

```

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,
11             MaxLen

13 ASSUME   $\wedge \text{MaxPos} \in \text{Nat} \setminus \{0\}$   WARNING: index from 1
14          $\wedge \text{MaxPr} \in \text{Nat} \setminus \{0\}$ 
15          $\wedge \text{MaxLen} \in \text{Nat} \setminus \{0\}$ 
16 |-----|
17 List  $\triangleq \text{Seq}(\text{Char}) \setminus *$  The set of all lists.
18 List  $\triangleq \text{UNION } \{[1 \dots m \rightarrow \text{Char}] : m \in 0 \dots \text{MaxLen}\}$ 

  |The set of all operations. In this specification, we will focus on “Ins” and “Del”.|
24 Op  $\triangleq$  [type: {“Rd”}]  $\cup \setminus *$  a read specifies no arguments
25          [type: {“Del”}, pos : 1 .. MaxPos]  $\cup$  a deletion specifies a position
26          [type: {“Ins”}, pos : 1 .. MaxPos, ch : Char, pr : 1 .. MaxPr] an insertion specifies a position, a character, a
27
28 Nop  $\triangleq \text{CHOOSE } v : v \notin \text{Op}$  Nop: an operation representing “doing nothing”
29 |-----|
  |Some operations for test.|
33 Del1  $\triangleq$  [type  $\mapsto$  “Del”, pos  $\mapsto$  1]
34 Del2  $\triangleq$  [type  $\mapsto$  “Del”, pos  $\mapsto$  2]
35 Del3  $\triangleq$  [type  $\mapsto$  “Del”, pos  $\mapsto$  3]
36 Del4  $\triangleq$  [type  $\mapsto$  “Del”, pos  $\mapsto$  4]
37 Ins1  $\triangleq$  [type  $\mapsto$  “Ins”, pos  $\mapsto$  1, ch  $\mapsto$  “a”, pr  $\mapsto$  1]
38 Ins2  $\triangleq$  [type  $\mapsto$  “Ins”, pos  $\mapsto$  2, ch  $\mapsto$  “b”, pr  $\mapsto$  2]
39 Ins3  $\triangleq$  [type  $\mapsto$  “Ins”, pos  $\mapsto$  3, ch  $\mapsto$  “c”, pr  $\mapsto$  3]
40 Ops  $\triangleq \langle \text{Ins2}, \text{Del3}, \text{Ins1}, \text{Del2}, \text{Ins3}, \text{Del1} \rangle$ 
41 |-----|
  |The “Apply” operator which applies an operation op on the list l.|
45 Apply(op, l)  $\triangleq$ 
46   LET len  $\triangleq \text{Len}(l)$ 
47       pos  $\triangleq \text{op.pos}$ 
48   IN CASE op = Nop  $\rightarrow l$ 
49        $\square$  op.type = “Del”  $\rightarrow \text{SubSeq}(l, 1, \text{pos} - 1) \circ \text{SubSeq}(l, \text{pos} + 1, \text{len})$ 
50        $\square$  op.type = “Ins”  $\rightarrow \text{Append}(\text{SubSeq}(l, 1, \text{pos} - 1), \text{op.ch}) \circ \text{SubSeq}(l, \text{pos}, \text{len})$ 

  |The “ApplyOps” operator which applies an operation sequence ops on the list l.|
56 RECURSIVE ApplyOps(-, -)
57 ApplyOps(ops, l)  $\triangleq$ 
58   IF ops =  $\langle \rangle$ 

```

```

59      THEN  $l$ 
60      ELSE  $Apply(Last(ops), ApplyOps(AllButLast(ops), l))$ 
61  ]

```

```

\ * Modification History
\ * Last modified Tue Jul 03 16:03:00 CST 2018 by hengxin
\ * Created Sat Jun 23 20:56:53 CST 2018 by hengxin

```