Example - Circular List (of integers) ADT

Signature

CREATE: CList
ISEMPTY: CList Boolean
INSERT: CList, Int CList
VALUE: CList Int
DELETE: CList CList
RIGHT: CList CList
JOIN: CList, CList CList

with pre-defined types Boolean and Int.

Semantics: for all c, c1 CList and i, i1 Int

9. ISEMPTY(CREATE) = true
10. ISEMPTY(INSERT(c, i)) = false
11. DELETE(CREATE) = CREATE
12. DELETE(INSERT(c, i)) = c
13. VALUE(CREATE) = "UNDEF"
14. VALUE(INSERT(c, i)) = i
15. RIGHT(CREATE) = CREATE
16. RIGHT(INSERT(CREATE, i)) = INSERT(CREATE, i)
17. RIGHT(INSERT(INSERT(c, i), i1)) = INSERT(RIGHT(INSERT(c, i1)), i)
18. JOIN(c, CREATE) = c
19. JOIN(c, INSERT(c1, i)) = INSERT(JOIN(c, c1), i)

Circular Lists amount to Stacks augmented with the two additional operations, RIGHT and JOIN. With the correspondence shown below, the behavior of the other Circular List operations is precisely that of Stacks.

<table>
<thead>
<tr>
<th>CList</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>NEWSTACK</td>
</tr>
<tr>
<td>INSERT</td>
<td>PUSH</td>
</tr>
<tr>
<td>DELETE</td>
<td>POP</td>
</tr>
<tr>
<td>VALUE</td>
<td>TOP</td>
</tr>
<tr>
<td>ISEMPTY</td>
<td>ISNEWSTACK</td>
</tr>
</tbody>
</table>