A stack, according to our axioms, has an unlimited capacity to have items pushed onto it. A limited-stack is a similar data structure but with a limited capacity to have items pushed onto it.

(a) Design axioms for a limited-data-stack.
§ I suppose \( \text{limit} \) is a given natural, and \( X \) is a given bunch. I introduce the following new syntax: \( \text{stack} \), \( \text{push} \), \( \text{pop} \), \( \text{top} \), \( \text{empty} \), \( \text{size} \). Here are the axioms: Let \( s, t: \text{stack} \) and \( x, y: X \). Then

\[
\begin{align*}
\text{empty}: \text{stack} \\
\text{size empty} &= 0 \\
\text{size } s > 0 &\implies \text{pop } s: \text{stack} \\
\text{size } s > 0 &\implies \text{size } (\text{pop } s) = \text{size } s - 1 \\
\text{size } s < \text{limit} &\implies \text{push } s x: \text{stack} \\
\text{size } s < \text{limit} &\implies \text{size } (\text{push } s x) = \text{size } s + 1 \\
\text{size } s < \text{limit} &\implies \text{push } s x \neq \text{empty} \\
\text{size } s < \text{limit} &\implies (\text{push } s x = \text{push } t y \iff s=t \land x=y) \\
\text{size } s < \text{limit} &\implies \text{pop } (\text{push } s x) = s \\
\text{size } s < \text{limit} &\implies \text{top } (\text{push } s x) = x
\end{align*}
\]

(b) Design axioms for a limited-program-stack.
§ I suppose \( \text{limit} \) is a given natural, and \( X \) is a given bunch. I introduce the following new syntax: \( \text{push} \), \( \text{pop} \), \( \text{top} \), \( \text{mkempty} \), \( \text{size} \). Here are the axioms: Let \( x: X \). Then

\[
\begin{align*}
\text{top’} &= x \land \text{size’} = \text{size} + 1 \iff \text{size} < \text{limit} \land \text{push } x \\
\text{size’} &= \text{size} - 1 \iff \text{size} > 0 \land \text{pop} \\
\text{size’} &= 0 \iff \text{mkempty} \\
\text{ok} &\iff \text{size} < \text{limit} \land (\text{push } x. \text{pop})
\end{align*}
\]

(c) Can the limit be 0?
§ Sure. Why not? Then the empty stack is also full, and no operations are possible. But there's no logical problem.