(weak data-stack) 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.

After trying the question, scroll down to the solution.
§

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

\[ push \ stack \ X: \ stack \]

\[ \text{top} (push \ s \ x) = x \]

\[ \text{top} (\text{balance} \ s) = \text{top} \ s \]

\[ s, \text{pop} (\text{balance} (push \ s \ X)): \text{balance} \ s \]

Now we can add

\[ \text{count}: \ stack \rightarrow \text{nat} \]

\[ \text{count} (push \ s \ x) = \text{count} \ s + 1 \]

\[ \text{count} (\text{pop} \ s) = \text{count} \ s + 1 \]

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.

\[ stack = [\text{nat}; *X] \]

\[ push = \langle s: \text{stack}; \langle x: X; [s \ 0 + 1]; [s[1;..#s]; [x]] \rangle \rangle \]

\[ pop = \langle s: \text{stack}; [s \ 0 + 1]; [s[1;..#s-1]] \rangle \]

\[ top = \langle s: \text{stack}; s(#s-1) \rangle \]

\[ count = \langle s: \text{stack}; s \ 0 \rangle \]

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

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

but we don’t need to implement it.