441 (circular list) Design axioms for a circular list. There should be operations to create an empty list, to move along one position in the list (the first item comes after the last, in circular fashion), to insert an item at the current position, to delete the current item, and to give the current item.

After trying the question, scroll down to the solution.

§ Let the type of items in the list be X . I introduce two implementer's variables. Let L be a list of items of type X , and let i be an index of L . The theory introduces six names.

new a program that makes an empty list
mvr a program that moves right one position in the list
mvl a program that moves left one position in the list
ins a program with parameter x that inserts x into the list at the current position
del a program that deletes the item at the current position
val the value of the item at the current position in the list

Using *nat* for X , here is an example list and current position in the list.

$$L = [\,5\,;7\,;\underset{\substack{\uparrow\\ i=2}}{}1\,;8\,;6\,]$$

With this list and current position, $val=1$ and *del* deletes the 1 . When the current position is $\#L-1$ (arrow under the last ;), *mvr* makes it 0 (under [ ). When the current position is 0 (arrow under [ ), *mvl* makes it $\#L-1$ (under the last ;). For an empty list and a one-item list, the only possible current position is 0 . I think the clearest and easiest way to present the axioms might be to implement them.

$new \;=\; L{:=}\,[nil]$
$mvr \;=\; \textbf{if } \#L=0 \lor i=\#L-1 \textbf{ then } i{:=}\,0 \textbf{ else } i{:=}\,i+1 \textbf{ fi}$
$mvl \;=\; \textbf{if } \#L=0 \textbf{ then } i{:=}\,0 \textbf{ else if } i=0 \textbf{ then } i{:=}\,\#L-1 \textbf{ else } i{:=}\,i-1 \textbf{ fi fi}$
$ins \;=\; \langle x{:}\, X\cdot L{:=}\,L[0;..i] \;;;\; [x] \;;;\; L[i;..\#L]\rangle$
$del \;=\; L{:=}\,L[0;..i] \;;;\; L[i+1;..\#L]$
$val \;=\; L\,i$

If the list is empty, then *del* and *val* are undefined. We could strengthen the theory by defining them. Maybe we should add

emp a binary saying whether the list is empty

and define it as

$emp \;=\; \#L=0$

so the programmer can test whether the list is empty before using *val* and *del* . If we do add it, we can simplify *mvr* and *mvl*

$mvr \;=\; \textbf{if } i=\#L-1 \textbf{ then } i{:=}\,0 \textbf{ else } i{:=}\,i+1 \textbf{ fi}$
$mvl \;=\; \textbf{if } i=0 \textbf{ then } i{:=}\,\#L-1 \textbf{ else } i{:=}\,i-1 \textbf{ fi}$

because the programmer can test whether the list is empty before using *mvr* and *mvl* .