```
1  ---------- MODULE Op ----------
   Model checking basic operations on strings (i.e., list of characters).
6  EXTENDS Naturals, Sequences, AdditionalSequenceOperators
7  ----------
8  CONSTANTS   Char,
9              MaxPos,
10             MaxPr

12 ASSUME  ∧ MaxPos ∈ Nat \ {0}   WARNING: index from 1
13         ∧ MaxPr ∈ Nat \ {0}
14 ----------
15 List ≜ Seq(Char)    The set of all lists.

   The set of all operations. In this specification, we will focus on "Ins" and "Del".
21 Op ≜ [type : {"Rd"}] ∪   a read specifies no arguments
22       [type : {"Del"}, pos : MaxPos] ∪   a deletion specifies a position (from 1)
23       [type : {"Ins"}, pos : MaxPos, ch : Char, pr : MaxPr]   an insertion specifies a position (from 1), a character, ar
24 Nop ≜ CHOOSE v : v ∉ Op   Nop: an operation representing "doing nothing"
25 ----------
   Some operations for test.
29 Del1 ≜ [type ↦ "Del", pos ↦ 1]
30 Del2 ≜ [type ↦ "Del", pos ↦ 2]
31 Del3 ≜ [type ↦ "Del", pos ↦ 3]
32 Del4 ≜ [type ↦ "Del", pos ↦ 4]
33 Ins1 ≜ [type ↦ "Ins", pos ↦ 1, ch ↦ "a", pr ↦ 1]
34 Ins2 ≜ [type ↦ "Ins", pos ↦ 2, ch ↦ "b", pr ↦ 2]
35 Ins3 ≜ [type ↦ "Ins", pos ↦ 3, ch ↦ "c", pr ↦ 3]
36 Ops ≜ ⟨Ins2, Del3, Ins1, Del2, Ins3, Del1⟩
37 ----------
   The "Apply" operator which applies an operation op on the list l.
41 Apply(op, l) ≜
42     LET len ≜ Len(l)
43         pos ≜ op.pos
44     IN  CASE op = Nop → l
45           □  op.type = "Del" → SubSeq(l, 1, pos − 1) ∘ SubSeq(l, pos + 1, len)
46           □  op.type = "Ins" → Append(SubSeq(l, 1, pos − 1), op.ch) ∘ SubSeq(l, pos, len)

   The "ApplyOps" operator which applies an operation sequence ops on the list l.
52 RECURSIVE ApplyOps(_, _)
53 ApplyOps(ops, l) ≜
54     IF ops = ⟨⟩
55      THEN l
56      ELSE Apply(Last(ops), ApplyOps(AllButLast(ops), l))
57 ----------

   \* Modification History
```

* *Last* modified *Tue Jul* 03 15:37:44 *CST* 2018 by *hengxin*
* Created Sat *Jun* 23 20:56:53 *CST* 2018 by *hengxin*