theory.

\[
\begin{align*}
\text{stack} & \neq \text{null} \\
\text{push} \text{ stack } X : \text{ stack} \\
\text{top} (\text{push } s \ x) & = x \\
\text{top} (\text{balance } s) & = \text{top } s \\
\text{s, pop} (\text{balance} (\text{push } s \ X)) : \text{ balance } s
\end{align*}
\]

Now we can add

\[
\begin{align*}
\text{count} : \text{ stack} \rightarrow \text{nat} \\
\text{count} (\text{push } s \ x) & = \text{count } s + 1 \\
\text{count} (\text{pop } s) & = \text{count } s + 1
\end{align*}
\]

We don't need an empty stack to start; we can just take note of the count at the start and subtract that whenever we want the relative count. Here's an implementation.

\[
\begin{align*}
\text{stack} & = [\text{nat}; \ast X] \\
\text{push} & = \langle s: \text{ stack} \rightarrow \langle x: X \rightarrow [s \ 0 + 1] \ast s[1;..\#s] \ast [x] \rangle \rangle \\
\text{pop} & = \langle s: \text{ stack} \rightarrow [s \ 0 + 1] \ast s[1;..\#s-1] \rangle \\
\text{top} & = \langle s: \text{ stack} \rightarrow s(\#s-1) \rangle \\
\text{count} & = \langle s: \text{ stack} \rightarrow s \ 0 \rangle
\end{align*}
\]

To prove the implementation, we need to define \text{balance}

\[
\begin{align*}
\text{balance} & = \langle s: \text{ stack} \rightarrow \mathcal{S}: \text{ stack} \ s[1;..\#s] = t[1;..\#t] \rangle
\end{align*}
\]

but we don't need to implement it.