

```

1  ┌────────────────────────── MODULE VectorClocks ───────────────────────────┐
2  EXTENDS Integers, TLC, Sequences
3  CONSTANTS Procs, MAX

6  --algorithm VectorClocks
7  variables
8     $msgs = [p \in Procs \mapsto [q \in Procs \mapsto 0]]$ ;  defined as Vector Clock

10 define
11   returns the maximum value for each element of two vectors
12    $PairMax(v1, v2) \triangleq [p \in Procs \mapsto \text{IF } v1[p] > v2[p] \text{ THEN } v1[p] \text{ ELSE } v2[p]]$ 
13   increments by 1 the 'e' element of the vector 'v'
14    $Increment(e, v) \triangleq [p \in Procs \mapsto \text{IF } p = e \text{ THEN } v[p] + 1 \text{ ELSE } v[p]]$ 
15 end define ;

17 fair process VectorClock  $\in Procs$ 
18 variables
19    $vc = [p \in Procs \mapsto 0]$   Initially all clocks are zero
20 begin
21   Main:
22     while ( $vc[self] < MAX$ ) do
23       either
24         Receive: increment local clock and calculates maximum of two clocks
25          $vc :=$ 
26           LET  $vM \triangleq PairMax(vc, msgs[self])$ 
27           IN  $Increment(self, vM)$ ;
28       or
29         Send: increment local clock and send it
30          $vc := Increment(self, vc)$ ;
31         with  $p \in Procs \setminus \{self\}$  do  send  $vc$  to 'p' via  $msgs[p]$ 
32            $msgs[p] := vc$ ;
33         end with ;
34       end either ;
35     end while ;
36 end process ;

38 end algorithm ;

40 BEGIN TRANSLATION
41 VARIABLES  $msgs, pc$ 

43 define statement
44    $PairMax(v1, v2) \triangleq [p \in Procs \mapsto \text{IF } v1[p] > v2[p] \text{ THEN } v1[p] \text{ ELSE } v2[p]]$ 
46    $Increment(e, v) \triangleq [p \in Procs \mapsto \text{IF } p = e \text{ THEN } v[p] + 1 \text{ ELSE } v[p]]$ 
48 VARIABLE  $vc$ 
50 vars  $\triangleq \langle msgs, pc, vc \rangle$ 

```

```

52  $ProcSet \triangleq (Procs)$ 

54  $Init \triangleq$  Global variables
55  $\wedge msgs = [p \in Procs \mapsto [q \in Procs \mapsto 0]]$ 
56 Process  $VectorClock$ 
57  $\wedge vc = [self \in Procs \mapsto [p \in Procs \mapsto 0]]$ 
58  $\wedge pc = [self \in ProcSet \mapsto \text{"Main"}]$ 

60  $Main(self) \triangleq$   $\wedge pc[self] = \text{"Main"}$ 
61  $\wedge \text{IF } (vc[self][self] < MAX)$ 
62  $\quad \text{THEN } \wedge \vee \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Receive"}]$ 
63  $\quad \vee \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Send"}]$ 
64  $\quad \text{ELSE } \wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Done"}]$ 
65  $\wedge \text{UNCHANGED } \langle msgs, vc \rangle$ 

67  $Receive(self) \triangleq$   $\wedge pc[self] = \text{"Receive"}$ 
68  $\wedge vc' = [vc \text{ EXCEPT } ![self] = \text{LET } vM \triangleq PairMax(vc[self], msgs[self])$ 
69  $\quad \text{IN } Increment(self, vM)]$ 
70  $\wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Main"}]$ 
71  $\wedge msgs' = msgs$ 

73  $Send(self) \triangleq$   $\wedge pc[self] = \text{"Send"}$ 
74  $\wedge vc' = [vc \text{ EXCEPT } ![self] = Increment(self, vc[self])]$ 
75  $\wedge \exists p \in Procs \setminus \{self\} :$ 
76  $\quad msgs' = [msgs \text{ EXCEPT } ![p] = vc'[self]]$ 
77  $\wedge pc' = [pc \text{ EXCEPT } ![self] = \text{"Main"}]$ 

79  $VectorClock(self) \triangleq Main(self) \vee Receive(self) \vee Send(self)$ 

81  $Next \triangleq (\exists self \in Procs : VectorClock(self))$ 
82  $\vee$  Disjunct to prevent deadlock on termination
83  $((\forall self \in ProcSet : pc[self] = \text{"Done"}) \wedge \text{UNCHANGED } vars)$ 

85  $Spec \triangleq$   $\wedge Init \wedge \square [Next]_{vars}$ 
86  $\wedge \forall self \in Procs : \text{WF}_{vars}(VectorClock(self))$ 

88  $Termination \triangleq \diamond (\forall self \in ProcSet : pc[self] = \text{"Done"})$ 

90 END TRANSLATION

92 Boundedness
93  $VectorClockOK \triangleq (\forall k, l \in Procs : vc[k][k] \geq vc[l][k])$ 

```

```

\ * Modification History
\ * Last modified Sun Nov 25 09:11:52 PST 2018 by ocosta
\ * Created Sat Nov 17 15:07:39 PST 2018 by ocosta

```