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

(a) Design axioms for a limited-data-queue.

I'm introducing new name \textit{full}, which tells whether a queue is full (of course). This allows an implementation in which \textit{full} might say \(T\) for a queue with 3 long items, and \(\bot\) for another queue with 3 short items in it. It also allows an implementation to allocate more space at any time, and deallocate unused space at any time. I also think it's the easiest solution.

\[\begin{align*}
\text{emptyq} & : \text{queue} \\
\text{full} & : \text{queue} \rightarrow \text{bin} \\
\neg \text{full} \ q & \Rightarrow \ \text{join} \ q \ x : \text{queue} \\
\neg \text{full} \ q & \Rightarrow \ \text{join} \ q \ x \neq \text{emptyq} \\
\neg \text{full} \ q \land \neg \text{full} \ r & \Rightarrow (\text{join} \ q \ x = \text{join} \ r \ y \equiv q=r \land x=y) \\
q \neq \text{emptyq} & \Rightarrow \ !\text{leave} \ q : \text{queue} \\
q \neq \text{emptyq} & \Rightarrow \ \text{front} \ q : X \\
\neg \text{full} \ \text{emptyq} & \Rightarrow \ !\text{leave} \ (\text{join} \ \text{emptyq} \ x) = \text{emptyq} \\
q \neq \text{emptyq} \land \neg \text{full} \ q & \Rightarrow \ !\text{leave} \ (\text{join} \ q \ x) = \text{join} \ (\text{leave} \ q) \ x \\
\neg \text{full} \ \text{emptyq} & \Rightarrow \ !\text{front} \ (\text{join} \ \text{emptyq} \ x) = x \\
q \neq \text{emptyq} \land \neg \text{full} \ q & \Rightarrow \ \text{front} \ (\text{join} \ q \ x) = \text{front} \ q
\end{align*}\]

(b) Design axioms for a limited-program-queue.

\[\begin{align*}
\text{mkemptyq} & \Rightarrow \ \text{isemptyq}' \\
\text{isemptyq} \land \neg \text{isfullq} \land \text{join} \ x & \Rightarrow \ \text{front}'=x \land \neg \text{isemptyq}' \\
\neg \text{isemptyq} \land \text{leave} & \Rightarrow \ \neg \text{isfullq}' \\
\neg \text{isemptyq} \land \neg \text{isfullq} \land \text{join} \ x & \Rightarrow \ \text{front}'=\text{front} \land \neg \text{isemptyq}' \\
\text{isemptyq} \land \neg \text{isfullq} & \Rightarrow (\text{join} \ x. \ \text{leave} \equiv \text{mkemptyq}) \\
\neg \text{isemptyq} \land \neg \text{isfullq} & \Rightarrow (\text{join} \ x. \ \text{leave} \equiv \text{leave}. \ \text{join} \ x)
\end{align*}\]

(c) Can the limit be 0?

The limit can be 0. That happens in (a) when \textit{full} is the constant \(T\) function; even \textit{full emptyq} is \(T\). In (b) it happens when \textit{isfullq} is identically \(T\).