

```

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
11 List  $\triangleq$  Seq(Char)    all possible lists/strings
12 |-----|
   | The set of all operations.
16 Rd  $\triangleq$  [type : { "Rd" }] a read specifies no arguments
17 Ins  $\triangleq$  [type : { "Del" }, pos : PosInt] a deletion specifies a position, indexed from 1
18 Del  $\triangleq$  [type : { "Ins" }, pos : PosInt, ch : Char, pr : PosInt] an insertion also specifies a character and a priority
20 Op  $\triangleq$  Ins  $\cup$  Del Now we focus on "Ins" and "Del".
21 Nop  $\triangleq$  PickNone(Op) Nop: an operation representing "doing nothing"
22 |-----|
   | Some operations for test.
26 Del1  $\triangleq$  [type  $\mapsto$  "Del", pos  $\mapsto$  1]
27 Del2  $\triangleq$  [type  $\mapsto$  "Del", pos  $\mapsto$  2]
28 Del3  $\triangleq$  [type  $\mapsto$  "Del", pos  $\mapsto$  3]
29 Del4  $\triangleq$  [type  $\mapsto$  "Del", pos  $\mapsto$  4]
30 Ins1  $\triangleq$  [type  $\mapsto$  "Ins", pos  $\mapsto$  1, ch  $\mapsto$  "a", pr  $\mapsto$  1]
31 Ins2  $\triangleq$  [type  $\mapsto$  "Ins", pos  $\mapsto$  2, ch  $\mapsto$  "b", pr  $\mapsto$  2]
32 Ins3  $\triangleq$  [type  $\mapsto$  "Ins", pos  $\mapsto$  3, ch  $\mapsto$  "c", pr  $\mapsto$  3]
33 Ops  $\triangleq$   $\langle$ Ins2, Del3, Ins1, Del2, Ins3, Del1 $\rangle$ 
34 |-----|
   | 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.
43 Apply(op, l)  $\triangleq$  CASE op = Nop  $\rightarrow$  l
44                      $\square$  op.type = "Rd"  $\rightarrow$  l
45                      $\square$  op.type = "Del"  $\rightarrow$  DeleteElement(l, op.pos)
46                      $\square$  op.type = "Ins"  $\rightarrow$  InsertElement(l, op.ch, op.pos)
   |
   | The "ApplyOps" operator which applies an operation sequence ops on the list l.
52 RECURSIVE ApplyOps(-, -)
53 ApplyOps(ops, l)  $\triangleq$ 
54     IF ops =  $\langle$  $\rangle$ 
55     THEN l
56     ELSE Apply(Last(ops), ApplyOps(AllButLast(ops), l))
57 |-----|
   | Check whether an operation op is legal with respect to the list l.

```

$$\begin{array}{ll} 61 & IsLegalOp(op, l) \triangleq \text{CASE } op.type = \text{"Del"} \rightarrow op.pos \leq Len(l) \\ 62 & \quad \square \quad op.type = \text{"Ins"} \rightarrow op.pos \leq Len(l) + 1 \end{array}$$

Legalize an operation  $op$  with respect to the list  $l$ .

$$\begin{aligned} \text{67 } & LegalizeOp(op, l) \triangleq \text{CASE } op.type = "Del" \rightarrow [op \text{ EXCEPT !}.pos = Min(@, Len(l))] \\ \text{68 } & \quad \square \quad op.type = "Ins" \rightarrow [op \text{ EXCEPT !}.pos = Min(@, Len(l) + 1)] \\ \text{69 } & \end{aligned}$$

```
\ * Modification History
\ * Last modified Sat Jul 07 21:26:42 CST 2018 by hengxin
\ * Created Sat Jun 23 20:56:53 CST 2018 by hengxin
```