Algorithm – Atomic Register

Algorithm 1 Read-Impose Write-Majority

Implements:

$(1, N)$-AtomicRegister, **instance** *onar*.

Uses:

BestEffortBroadcast, **instance** *beb*.

PerfectPointToPointLinks, **instance** *pp2p*.

1: **upon event** $\langle$ *Init* $\rangle$ **do**
2: $(ts, val) := (0, \bot)$
3: $wts := 0$
4: $acks := 0$
5: $rid := 0$
6: $\forall_{p \in \Pi}$ $readlist[p] := \bot$
7: $readval := \bot$
8: $reading :=$ FALSE
9: **upon event** $\langle$ *onar*, *Read* $\rangle$ **do**
10: $rid := rid + 1$
11: $acks := 0$
12: $\forall_{p \in \Pi}$ $readlist[p] := \bot$
13: $reading :=$ TRUE
14: **trigger** $\langle$ *beb*, *Broadcast* | [READ, *rid*] $\rangle$
15: **upon event** $\langle$ *beb*, *Deliver* | *p*, [READ, *r*] $\rangle$ **do**
16: **trigger** $\langle$ *pp2p*, *Send* | *p*, [VALUE, *r*, *ts*, *val*] $\rangle$
17: **upon event** $\langle$ *pp2p*, *Deliver* | *p*, [VALUE, *r*, *ts*′, *v*′] $\rangle$ **do**
18: **if** $r = rid$ **then**
19: $readlist[p] := (ts', v')$
20: **if** $|readlist| > \frac{N}{2}$ **then** ▷ Where $N = |\Pi|$.
21: $(maxts, readval) :=$ HIGHEST($readlist$)
22: $\forall_{q \in \Pi}$ $readlist[q] := \bot$
23: **trigger** $\langle$ *beb*, *Broadcast* | [WRITE, *rid*, *maxts*, *readval*] $\rangle$

```
24: upon event ⟨ onar, Write | v ⟩ do
25:     rid := rid + 1
26:     wts := wts + 1
27:     acks := 0
28:     trigger ⟨ beb, Broadcast | [WRITE, rid, wts, v] ⟩
29: upon event ⟨ beb, Deliver | p, [WRITE, r, ts′, v′] ⟩ do
30:     if ts′ > ts then
31:         (ts, val) := (ts′, v′)
32:     trigger ⟨ pp2p, Send | p, [ACK, r] ⟩
33: upon event ⟨ pp2p, Deliver | p, [ACK, r] ⟩ do
34:     if r = rid then
35:         acks := acks + 1
36:         if acks > N/2 then
37:             acks := 0
38:             if reading then
39:                 reading := FALSE
40:                 trigger ⟨ onar, ReadReturn | readval ⟩
41:             else
42:                 trigger ⟨ onar, WriteReturn |  ⟩
```

Algorithm 2 Read-Impose Write-Consult-Majority

Implements:

(N, N)-AtomicRegister, **instance** *nnar*.

Uses:

BestEffortBroadcast, **instance** *beb*.

PerfectPointToPointLinks, **instance** *pp2p*.

1: **upon event** $\langle$ *Init* $\rangle$ **do**
2: $(ts, wr, val) := (0, 0, \bot)$
3: $acks := 0$
4: $writeval := \bot$
5: $rid := 0$
6: $\forall_{p \in \Pi}\ readlist[p] := \bot$
7: $readval := \bot$
8: $reading :=$ FALSE
9: **upon event** $\langle$ *nnar*, *Read* $\rangle$ **do**
10: $rid := rid + 1$
11: $acks := 0$
12: $\forall_{p \in \Pi}\ readlist[p] := \bot$
13: $reading :=$ TRUE
14: **trigger** $\langle$ *beb*, *Broadcast* | [READ, *rid*] $\rangle$
15: **upon event** $\langle$ *beb*, *Deliver* | *p*, [READ, *r*] $\rangle$ **do**
16: **trigger** $\langle$ *pp2p*, *Send* | *p*, [VALUE, *r*, *ts*, *wr*, *val*] $\rangle$
17: **upon event** $\langle$ *pp2p*, *Deliver* | *p*, [VALUE, *r*, *ts′*, *wr′*, *v′*] $\rangle$ **do**
18: **if** $r = rid$ **then**
19: $readlist[p] := (ts', wr', v')$
20: **if** $|readlist| > \frac{N}{2}$ **then** ▷ Where $N = |\Pi|$.
21: $(maxts, rr, readval) :=$ HIGHEST($readlist$)
22: $\forall_{q \in \Pi}\ readlist[q] := \bot$
23: **if** *reading* **then**
24: $bcastval := readval$
25: **else**
26: $rr :=$ RANK(*self*)
27: $maxts := maxts + 1$
28: $bcastval := writeval$
29: **trigger** $\langle$ *beb*, *Broadcast* | [WRITE, *rid*, *maxts*, *rr*, *bcastval*] $\rangle$

```
30: upon event ⟨ nnar, Write | v ⟩ do
31:     rid := rid + 1
32:     writeval := v
33:     acks := 0
34:     ∀_{p∈Π} readlist[p] := ⊥
35:     trigger ⟨ beb, Broadcast | [READ, rid] ⟩
36: upon event ⟨ beb, Deliver | p, [WRITE, r, ts′, wr′, v′] ⟩ do
37:     if (ts′, wr′) > (ts, wr) then                ▷ Tuple comparison.
38:         (ts, wr, val) := (ts′, wr′, v′)
39:     trigger ⟨ pp2p, Send | p, [ACK, r] ⟩
40: upon event ⟨ pp2p, Deliver | p, [ACK, r] ⟩ do
41:     if r = rid then
42:         acks := acks + 1
43:         if acks > N/2 then
44:             acks := 0
45:             if reading then
46:                 reading := FALSE
47:                 trigger ⟨ nnar, ReadReturn | readval ⟩
48:             else
49:                 trigger ⟨ nnar, WriteReturn | ⟩
```