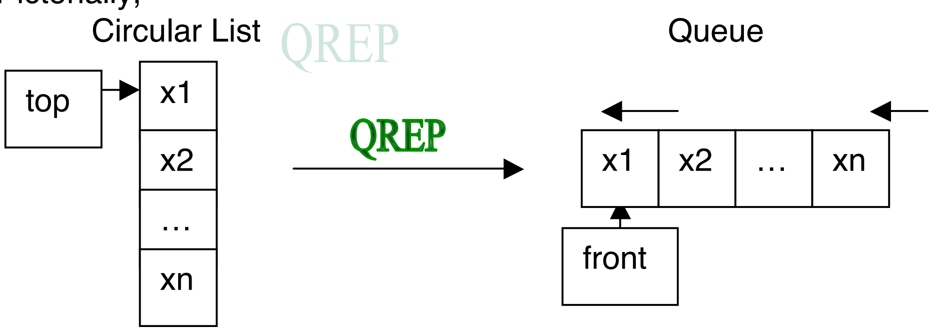
Implementing the Queue ADT with Circular Lists

This implementation is described by writing “programs” for the Queue operations using the operations available in the Circular List ADT. Of course, since these operations only apply to Circular Lists, we must choose a way to represent Queues using Circular Lists.
This implementation can be shown correct — that is, all the equations of the Queue can be shown to be satisfied using only the equations of the Circular List ADT, plus equations used to describe our representation.
The Queue ADD(ADD(… ADD(NEW,x1),x2), … , xn) is represented by Circular List INSERT(INSERT(… INSERT(CREATE,xn), … x2),x1) — that is, the Circular List in top-to-bottom order corresponds to the QUEUE in front-to-back order. Pictorially,



With this representation, the programs for the Queue operations are :
NEW □ □ CREATE
ISNEW □ □ ISEMPTY
FRONT □ □ VALUE
DEL □ □ DELETE
ADD □ □ RIGHT(INSERT(…))

The representation function, QREP: CList □ Queue is described via equations that are determined by the programs for the Queue operations. For all c and x:
20. NEW = QREP(CREATE)
21. ADD(QREP(c), x) = QREP(RIGHT(INSERT(c,x)))
22. DEL(QREP(c)) = QREP(DELETE(c))
23. FRONT(QREP(c)) = VALUE(c)
24. ISNEW(QREP(c)) = ISEMPTY(c)

From the equations for Circular Lists, and those describing the representations, we can verify each of the equations for Queue operations are satisfied by these programs.

Before beginning to prove other properties, we need to establish that the representation function QREP is surjective (onto). It should be clear from the definition that every Queue object has a representation. To formalize a proof we would need to characterize canonical representatives of all the Queue equivalence classes (we did this informally earlier), and show that QREP provides a mapping to each of them. We outlined this above and will not provide any more details.

Claim 1: DEL(NEW) = NEW.
Proof:
DEL(NEW) = DEL(QREP(CREATE)) by 20
= QREP(DELETE(CREATE)) by 22
= QREP(CREATE) by 11
= NEW by 20

Claim 2: FRONT(ADD(q,n)) = **if** ISNEW(q) **then** n **else** FRONT(q)
Proof:
Since QREP is surjective, there is a Circular List c so that QREP(c) = q. Then
FRONT(ADD(q,n)) = FRONT(ADD(QREP(c),n))
= FRONT(QREP(RIGHT(INSERT(c,n)))) by 21
= VALUE(RIGHT(INSERT(c,n))) by 23
At this point we must be aware that the constructors for Circular Lists are CREATE and INSERT so that every Circular List c is either CREATE, or INSERT(c1,x). Then the analysis proceeds by two cases.

<u>Case A</u>: c = CREATE and we continue with

VALUE(RIGHT(INSERT(c,n))) = VALUE(INSERT(c,n))) by 16
= n by 14

And in this case

ISNEW(q) = ISNEW(QREP(c))
= ISEMPTY(c) by 24
= true by 9

and hence

if ISNEW(q) **then** n **else** FRONT(q) = n

which verifies the equality for this case.

<u>Case B</u>: c = INSERT(c1,x) and we continue with

VALUE(RIGHT(INSERT(c,n))) = VALUE(RIGHT(INSERT(INSERT(c1,x),n)))
= VALUE(INSERT(RIGHT(INSERT(c,n),x))) by 17
= x by 14

And in this case

ISNEW(q) = ISNEW(QREP(c))
= ISEMPTY(INSERT(c1,x)) by 24
= false by 10

and hence

if ISNEW(q) **then** n **else** FRONT(q) = FRONT(q)
= FRONT(QREP(INSERT(c1,x)))
= VALUE(INSERT(c1,x)) by 23
= x by 14

which verifies the equality for this case.

The other Queue equations can be verified in a similar manner and the details are omitted here.