439 (limited-queue) 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.

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

(c) Can the limit be 0 ?

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

§ I'm introducing new name *full* , which tells whether a queue is full (of course). This allows an implementation in which *full* might say $\top$ 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.

$$emptyq\text{: } queue$$
$$full\text{: } queue \to bin$$
$$\neg\ full\ q\ \Rightarrow\ join\ q\ x\text{: } queue$$
$$\neg\ full\ q\ \Rightarrow\ join\ q\ x \neq emptyq$$
$$\neg\ full\ q \land \neg\ full\ r\ \Rightarrow\ (join\ q\ x = join\ r\ y\ \ =\ \ q{=}r \land x{=}y)$$
$$q \neq emptyq\ \Rightarrow\ leave\ q\text{: } queue$$
$$q \neq emptyq\ \Rightarrow\ front\ q\text{: } X$$
$$\neg\ full\ emptyq\ \Rightarrow\ leave\ (join\ emptyq\ x)\ =\ emptyq$$
$$q \neq emptyq \land \neg\ full\ q\ \Rightarrow\ leave\ (join\ q\ x) = join\ (leave\ q)\ x$$
$$\neg\ full\ emptyq\ \Rightarrow\ front\ (join\ emptyq\ x)\ =\ x$$
$$q \neq emptyq \land \neg\ full\ q\ \Rightarrow\ front\ (join\ q\ x) = front\ \ q$$

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

§

$$mkemptyq \Rightarrow isemptyq'$$
$$isemptyq \land \neg isfullq \land join\ x\ \Rightarrow\ front'{=}x \land \neg isemptyq'$$
$$\neg isemptyq \land leave\ \Rightarrow\ \neg isfullq'$$
$$\neg isemptyq \land \neg isfullq \land join\ x\ \Rightarrow\ front'{=}front \land \neg isemptyq'$$
$$isemptyq \land \neg isfullq\ \Rightarrow\ (join\ x.\ \ leave\ \ =\ \ mkemptyq)$$
$$\neg isemptyq \land \neg isfullq\ \Rightarrow\ (join\ x.\ \ leave\ \ =\ \ leave.\ \ join\ x)$$

(c) Can the limit be 0 ?

§ The limit can be 0 . That happens in (a) when *full* is the constant $\top$ function; even *full emptyq* is $\top$. In (b) it happens when *isfullq* is identically $\top$.