

```

1  ┌────────────────────────── MODULE or_set ───────────────────────────┐
2  EXTENDS Integers, Sequences, TLC
3  CONSTANTS N, Data
4
5  Procs  $\triangleq$  1 .. N
6
7
8  --algorithm or_set
9  variables
10   ops = [j ∈ Procs ↦ ⟨⟩];  to broadcast operations
11
12  define  atSource phases
13   _Lookup(set, e)  $\triangleq$   $\exists s \in \text{set} : s.val = e$ 
14
15   _Add(proc, e)  $\triangleq$  { [key ↦ ToString(proc) ∘ ToString(e), val ↦ e] }
16
17   _Remove(set, e)  $\triangleq$  IF _Lookup(set, e) THEN
18     { CHOOSE s ∈ set : s.val = e }
19   ELSE
20     {}
21  end define ;
22
23   send a operation to all
24  macro Broadcast(o, s) begin
25    if s ≠ {} then
26      ops := [j ∈ Procs ↦ Append(ops[j], [op ↦ o, set ↦ s])];
27    end if
28  end macro ;
29
30   receive and process operations, one by one
31  macro Update(s) begin
32    if Len(ops[self]) > 0 then
33      if Head(ops[self]).op = "A" then
34        s := s ∪ Head(ops[self]).set;
35      elsif Head(ops[self]).op = "R" then
36        s := s \ Head(ops[self]).set;
37      end if ;
38      ops[self] := Tail(ops[self]);  clear processed operation
39    end if ;
40  end macro ;
41
42  process Set ∈ Procs
43  variables
44   set = {};  local set of pairs [key ↦ "", val ↦ ""]
45  begin Main:
46    while TRUE do
47      Update(set);
48      either Add:
49        with var ∈ Data do  select a random value to add

```

```

50   Broadcast("A", _Add(self, var));
51   end with ;
52   or Remove:
53   with var ∈ Data do select a random value to remove
54   Broadcast("R", _Remove(set, var));
55   end with ;
56   end either ;
57   end while ;
58 end process ;

60 end algorithm ;

62 BEGIN TRANSLATION
63 VARIABLES ops, pc

64 define statement
65  $\_Lookup(set, e) \triangleq \exists s \in set : s.val = e$ 
66  $\_Add(proc, e) \triangleq \{[key \mapsto ToString(proc) \circ ToString(e), val \mapsto e]\}$ 
67  $\_Remove(set, e) \triangleq \text{IF } \_Lookup(set, e) \text{ THEN}$ 
68   {CHOOSE  $s \in set : s.val = e$ }
69 ELSE
70   {}
71 END IF

72 VARIABLE set

73 vars  $\triangleq \langle ops, pc, set \rangle$ 

74 ProcSet  $\triangleq (Procs)$ 

75 Init  $\triangleq$  Global variables
76    $\wedge ops = [j \in Procs \mapsto \langle \rangle]$ 
77   Process Set
78    $\wedge set = [self \in Procs \mapsto \{\}]$ 
79    $\wedge pc = [self \in ProcSet \mapsto \text{"Main"}]$ 

80 Main(self)  $\triangleq$   $\wedge pc[self] = \text{"Main"}$ 
81    $\wedge \text{IF } Len(ops[self]) > 0$ 
82     THEN  $\wedge \text{IF } Head(ops[self]).op = \text{"A"}$ 
83       THEN  $\wedge set' = [set \text{ EXCEPT } ![self] = set[self] \cup Head(ops[self]).set]$ 
84       ELSE  $\wedge \text{IF } Head(ops[self]).op = \text{"R"}$ 
85         THEN  $\wedge set' = [set \text{ EXCEPT } ![self] = set[self] \setminus Head(ops[self])]$ 
86         ELSE  $\wedge \text{TRUE}$ 
87          $\wedge set' = set$ 
88        $\wedge ops' = [ops \text{ EXCEPT } ![self] = Tail(ops[self])]$ 
89     ELSE  $\wedge \text{TRUE}$ 
90      $\wedge \text{UNCHANGED } \langle ops, set \rangle$ 
91    $\wedge \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Add"}]$ 

```

