```
1  ---------- MODULE Op ----------
   Model checking basic operations on strings (i.e., list of characters).
6  EXTENDS Naturals, Sequences,
7      AdditionalMathOperators, AdditionalSetOperators, AdditionalSequenceOperators
8  ------------------------------
9  CONSTANTS   Char,     set of characters allowed
10             MaxPos,   max position to insert into or delete
11             MaxPr,    max priority
12             MaxLen    max length of list

14 ASSUME  ∧ MaxPos ∈ PosInt   WARNING: index from 1
15         ∧ MaxPr ∈ PosInt
16         ∧ MaxLen ∈ PosInt
17 ------------------------------
18 List ≜ SeqMaxLen(Char, MaxLen)

   The set of all operations. In this specification, we will focus on "Ins" and "Del".
24 Rd ≜ [type : {"Rd"}]  a read specifies no arguments
25 Ins ≜ [type : {"Del"}, pos : 1 .. MaxPos]  a deletion specifies a position
26 Del ≜ [type : {"Ins"}, pos : 1 .. MaxPos, ch : Char, pr : 1 .. MaxPr]  an insertion specifies a position, a character, a

28 Op ≜ Ins ∪ Del

30 Nop ≜ PickNone(Op)   Nop: an operation representing "doing nothing"
31 ------------------------------
   Some operations for test.
35 Del1 ≜ [type ↦ "Del", pos ↦ 1]
36 Del2 ≜ [type ↦ "Del", pos ↦ 2]
37 Del3 ≜ [type ↦ "Del", pos ↦ 3]
38 Del4 ≜ [type ↦ "Del", pos ↦ 4]
39 Ins1 ≜ [type ↦ "Ins", pos ↦ 1, ch ↦ "a", pr ↦ 1]
40 Ins2 ≜ [type ↦ "Ins", pos ↦ 2, ch ↦ "b", pr ↦ 2]
41 Ins3 ≜ [type ↦ "Ins", pos ↦ 3, ch ↦ "c", pr ↦ 3]
42 Ops ≜ ⟨Ins2, Del3, Ins1, Del2, Ins3, Del1⟩

   Legal operations with respect to a list l.
47 InsOp(l) ≜ {op ∈ Ins : op.pos ≤ Len(l) + 1}  Position of an insertion cannot be too large.

49 DelOp(l) ≜
50     IF l = ⟨⟩
51       THEN {}  Not allowed to delete elements from an empty list.
52       ELSE {op ∈ Del : op.pos ≤ Len(l)}  Position of a deletion cannot be too large.
53 OpOnList(l) ≜ InsOp(l) ∪ DelOp(l)
54 ------------------------------
```

The "Apply" operator which applies an operation *op* on the list l. *Del*: If $pos > Len(l)$, the last element of l is deleted.

This is realized by the *DeleteElement* operator.

Ins: If $pos > Len(l)$, the new element is appended to l. This is realized by the *InsertElement* operator.

```
62 Apply(op, l) ≜ CASE op = Nop → l
63                  □  op.type = "Del" → DeleteElement(l, op.pos)
64                  □  op.type = "Ins" → InsertElement(l, op.ch, op.pos)
```

The "*ApplyOps*" operator which applies an operation sequence *ops* on the list l.

```
70 RECURSIVE ApplyOps(_, _)
71 ApplyOps(ops, l) ≜
72    IF ops = ⟨⟩
73     THEN l
74     ELSE Apply(Last(ops), ApplyOps(AllButLast(ops), l))
75 ⌊
```

* Modification History
* *Last* modified *Fri Jul* 06 16:25:29 *CST* 2018 by *hengxin*
* Created Sat *Jun* 23 20:56:53 *CST* 2018 by *hengxin*