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.

\[\text{stack \neq null}\]

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

Now we can add

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

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.

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

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

\[\text{balance} = \langle s: \text{stack} : \$t: \text{stack} : s [1;..#s]=t [1;..#t] \rangle \]

but we don't need to implement it.