```
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
10     Char    set of characters allowed
12 List ≜ Seq(Char)    all possible lists/strings
13 MaxPos ≜ Cardinality(Char) + 1    max possible position allowed
14 ------------------------------
   The set of all operations.
18 Rd ≜ [type : {"Rd"}]    a read specifies no arguments
19 Del ≜ [type : {"Del"}, pos : 1 .. MaxPos]    a deletion specifies a position, indexed from 1
20 Ins ≜ [type : {"Ins"}, pos : 1 .. MaxPos, ch : Char, pr : PosInt]    an insertion also specifies a character and a priority
22 Op ≜ Ins ∪ Del    now we don't consider Rd operations
23 Nop ≜ PickNone(Op)    Nop: a special operation representing "doing nothing"
24 ------------------------------
   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.
33 Apply(op, l) ≜ CASE op = Nop → l
34                  □  op.type = "Rd" → l
35                  □  op.type = "Del" → DeleteElement(l, op.pos)
36                  □  op.type = "Ins" → InsertElement(l, op.ch, op.pos)
   The "ApplyOps" operator which applies an operation sequence ops on the list l.
42 RECURSIVE ApplyOps(_, _)
43 ApplyOps(ops, l) ≜
44     IF ops = ⟨⟩
45      THEN l
46      ELSE Apply(Last(ops), ApplyOps(AllButLast(ops), l))
47 ------------------------------
   Check whether an operation op is legal with respect to the list l.
51 IsLegalOp(op, l) ≜ CASE op.type = "Del" → op.pos ≤ Len(l)
52                       □  op.type = "Ins" → op.pos ≤ Len(l) + 1
53 ------------------------------
   Some operations for test.
57 Del1 ≜ [type ↦ "Del", pos ↦ 1]
58 Del2 ≜ [type ↦ "Del", pos ↦ 2]
59 Del3 ≜ [type ↦ "Del", pos ↦ 3]
60 Del4 ≜ [type ↦ "Del", pos ↦ 4]
```

$Ins1 \triangleq [type \mapsto \text{“Ins”},\ pos \mapsto 1,\ ch \mapsto \text{“a”},\ pr \mapsto 1]$
$Ins2 \triangleq [type \mapsto \text{“Ins”},\ pos \mapsto 2,\ ch \mapsto \text{“b”},\ pr \mapsto 2]$
$Ins3 \triangleq [type \mapsto \text{“Ins”},\ pos \mapsto 3,\ ch \mapsto \text{“c”},\ pr \mapsto 3]$
$Ops \triangleq \langle Ins2, Del3, Ins1, Del2, Ins3, Del1 \rangle$

* Modification History
* *Last* modified *Tue Aug* 28 14:53:36 *CST* 2018 by *hengxin*
* Created Sat *Jun* 23 20:56:53 *CST* 2018 by *hengxin*